NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCE

DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

POSTGRADUATE STUDIES

MASTER THESIS

Visualizing and Exploring Time-Evolving Linked Geospatial Data

Georgios Stamoulis

Supervisor: **Manolis Koubarakis**, Professor UoA
Co-Supervisor: **Charalampos Nikolaou**, Ph.D. Candidate UoA

ATHENS

July 2015

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΜΕΤΑΠΤΥΧΙΑΚΕΣ ΣΠΟΥΔΕΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Οπτικοποίηση και Εξερεύνηση Χρονικά Εξελισσόμενων Διασυνδεδεμένων Γεωχωρικών Δεδομένων

Γεώργιος Σταμούλης

Επιβλέπων: **Μανόλης Κουμπαράκης**, Καθηγητής ΕΚΠΑ
Συνεπιβλέπων: **Χαράλαμπος Νικολάου**, Υποψήφιος Διδάκτωρ ΕΚΠΑ

ΑΘΗΝΑ

Ιούλιος 2015

MASTER THESIS

Visualizing and Exploring Time-Evolving Linked Geospatial Data

Georgios Stamoulis

R.N.: M1224

SUPERVISOR:

Manolis Koubarakis, Professor UoA

CO-SUPERVISOR:

Charalampos Nikolaou, Ph.D. Candidate UoA

EXAMINATION COMMITEE:

Stathes Hadjiefthymiades, Assistant Professor UoA

ATHENS

July 2015

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Οπτικοποίηση και Εξερεύνηση Χρονικά Εξελισσόμενων Διασυνδεδεμένων Γεωχωρικών Δεδομένων

Γεώργιος Σταμούλης

Α.Μ.: Μ1224

ΕΠΙΒΛΕΠΩΝ:
Μανόλης Κουμπαράκης, Καθηγητής ΕΚΠΑ
ΣΥΝΕΠΙΒΛΕΠΩΝ:
Χαράλαμπος Νικολάου, Υποψήφιος Διδάκτωρ ΕΚΠΑ

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ:
Ευστάθιος Χατζηευθυμιάδης, Αναπληρωτής Καθηγητής ΕΚΠΑ

ΑΘΗΝΑ
Ιούλιος 2015

Abstract

Linked geospatial data are receiving growing attention as researchers and practitioners are using the matured semantic web technologies to publish and structure web data. New data are constantly produced, but also changes in form of updates occur over the existing data sets and the temporal dimension has started to play a crucial role. While the list of geospatial and temporal data made available as linked data is growing, the need of managing and using this information to address broad environmental or social needs across geographic scales has emerged. Applications for exploiting this abundance of geospatial information have started to emerge, that focus on browsing and exploring linked geospatial datasets and combine them with other heterogeneous geospatial data to create thematic maps that are useful for analyzing or assessing an event or situation.

In this thesis we try to remedy the shortcomings of the original version of Sextant and focus on creating a user-friendly application enhanced with features, which exist in the matured Geographic Information Systems, allowing both experts and non-experts to explore and visualize linked geospatial data. A web-based and mobile ready platform for exploring and interacting with linked geospatial and temporal data, producing statistical charts, as well as creating, sharing and searching for thematic maps that combine geospatial information from different sources. Many new features were added and new functionality has been introduced in an attempt to create a user-friendly application that would allow both domain experts and non-experts to take advantage of semantic web technologies, and convince them to adopt these technologies by presenting the benefits of the linked open geospatial Web through the use of Sextant.

SUBJECT AREA: Data Visualization

KEYWORDS: data visualization, data exploration, semantic web, linked geospatial data, linked spatiotemporal data, thematic maps

Περίληψη

Τα διασυνδεδεμένα δεδομένα λαμβάνουν όλο και περισσότερη προσοχή καθώς ερευνητές και επαγγελματίες χρησιμοποιούν τις τεχνολογίες του σημασιολογικού ιστού για να εκδόσουν και να δομήσουν τα δεδομένα του Ιστού. Νέα δεδομένα παράγονται συνεχώς, αλλά και αλλαγές στα υπάρχοντα δεδομένα λαμβάνουν χώρο με την μορφή επικαιροποιήσεων, και η χρονική διάσταση αρχίζει να παίζει σημαντικό ρόλο. Όσο πληθαίνουν τα δεδομένα που γίνονται διαθέσιμα σαν διασυνδεδεμένα, εμφανίζεται η ευκαιρία χρήσης της πληροφορίας αυτής για αντιμετώπιση προβλημάτων σε ένα ευρύ φάσμα κοινωνικών και περιβαλλοντολογικών αναγκών. Εφαρμογές που χρησιμοποιούν την πληθόρα γεωχωρικής πληροφορίας που είναι πλέον διαθέσιμη αρχίζουν να δημιουργούνται εστιάζοντας στην εξερεύνηση των διασυνδεδεμένων γεωχωρικών δεδομένων και συνδυασμό με άλλες ετερογενείς πηγές γεωχωρικών δεδομένων για τη δημιουργία θεματικών χαρτών που είναι χρήσιμοι για ανάλυση και αξιολόγηση καταστάσεων.

Σε αυτή τη διπλωματική εργασία προσπαθούμε να επαναπροσδιορίσουμε την αρχική έκδοση του Sextant και εστιάζουμε στη δημιουργία μιας εφαρμογής φιλικής προς τον χρήστη, εφοδιασμένη με εργαλεία που παρέχονται στα Συστήματα Γεωγραφικών Πληροφοριών, για την εξερεύνηση και οπτικοποίηση των διασυνδεδεμένων γεωχωρικών δεδομένων. Παρουσιάζουμε μια web εφαρμογή διαθέσιμης και σε android, για εξερεύνηση διασυνδεδεμένων γεωχωρικών και χρονικών δεδομένων, παραγωγή στατιστικών διαγραμμάτων, καθώς και δημιουργία, διαμοιρασμό και αναζήτηση θεματικών χαρτών που συνδυάζουν γεωχωρική πληροφορία από διάφορες πηγές. Νέα χαρακτηριστικά έχουν προστεθεί στα πλαίσια μιας προσπάθειας να δημιουργήσουμε ένα φιλικό στον χρήστη περιβάλλον που θα δίνει τη δυνατότητα τόσο στους ειδικούς, όσο και στους μη ειδικούς στις τεχνολογίες του σημασιολογικού ιστού να έρθουν σε επαφή με αυτές και να αναγνωρίσουν τα πλεονεκτήματά τους.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Οπτικοποίηση Δεδομένων

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: οπτικοποίηση δεδομένων, εξερεύνηση δεδομένων, σημασιολογικός ιστός, διασυνδεδεμένα γεωχωρικά δεδομένα, διασυνδεδεμένα χωρικά και χρονικά δεδομένα, θεματικοί χάρτες

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον καθηγητή μου και επιβλέποντα αυτής της διπλωματικής, Μανόλη Κουμπαράκη, καθώς και τον συνεπιβλέποντα, Χαράλαμπο Νικολάου, για την καθοδήγησή τους, την κοπέλα μου, Καλλιόπη Ζαφείρη, για την στήριξη και τις εικαστικές της παρεμβάσεις, τους φίλους μου για την συμπαράστασή τους και τις πολύτιμες συμβουλές τους, καθώς και όλη την ερευνητική μας ομάδα για την άριστη συνεργασία και την πολύτιμη στήριξή τους κατά τη διάρκεια εκπόνησης της εργασίας αυτής.

Αυτή η εργασία χρηματοδοτήθηκε μερικώς από τα FP7 projects LEO (611141) (`http://linkedeodata.eu`) και MELODIES (603525) (`http://www.melodiesproject.eu`).

Contents

List of Figures

Chapter 1
Introduction

Linked data is a research area which studies how one can make RDF data available on the Web, and interlink it with other data in order to increase its value for users [9]. The goal of Linked Data is to allow people to share structured data on the web as easily as they can do with documents today.

An important step for the evolution of the Web of documents to the Web of data is the transformation of the data from any form that they exist into a common format, the Resource Description Framework (RDF) so that it can be easily integrated with other data already transformed in this format. All the data that is compatible with the Linked Data Principles[1] composes the Linked Open Data (LOD) cloud (Figure 1.1).

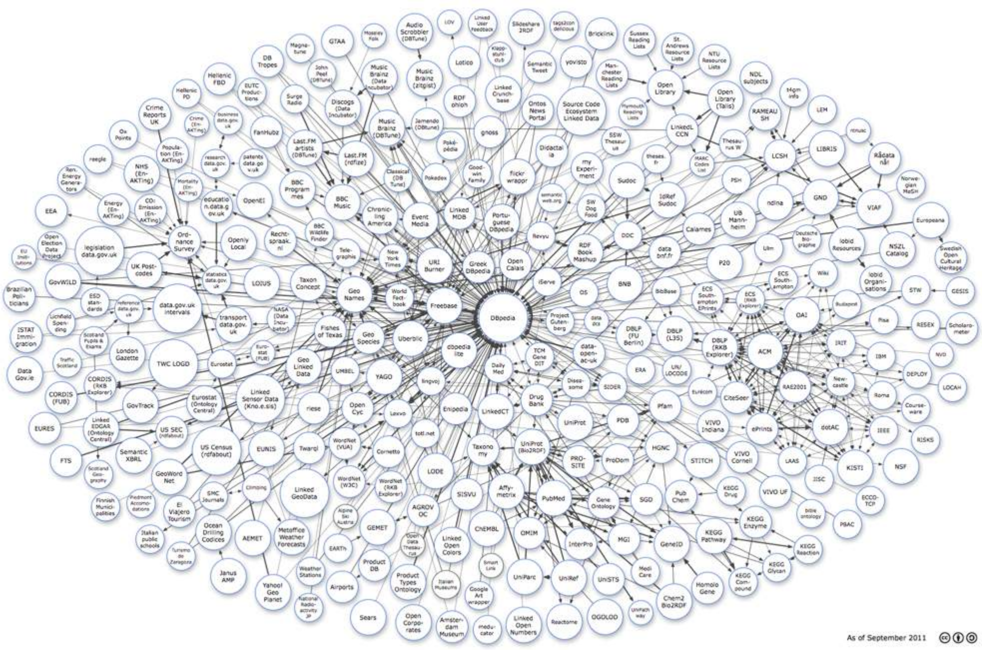


Figure 1.1: Recent state of the Linked Open Data cloud

Recently, spatial and temporal extensions to RDF have been proposed and implemented. GeoSPARQL [29] is a recent OGC standard that allows representing and querying geospatial data on the Semantic Web. Also, the data model stRDF accompanied by

[1] http://www.w3.org/DesignIssues/LinkedData.html

the query language stSPARQL [19] are extensions of the standard RDF and SPARQL for representing and querying geospatial data that changes over time. Both of the above extensions are implemented in the open source spatiotemporal RDF store Strabon [21].

Linked geospatial data has recently received attention as researchers and practitioners have started tapping the wealth of geospatial information available on the Web. As a result, in the last few years semantic web has been equipped with mature technologies that enable researchers and practitioners to represent geospatial information in RDF and query it using SPARQL [20, 21]. The technological arsenal of semantic web is offering also a wealth of tools and systems for querying [21, 31, 6], publishing [23], and linking [27, 33] geospatial information represented in RDF.

Advancements in the management of linked geospatial data has been followed and reinforced by many efforts which have jumped on the linked open data (LOD) bandwagon by populating the LOD cloud with geospatial information. The most important effort is Ordnance Survey, which is the first national mapping agency that has made various kinds of geospatial data from Great Britain available as open linked data[2]. Another representative example is LinkedGeoData[3] where OpenStreetMap (OSM) data are made available as RDF and queried using SPARQL [5]. A similar effort is GeoLinkedData[4] where geospatial data from Spain is made public using RDF [24]. Similarly, the Greek Linked Open Data portal[5], which is being developed by our group, has recently published a number of open datasets for the area of Greece as linked open data.

Applications for exploiting this abundance of geospatial information have also started to emerge. For example, in project TELEIOS[6], National and Kapodistrian University of Athens has built a service for real-time fire monitoring based on satellite images and open linked geospatial data [18]. In the same project, a burn scar mapping service has been developed which offers maps of burned areas of Greece for the period 1984-2012 based on Landsat images [15]. While developing such applications, we were faced with the tasks of browsing and exploration of multiple linked spatiotemporal datasets together with other heterogeneous geospatial data available on the web (e.g., geospatial data available in file formats, such as KML, shapefiles, and GeoTIFF). Another important task in these applications was the production of thematic maps, i.e., maps useful to end-users for analyzing or assessing an event or situation, like analysis of environmental damages, fire monitoring and management, etc. In its most general formulation, this problem is known

[2] http://www.ordnancesurvey.co.uk/oswebsite/products/os-opendata.html
[3] http://linkedgeodata.org/
[4] http://geo.linkeddata.es/
[5] http://linkedopendata.gr/
[6] http://earthobservatory.eu/

as *geospatial mapping*.

Our tool Sextant[7] was the first one to offer functionalities for supporting geospatial mapping. Sextant[8] can be used to produce thematic maps by layering geospatial and temporal information which exists in a number of heterogeneous data sources ranging from standard SPARQL endpoints, to SPARQL endpoints following the standard GeoSPARQL defined by the Open Geospatial Consortium (OGC), or the well-adopted geospatial file format KML [30].

Sextant has been introduced in the papers [28, 7, 16] where we demonstrated its usefulness in the domains of Earth Observation and Environment by presenting its browsing and visualization capabilities using a number of link geospatial datasets and other geospatial data sources publicly available on the Web. In particular, we showed how Sextant can be used by an expert in Earth Observation (EO) as a tool to serve the two use-cases of the TELEIOS project, namely, rapid mapping and knowledge discovery from satellite images. While the feedback received from the two user partners of the project, the German Aerospace Center (DLR) and the National Observatory of Athens (NOA) was encouraging in terms of the offered functionality, it was discouraging in terms of usability. Briefly, Sextant was considered hard to be operated by users not familiar with semantic web technologies, while it missed a handful of features needed by EO experts, such as visualization of statistical data, support of certain raster and vector formats, editing capabilities, and a mobile counterpart version.

1.1 Contributions of this thesis

In this thesis we try to remedy the shortcomings of the original version of Sextant and focus on creating a user-friendly application enhanced with features, which exist in the matured Geographic Information Systems, allowing both experts and non-experts to explore and visualize linked geospatial data.

Both the user interface and the server components were redesigned and re-implemented. Two clients were developed in parallel, the web-client that is designed to be cross-platform and cross-browser, and a mobile-client that runs on Android 4.3 and up. We manage to provide a user-friendly interface that enables the search and visualization of linked geospatial and temporal RDF data in an intuitive way, without the need of in depth knowledge of SPARQL. The main objective is to expose semantic web technologies and

[7] `http://sextant.di.uoa.gr/`

[8] The naming of our system is inspired by the navigation instrument sextant (`http://wikipedia.org/wiki/Sextant`) used to measure the angle between any two visible objects.

allow the exploration and visualization of RDF data through Sextant.

Many new features were added and new functionality has been introduced in an attempt to create an application that allows both domain experts and non-experts to use the full potential of semantic web technologies combined with the capabilities of Geographic Information Systems (GIS). To achieve this goal we added support for the most promising GIS file formats and introduced data analyzing techniques such as color maps and statistical charts. We also provide alternatives for non-experts to explore RDF data from various SPARQL endpoints, such as the predefined queries feature.

Thematic maps are the core of this application. We enhanced the map ontology with a list of metadata that provide a complete legend for a created map and allows the search for specific maps. Maps are resources and can easily be shared and loaded in Sextant through their unique URI. The representation of maps in RDF triples allows the storage in SPARQL endpoints that act as map registries. As these registries become populated with various maps, users can search for specific maps through the use of the map metadata. Sextant provides an intuitive map search on these metadata without the need of writing SPARQL queries, to assist non-experts in using this feature.

In this manner we manage to expose semantic web technologies to users from various domains, and even convince them to adopt these technologies by presenting the benefits of the linked open geospatial web through the use of Sextant.

1.2 Thesis Outline

The structure of this thesis is organized as follows. In Chapter 2 we present the background knowledge on the model RDF, the query language SPARQL, how we build ontologies to describe our worlds and how to represent geospatial information in RDF. We also discuss related work in the filed of exploring and visualizing geospatial data. In Chapter 3 we have the architecture of Sextant and a description of the components of the client and the server side of the application. We continue with Chapter 4 where we present the map ontology that is used to represent our map in the RDF model and assists us on organizing our maps in registries that allow easy sharing and search for maps. In Chapter 5 we present all the different types of data formats that have emerged in the GIS world and how we incorporate them in Sextant. Also we showcase the capabilities of Sextant in layer manipulation, such as coloring features. Chapter 6 introduces the time dimension, how it was incorporated in the RDF model and what tools we provide in Sextant to use this information to create time events. In Chapter 7 we discuss the need to create catalogues to assist us on searching for Earth Observation Data and showcase the use of GEOSS data core catalogues to enhance the search capabilities of Sextant in EO products. Also we present how Sextant can be used to query multiple SPARQL endpoints and create layers on the map using the results of those queries. Chapter 8 introduces the RDF Data Cube Vocabulary and a new method for enhancing an existing data set expressed in RDF, so that we can easily produce statistical charts. We demonstrate how to enhance a data set, and how to use Sextant to produce charts in an intuitive way. Finally, in Chapter 9 we conclude the work and discuss future directions.

Chapter 2
Background and Related Work

Linked geospatial data has recently received attention as researchers and practitioners have started tapping the wealth of geospatial information available on the Web. As a result, in the last few years semantic web has been equipped with mature technologies that enable researchers and practitioners to represent geospatial information in RDF and query it using SPARQL. The technological arsenal of semantic web is offering also a wealth of tools and systems for querying, publishing, and linking geospatial information represented in RDF.

2.1 The RDF framework

The Resource Description Framework[1] (RDF) was originally designed as a metadata data model and has become a standard model for data interchange on the web. RDF extends the linking structure of the Web to use URIs to name the relationship between things. This simple model allows structured and semi-structured data to be mixed, exposed, and shared across different applications. In RDF, each resource is described using triples. A triple consists of three elements, the `subject`, the `predicate` and the `object`. The `subject` denotes the resource, and the `predicate` denotes traits or aspects of the resource and expresses a relationship between the `subject` and the `object`. RDF is an abstract model with several serialization formats, and so the particular way in which a resource or triple is encoded varies from format to format. A collection of RDF triples represents a labeled, directed multi-graph, that is known as an RDF graph.

2.2 Vocabularies and Ontologies

On the Semantic Web, vocabularies define the concepts and relationships used to describe and represent an area of concern. Vocabularies are used to classify the terms that can be used in a particular application, characterize possible relationships, and define possible constraints on using those terms. There is no clear division between what is referred to as “vocabularies” and “ontologies”. The trend is to use the word *ontology*

[1] http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/

for more complex, and possibly quite formal collection of terms, whereas *vocabulary* is used when such strict formalism is not necessarily used or only in a very loose sense. Vocabularies are the basic building blocks for inference techniques on the Semantic Web.

The role of vocabularies on the Semantic Web are to help data integration and knowledge organization. Libraries, museums, newspapers, government portals, enterprises, social networking applications, and other communities that manage large collections of books, historical artifacts, news reports, business glossaries, blog entries, and other items can now use vocabularies, using standard formalisms, to leverage the power of linked data.

To satisfy different needs on the complexity of vocabularies, W3C offers a large palette of techniques to describe and define different forms of vocabularies in a standard format. These include RDF and RDF Schemas[2], Simple Knowledge Organization System[3] (SKOS), Web Ontology Language[4] (OWL), and the Rule Interchange Format[5] (RIF). The choice among these different technologies depend on the complexity and rigor required by a specific application.

2.3 The RDF query language SPARQL

RDF is a directed, labeled graph data format for representing information in the Web. This specification defines the syntax and semantics of the SPARQL[6] query language for RDF. SPARQL can be used to express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF via middleware. SPARQL contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions. SPARQL also supports aggregation, subqueries, negation, creating values by expressions, extensible value testing, and constraining queries by source RDF graph. The results of SPARQL queries can be result sets or RDF graphs.

Most forms of SPARQL query contain a set of triple patterns called a basic graph pattern. Triple patterns are like RDF triples except that each of the subject, predicate and object may be a variable. A basic graph pattern matches a subgraph of the RDF data when RDF terms from that subgraph may be substituted for the variables and the result is RDF graph equivalent to the subgraph. The example below shows a SPARQL query to find the title of a book from the given data graph. The query consists of two parts: the

[2] `http://www.w3.org/standards/techs/rdf`
[3] `http://www.w3.org/standards/techs/skos`
[4] `http://www.w3.org/standards/techs/owl`
[5] `http://www.w3.org/standards/techs/rif`
[6] `http://www.w3.org/TR/sparql11-query/`

`SELECT` clause identifies the variables to appear in the query results, and the `WHERE` clause provides the basic graph pattern to match against the data graph. The basic graph pattern in this example consists of a single triple pattern with a single variable `?title` in the object position.

```
DATA:
<http://example.org/book/book1> <http://purl.org/dc/elements/1.1/title> "SPARQL Tutorial" .

QUERY:
SELECT ?title
WHERE
{
  <http://example.org/book/book1> <http://purl.org/dc/elements/1.1/title> ?title .
}

RESULTS:
title
-----------------
"SPARQL Tutorial"
```

2.4 A Geographic Query Language for RDF Data

The OGC GeoSPARQL [29] standard supports representing and querying geospatial data on the Semantic Web. GeoSPARQL defines a vocabulary for representing geospatial data in RDF, and it defines an extension to the SPARQL query language for processing geospatial data. In addition, GeoSPARQL is designed to accommodate systems based on qualitative spatial reasoning and systems based on quantitative spatial computations.

The GeoSPARQL standard follows a modular design; it comprises several different components:

- A *core* component defines top-level RDFS/OWL classes for spatial objects.
- A *topology vocabulary* component defines RDF properties for asserting and querying topological relations between spatial objects.
- A *geometry* component defines RDFS data types for serializing geometry data, geometry-related RDF properties, and non-topological spatial query functions for geometry objects.

- A *geometry topology* component defines topological query functions.
- An *RDFS entailment* component defines a mechanism for matching implicit RDF triples that are derived based on RDF and RDFS semantics.
- A *query rewrite* component defines rules for transforming a simple triple pattern that tests a topological relation between two features into an equivalent query involving concrete geometries and topological query functions.

Each of the components described above forms a requirements class for GeoSPARQL. Implementations can provide various levels of functionality by choosing which requirements classes to support. For example, a system based purely on qualitative spatial reasoning may support only the *core* and *topological vocabulary* components.

In addition, GeoSPARQL is designed to accommodate systems based on qualitative spatial reasoning and systems based on quantitative spatial computations. Systems based on qualitative spatial reasoning, do not usually model explicit geometries, so queries in such systems will likely test for binary spatial relationships between features rather than between explicit geometries. To allow queries for spatial relations between features in quantitative systems, GeoSPARQL defines a series of query transformation rules that expand a feature-only query into a geometry-based query. With these transformation rules, queries about spatial relations between features will have the same specification in both qualitative systems and quantitative systems. The qualitative system will likely evaluate the query with a backward-chaining spatial “reasoner”, and the quantitative system can transform the query into a geometry-based query that can be evaluated with computational geometry.

2.5 Well-known text Serialization

Well-known text (WKT) [2] is a text markup language for representing vector geometry objects on a map, spatial reference systems of spatial objects and transformations between spatial reference systems. A binary equivalent, known as well-known binary (WKB), is used to transfer and store the same information on databases, such as PostGIS, Microsoft SQL Server and DB2. The formats were originally defined by the Open Geospatial Consortium (OGC) and described in their Simple Feature Access and Coordinate Transformation Service specifications. The current standard definition is in the ISO/IEC 13249-3:2011 standard, "Information technology -- Database languages -- SQL multimedia and application packages -- Part 3: Spatial" (SQL/MM).

WKT can represent 18 distinct geometric objects:

- Geometry
- Point, MultiPoint
- LineString, MultiLineString
- Polygon, MultiPolygon, Triangle
- CircularString
- Curve, MultiCurve, CompoundCurve
- CurvePolygon
- Surface, MultiSurface, PolyhedralSurface
- TIN
- GeometryCollection

Coordinates for geometries may be 2D (x, y), 3D (x, y, z), 4D (x, y, z, m) with an m value that is part of a linear referencing system or 2D with an m value (x, y, m). Three-dimensional geometries are designated by a `Z` after the geometry type and geometries with a linear referencing system have an `M` after the geometry type. Empty geometries which contain no coordinates can be specified by using the symbol `EMPTY` after the type name. WKT geometries are used throughout OGC specifications and are present in applications that implement these specifications. For example, PostGIS contains functions that can convert geometries to and from a WKT representation, making them human readable. The following are some examples of geometric WKT strings:

```
GEOMETRYCOLLECTION(POINT(4 6),LINESTRING(4 6,7 10))
POINT ZM (1 1 5 60)
POINT M (1 1 80)
POINT EMPTY
MULTIPOLYGON EMPTY
CIRCULARSTRING(1 5, 6 2, 7 3)
COMPOUNDCURVE(CIRCULARSTRING(0 0,1 1,1 0),(1 0,0 1))
MULTICURVE((5 5,3 5,3 3,0 3),CIRCULARSTRING(0 0,2 1,2 2))
TRIANGLE((0 0 0,0 1 0,1 1 0,0 0 0))
TIN (((0 0 0, 0 0 1, 0 1 0, 0 0 0)), ((0 0 0, 0 1 0, 1 1 0, 0 0 0)))
```

```
POLYHEDRALSURFACE Z (
    ((0 0 0, 0 1 0, 1 1 0, 1 0 0, 0 0 0)),
    ((0 0 0, 0 1 0, 0 1 1, 0 0 1, 0 0 0)),
    ((0 0 0, 1 0 0, 1 0 1, 0 0 1, 0 0 0)),
    ((1 1 1, 1 0 1, 0 0 1, 0 1 1, 1 1 1)),
    ((1 1 1, 1 0 1, 1 0 0, 1 1 0, 1 1 1)),
    ((1 1 1, 1 1 0, 0 1 0, 0 1 1, 1 1 1))
  )
```

2.6 Geography Markup Language

The OpenGIS® Geography Markup Language (GML)[1] is an XML grammar for expressing geographical features. GML serves as a modeling language for geographic systems as well as an open interchange format for geographic transactions on the Internet. As with most XML based grammars, there are two parts to the grammar – the schema that describes the document and the instance document that contains the actual data. A GML document is described using a GML Schema. This allows users and developers to describe generic geographic data sets that contain points, lines and polygons. However, the developers of GML envision communities working to define community-specific application schemas[7] that are specialized extensions of GML. Using application schemas, users can refer to roads, highways, and bridges instead of points, lines and polygons. If everyone in a community agrees to use the same schemas they can exchange data easily and be sure that a road is still a road when they view it. GML contains a rich set of primitives which are used to build application specific schemas or application languages. These primitives include:

- Feature
- Geometry
- Coordinate reference system
- Topology
- Time
- Dynamic feature
- Coverage (including geographic images)
- Unit of measure
- Directions

[7] https://en.wikipedia.org/wiki/GML_Application_Schemas

- Observations
- Map presentation styling rules

The original GML model was based on the World Wide Web Consortium's Resource Description Framework (RDF). Subsequently, the OGC introduced XML schemas into GML's structure to help connect the various existing geographic databases, whose relational structure XML schemas more easily define. The resulting XML-schema-based GML retains many features of RDF, including the idea of child elements as properties of the parent object (RDFS) and the use of remote property references.

2.6.1 GML Feature

GML defines features distinct from geometry objects. A feature is an application object that represents a physical entity, e.g. a building, a river, or a person. A feature may or may not have geometric aspects. A geometry object defines a location or region instead of a physical entity, and hence is different from a feature. In GML, a feature can have various geometry properties that describe geometric aspects or characteristics of the feature (e.g. the feature's Point or Extent properties). GML also provides the ability for features to share a geometry property with one another by using a remote property reference on the shared geometry property. Remote properties are a general feature of GML borrowed from RDF. An `xlink:href` attribute on a GML geometry property means that the value of the property is the resource referenced in the link. For example, a Building feature in a particular GML application schema might have a position given by the primitive GML geometry object type Point. However, the Building is a separate entity from the Point that defines its position. In addition, a feature may have several geometry properties (or none at all), for example an extent and a position.

2.6.2 Geometry Properties

Application-specific names shall be chosen for geometry properties in GML application schemas. The names of the properties should be chosen to express the semantics of the value. Using application specific names is the preferred method for names of properties including geometry properties. There are no inherent restrictions in the type of geometry property a feature type may have as long as the property value is a geometry object substitutable for `gml:AbstractGeometry`. The GML schema includes the following predefined property types that may be used as types of geometry property element:

- PointPropertyType
- CurvePropertyType

- SurfacePropertyType
- SolidPropertyType
- MultiPointPropertyType
- MultiCurvePropertyType
- MultiSurfacePropertyType
- MultiSolidPropertyType
- MultiGeometryPropertyType
- PointArrayPropertyType
- CurveArrayPropertyType
- SurfaceArrayPropertyType
- SolidArrayPropertyType

The following GML example illustrates the distinction between features and geometry objects. The Building feature has several geometry objects, sharing one of them (the Point with identifier p21) with the SurveyMonument feature:

```
<abc:Building gml:id="SearsTower">
     <abc:height>52</abc:height>
     <abc:position xlink:type="Simple" xlink:href="#p21"/>
 </abc:Building>
 <abc:SurveyMonument gml:id="g234">
     <abc:position>
         <gml:Point gml:id="p21">
             <gml:posList>100,200</gml:posList>
         </gml:Point>
     </abc:position>
 </abc:SurveyMonument>
```

2.7 Temporal RDF

The introduction of time in data models and query languages has been the subject of extensive research in the field of relational databases [10, 34]. Three distinct kinds of time were introduced and studied: *user-defined* time which has no special semantics (e.g., January 1st, 1963 when John has his birthday), *valid* time which is the time an event takes place or a fact is true in the application domain (e.g., the time 2000-2012 when John is a professor) and *transaction* time which is the time when a fact is current in the database (e.g., the system time that gives the exact period when the tuple representing that John is a professor from 2000 to 2012 is current in the database). Compared to the relational database case, little research has been done to extend the RDF data model and the query language SPARQL with temporal features. Gutierrez et al. [13, 14] were the first to propose a formal extension of the RDF data model with valid time support. They also introduce the concept of *anonymous timestamps* in general temporal RDF graphs, i.e., graphs containing quads of the form $(s, p, o)[t]$ where t is a timestamp or an anonymous timestamp x stating that the triple (s, p, o) is valid in some unknown time point x.

A formal extension of RDF, called stRDF, and the corresponding query language stSPARQL for the representation and querying of temporal and spatial data using linear constraints is presented in [20]. stRDF and stSPARQL were later redefined in [22] so that geometries are represented using the Open Geospatial Consortium standards Well-Known-Text (WKT) and Geography Markup Language (GML). Although the valid time dimension of stRDF and stSPARQL is in the spirit of [32], it is introduced in a language with a much more mature geospatial component based on OGC standards [22]. In addition, the valid time component of stSPARQL offers a richer set of functions for querying valid times than the ones in [32]. With the temporal dimension, stSPARQL also becomes more expressive than the OGC standard GeoSPARQL [3]. While stSPARQL can represent and query geospatial data that changes over time, GeoSPARQL only supports static geospatial data.

2.7.1 stRDF

The model stRDF [21, 8] is an extension of RDF for representing and querying geospatial data that changes over time. Two kinds of time primitives are supported: time instants and time periods. A *time instant* is an element of the time line. A *time period* (or simply period) is an expression of the form $[B, E)$, $(B, E]$, (B, E), or $[B, E]$ where B and E are time instants called the *beginning* and the *ending* of the period respectively. Since the time line is discrete, we often assume only periods of the form $[B, E)$ with no loss of generality. Syntactically, time periods are represented by literals of the new datatype `strdf:period` that was introduce in stRDF. The value space of `strdf:period` is the set of all time periods covered by the above definition. The lexical space of `strdf:period` is trivially defined from the lexical space of `xsd:dateTime` and the closed/open period notation introduced above. Time instants can also be represented as closed periods with the same beginning and ending time. Values of the datatype `strdf:period` can be used as objects of a triple to represent *user-defined time*. In addition, they can be used to represent *valid times* of temporal triples which are defined as follows. A *temporal triple (quad)* is an expression of the form `s p o t.` where `s p o.` is an RDF triple and `t` is a time instant or a time period called the *valid time* of a triple. An *stRDF graph* is a set of triples and temporal triples. In other words, some triples in an stRDF graph might not be associated with a valid time.

2.7.2 stSPARQL

The query language stSPARQL [17, 21, 8] is an extension of SPARQL 1.1. For the valid time dimension stSPARQL provides the following features:

Temporal Triple Patterns. Temporal triple patterns are introduced as the most basic way of querying temporal triples. A *temporal triple pattern* is an expression of the form `s p o t.`, as explained in the previous section.

Temporal Extension Functions. Temporal extension functions are defined in order to express temporal relations between expressions that evaluate values of the datatypes `xsd:dateTime` and `strdf:period`. The first set of such temporal functions are 13 Boolean functions that correspond to the 13 binary relations of Allen's Interval Algebra. stSPARQL offers nine functions that are "syntactic sugar" i.e., they encode frequently-used disjunctions of these relations. There are also three functions that allow relating an instant with a period:

- `xsd:Boolean strdf:during(xsd:dateTime i2, strdf:period p1)`: returns true if instant `i2` is during the period `p1`.
- `xsd:Boolean strdf:before(xsd:dateTime i2, strdf:period p1)`: returns true if instant `i2` is before the period `p1`.
- `xsd:Boolean strdf:after(xsd:dateTime i2, strdf:period p1)`: returns true if instant `i2` is after the period `p1`.

Furthermore, stSPARQL offers a set of functions that construct new (closed-open) periods from existing ones. There are also the functions `strdf:period_start` and `strdf:period_end` that take as input a period `p` and return an output of type `xsd:dateTime` which is the beginning and ending time of the period `p` respectively. Finally, stSPARQL defines the following functions that compute temporal aggregates:

- `strdf:period strdf:intersectAll(set of period p)`: Returns a period that is the intersection of the elements of the input set that have a common intersection.
- `strdf:period strdf:maximalPeriod(set of period p)`: Constructs a period that begins with the smallest beginning point and ends with the maximum endpoint of the set of periods given as input.

2.8 Strabon, a spatiotemporal RDF store

Strabon [21] is a DBMS for the storage and querying of linked spatiotemporal data encoded in the model stRDF and queried by stSPARQL or GeoSPARQL. The expressive power of stSPARQL makes Strabon the only fully implemented RDF store with rich spatial and temporal functionalities available today. Strabon supports spatial datatypes enabling the serialization of geometric objects in OGC standards WKT and GML. It also offers spatial and temporal selections, spatial and temporal joins, a rich set of spatial functions similar to those offered by geospatial relational database systems and support for multiple Coordinate Reference Systems. Strabon can be used to model temporal domains and concepts such as events, facts that change over time etc. through its support for valid time of triples, and a rich set of temporal functions. Strabon is an open source system and has been developed by extending the well-known RDF store Sesame[8] and using PostGIS[9] and PostgreSQL Temporal[10] for the back-end storage and querying of spatiotemporal data. As shown in [21, 8], Strabon is currently the most functional spatiotemporal RDF store available, while at the same time it offers equal or better performance when compared with other RDF stores such as Parliament [6], uSeekM [31], and well-known commercial systems [12].

2.9 Geospatial Technologies

Geospatial technologies is a term used to describe the range of modern tools contributing to the geographic mapping and analysis of the Earth and human societies. Satellites allowed images of the Earth's surface and human activities therein with certain limitations. Computers allowed storage and transfer of imagery together with the development of associated digital software, maps, and data sets on socioeconomic and environmental phenomena, collectively called geographic information systems (GIS). An important aspect of a GIS is its ability to assemble the range of geospatial data into a layered set of maps which allow complex themes to be analyzed and then communicated to wider audiences. This *layering* is enabled by the fact that all such data includes information on its precise location on the surface of the Earth, hence the term *geospatial*.

Especially in the last decade, these technologies have evolved into a network of national security, scientific, and commercially operated satellites complemented by powerful desktop GIS. In addition, aerial remote sensing platforms, including unmanned aerial vehicles, are seeing increased non-military use as well. High quality hardware and data is now available to new audiences such as universities, corporations, and non-governmental organizations. The fields and sectors deploying these technologies are currently growing at a rapid pace, informing decision makers on topics such as industrial engineering, biodiversity conservation, forest fire suppression, agricultural monitoring, humanitarian relief, and much more.

There are now a variety of types of geospatial technologies potentially applicable to human rights, including the following:

- *Remote Sensing*: imagery and data collected from space or airborne camera and sensor

[8] `http://rdf4j.org`
[9] `http://postgis.net`
[10] `http://pgfoundry.org/projects/temporal/`

platforms. Some commercial satellite image providers now offer images showing details of one-meter or smaller, making these images appropriate for monitoring humanitarian needs and human rights abuses.

- *Geographic Information Systems (GIS)*: a suite of software tools for mapping and analyzing data which is georeferenced (assigned a specific location on the surface of the Earth, otherwise known as geospatial data). GIS can be used to detect geographic patterns in other data, such as disease clusters resulting from toxins, sub-optimal water access, etc.
- *Global Positioning System (GPS)*: is a space-based satellite navigation system that provides location and time information in all weather conditions, anywhere on or near the earth where there is an unobstructed line of sight to four or more GPS satellites. The system provides critical capabilities to military, civil, and commercial users around the world. The United States government created the system, maintains it, and makes it freely accessible to anyone with a GPS receiver. A similar European system called Galileo will be operational within the next several years while a Russian system is functioning but restricted.
- *Digital Mapping Technologies*: software programs like Google Earth and web features like Microsoft Virtual Earth are changing the way geospatial data is viewed and shared. The developments in user interface are also making such technologies available to a wider audience whereas traditional GIS has been reserved for specialists and those who invest time in learning complex software programs.

2.10 Digital Mapping Technologies

Among the earliest maps to appear on the World Wide Web was the Xerox Palo Alto Research Center (PARC) Map Viewer in 1993. This interactive world map enabled users to toggle the display of national boundaries and rivers, change scale, change projection and add placemarks. In 1995, a researcher at UC-Berkeley developed GRASSLinks[11], a web interface on top of GRASS, an open source GIS package originally developed by the US Army Corps of Engineers. GRASSLinks made it possible to view any GRASS dataset, pan and zoom, and, importantly, click on the map to obtain information about the clicked location. By 1997, the U.S. Census Bureau had developed a web interface for its enormously rich TIGER dataset called the TIGER Map Server. The TIGER Map Server made it possible to toggle on/off many of the geographic entities in the dataset. Finally, the mid-90s saw the first interactive mapping site aimed at consumers, MapQuest[12], launched in 1996. MapQuest provided turn-by-turn driving directions.

Each of these sites is part of what BYU[13] researcher Brandon Plewe calls the 1st generation in

[11] `http://pnwpest.org/glinks/`
[12] `http://www.mapquest.com`
[13] `http://home.byu.edu/home/`

the evolution of web mapping. They are characterized by simple HTML protocols and mostly static maps. These 1st generation sites were common during the period 1993-1999. A 2nd generation of web mapping applications took advantage of emerging technologies like Dynamic HTML, Java, and ActiveX to produce sites with greater interactivity and performance. This so-called *WebGIS* era saw GIS vendors develop server-based software (such as ESRI's ArcIMS and Intergraph's GeoMedia Web Map) so that their clients (particularly public sector agencies) could put their geographic data online. This generation of sites represented the state of the art in 1995 through 2004. While it was certainly better to have map data made publicly available through these sites than not, they suffered from two main flaws: complicated user interfaces that were intimidating to the general public, and slow performance.

The next major development in web mapping in 2005 brought significant improvements in performance. A new technology called Ajax (Asynchronous JavaScript and XML) enabled web developers to finally develop sites (both mapping and non-mapping) that responded more like desktop applications than the "click-and-wait" applications of the past. This 3rd generation of web mapping is best exemplified by the site that first took advantage of Ajax programming techniques — Google Maps[14]. Other online mapping companies quickly adopted the technology as well, most notably Yahoo! and MapQuest. ESRI improved on its web server software by incorporating Ajax into its ArcGIS[15] Server product.

Plewe also identifies a 4th generation of web mapping. This generation is characterized by applications that strive for greater realism in our representations of the world through the use of 3D globes and immersive environments. Google is also a pioneer in this generation with its Google Earth[16] application. Other popular globe technologies include Microsoft's Virtual Earth[17] and NASA's WorldWind[18].

2.11 Visualizing Geospatial Data

Advancements in the management of linked geospatial data has been followed and reinforced by many efforts which have jumped on the LOD bandwagon by populating the LOD cloud with geospatial information. In this section we present the actual status of the visualization of geospatial data approaches.

[14] https://maps.google.com/
[15] http://www.esri.com/software/arcgis
[16] https://earth.google.com/
[17] http://microsoft-virtual-earth-3d.en.uptodown.com
[18] http://worldwind.arc.nasa.gov/java/

2.11.1 LDVizWiz

With the goal of providing general purpose visualizations of any SPARQL endpoint, LDVizWiz[4] inspects the features of the dataset in order to understand the underlying data and detect categories. Based on the classification provided by Shneiderman, LDVizWiz performs ASK queries to categorize data in one of the following classes: Geography, Temporal, Event, Agent/Person, Organization, Statistics and Knowledge. This non-datatype based approach loses some of the advantages provided by best practices and lessons learned from the data visualization field, although class-category template filling exhibits a more robust performance if the schema is known beforehand. The biggest drawback is the need to adapt and extend the ASK queries to new vocabularies whenever they are detected. This tool was one of the first attempts to visualize RDF data and is focused in visualizing only the ontology scheme in a SPARQL endpoint, thus it cannot be used to create layers over the existing data by combining specific features from the dataset and can only represent data from one SPARQL endpoint at a time.

2.11.2 LinkedGeoData Browser

The LinkedGeoData Browser[19] and Editor allows to browse the world by using a slippy map. Once a region is selected, the browser analyzes the descriptions of nodes and ways in that region and generates facets for filtering. Once a facet or a specific facet value has been selected, matching elements are displayed as markers on the map and in a list. If the selected region is changed, these are updated accordingly. If a user logs into the application by using her Open Street Map (OSM) credentials, the displayed elements can directly be edited in the map view. For this, the browser generates a dynamic form based on existing properties. The form also allows to add arbitrary additional properties. In oder to encourage reuse of both properties and property values, the editor performs a type-ahead search for existing properties and property values and ranks them according to the usage frequency. When changes are made, these are stored locally and propagated to the main OSM database by using the OSM API. This is one of the most promising tools, but offers browsing functionality only for OSM data. We would like to use all these features over any existing SPARQL endpoint so that we can combine different datasets in one map.

2.11.3 Map4RDF

Map4RDF[20] [25] is another browser for RDF data with geospatial information that supports faceted browsing of RDF data based on the `rdf:type` property, filtering of query results in a specified bounding box, and visualization of statistical data modeled according to the SCOVO vocabulary[21] using predefined statistical indices. Although this tool can be used to explore data in a SPARQL

[19] http://linkedgeodata.org/LGD%20Browser
[20] http://oeg-dev.dia.fi.upm.es/map4rdf/
[21] http://vocab.deri.ie/scovo

endpoint and provide statistical charts for them, we cannot create layers by combining data from different classes or write SPARQL queries to explore and visualize our dataset and also we can explore one SPARQL endpoint at a time, thus we cannot combine data from various datasets in one map.

2.11.4 Facete Browser

Facete browser[22] offers faceted search over SPARQL endpoints by filtering RDF data by any property and providing hierarchical facets and links from facets to table columns and map view. At the moment facete always queries for the exact facet/facet value counts. Therefore, Facete may not work with low selective facet selections on large datasets. This tool is a more sophisticated version of Map4RDF. With Facete we filter the results of a class by choosing specific properties, or by using the hierarchical structure of the classes. The main disadvantages again are the lack of exploration with SPARQL queries and the ability to combine different data sources in on map.

2.11.5 LODVisualization

The LODVisualization[23] tool is based on the Linked Data Visualization Model[11] for visualizing RDF data. These visualizations allow users to obtain an overview of RDF datasets and realize what the data is about: their main types, properties, etc. LODVisualization is compatible with most of SPARQL endpoints as long as they support JSON and SPARQL 1.1. This tool overcomes the severe limitations of all the above tools when they are faced with the task of exploring the linked geospatial data cloud. Although LODVisualization seems a very promising tool, it has very limited support regarding visualization of geospatial data and construction of meaningful thematic maps.

[22] `https://github.com/GeoKnow/Facete`
[23] `http://lodvisualization.appspot.com`

Chapter 3
Sextant: Visualizing and Exploring Time-Evolving Linked Geospatial Data

Advancements in the management of linked geospatial data has been followed and reinforced by many efforts which have jumped on the linked open data (LOD) bandwagon by populating the LOD cloud with geospatial information. Although applications for exploiting this abundance of geospatial information have also started to emerge, the most promising ones, that were presented in the previous chapter, are either focused in the visualization and exploration of the ontology scheme, or provide tools for visualization and exploration of a single SPARQL endpoint though extensive use of SPARQL from the user point of view. Moreover, none of the existing tools provide functionality for utilization of the temporal dimension of data. In this thesis we try to remedy the shortcomings of the existing tools and the original version of Sextant and focus on creating a user-friendly application enhanced with features, which exist in the matured Geographic Information Systems, allowing both experts and non-experts to explore and visualize linked geospatial and temporal data.

3.1 Sextant, a new approach

The core feature of Sextant is the ability to create thematic maps by combining geospatial and temporal information that exists in a number of heterogeneous data sources ranging from standard SPARQL endpoints, to SPARQL endpoints following the standard GeoSPARQL defined by the Open Geospatial Consortium (OGC), or well-adopted geospatial file formats, like KML, GML and GeoTIFF. In this manner we overcome the main disadvantage of the existing tools that allows the visualization of a single SPARQL endpoint, and provide functionality to domain experts from different fields in creating thematic maps, which emphasize spatial variation of one or a small number of geographic distributions. Moreover we go beyond and present a map ontology that assists on modeling these maps in RDF and allow for easy sharing, editing and search mechanisms over existing maps.

A lot of effort was put the past years in Geographic Information Systems that resulted in creating a rich technological arsenal for the traditional GIS area. Many different GIS file formats were created by government mapping agencies or software developers and have become standards in encoding geographical information. Two are the main categories of these file formats according to the way the information is represented in the files. These are the raster and the vector file formats. It is crucial for any visualization tool to incorporate these file formats and provide ways to visualize them as layers on the map and interact with them. Sextant achieves that by supporting the visualization of KML, GML and GeoTIFF, that are the most promising file formats, and also provide tools for interaction with these layers, such as the colorization of geometry features according to specific values to create color maps for better understanding of the various aspects of layers.

Another important feature is the utilization of the temporal dimension. Implementation of he valid time component of stRDF and stSPARQL in system Strabon allows us to query both the spatial and the temporal dimension. Enriching our results with temporal information allows us to create layers with valid time. Using the SIMILE Timeline widget we can make these layers appear and disappear from the map according to their valid time. This feature allows the creation of thematic maps that change over time and can assist experts in the fields of agriculture, biodiversity, climate, disasters, ecosystems, energy, water and weather, in visualizing temporal maps that help them understand the evolution of data.

Apart from visualizing the spatial and temporal dimension, statistical charts play an important role in understanding the various measures of datasets. Statistical data is a foundation for policy prediction, planning and adjustments and underpins many of the mash-ups and visualizations we see on the web. There is strong interest in being able to publish statistical data in a web-friendly format to enable it to be linked and combined with related information. At the heart of a statistical dataset is a set of observed values organized along a group of dimensions, together with associated metadata. The Data Cube vocabulary enables such information to be represented using the W3C RDF standard and published following the principles of linked data. We demonstrate how to utilize the Data Cube vocabulary to enhance existing datasets and allow the creation of charts through Sextant in an intuitive way that does not involve the use of SPARQL from the user point of view.

3.2 Architecture of Sextant

Sextant is a web-based and mobile ready application for exploring, interacting and visualizing time-evolving linked geospatial data. What we wanted to achieve is develop an application that is flexible, portable and interoperable with other GIS tools.

3.2.1 Client

Sextant is build using a client-server architecture model as depicted in Figure 3.1. We used Bootstrap framework[1] to implement a responsive single code base user interface (UI), that is used in both the web application and the mobile application. For map rendering we decided to use OpenLayers 2.13[2] JavaScript Mapping Library, that enables us to run the application in a local

[1] http://getbootstrap.com/
[2] http://openlayers.org/two/

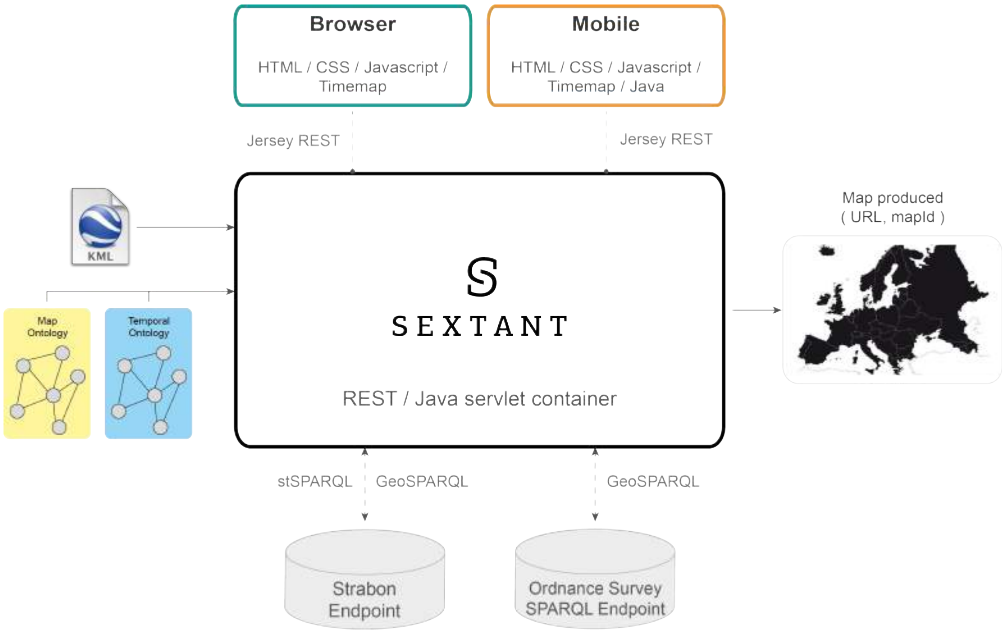


Figure 3.1: High-level architecture of Sextant

environment and supports a great variety of GIS file formats. To handle the evolution of geospatial data through time, we use Timemap JavaScript library[3], that integrates the SIMILE Timeline widget[4] with OpenLayers, to allow visualization of KML files enriched with temporal information. A lot of effort was put in designing a user-friendly and flexible environment for the end-user, that is developed using HTML 5[5], CSS[6] and JavaScript technologies.

In Figure 3.2 we show the main menu of Sextant and the functionalities supported through it. From the first tab, users can create layers using existing KML, GML or GeoTIFF files, and also save or load maps. The second tab supports the basic functions such as hiding and showing all layers, zoom to map extent and show the color panel. The third tab enables us to communicate with SPARQL endpoints, pose queries and visualize the results on the map, select predefined queries to visualize and search for maps. In the forth tab we have the functionalities for statistical information extraction, where we can create charts and see all created charts. In the Android version, the menu bar has one more tab as depicted in Figure 3.3, that allows us to provide the URI for the server back-end that we want to use.

The user interface of Sextant is depicted in Figure 3.4. On top we have the menu bar we presented. The main window is the map with the zoom buttons and the two buttons that allow the users to add points of interest (POIs) and areas of interest(AOIs) on the map. To the right we have the main panels that we use to show metadata and handle layers and the timeline. Using

[3]`https://code.google.com/p/timemap/`
[4]`http://www.simile-widgets.org/timeline/`
[5]`http://www.w3.org/TR/html5/`
[6]`http://www.w3.org/Style/CSS/`

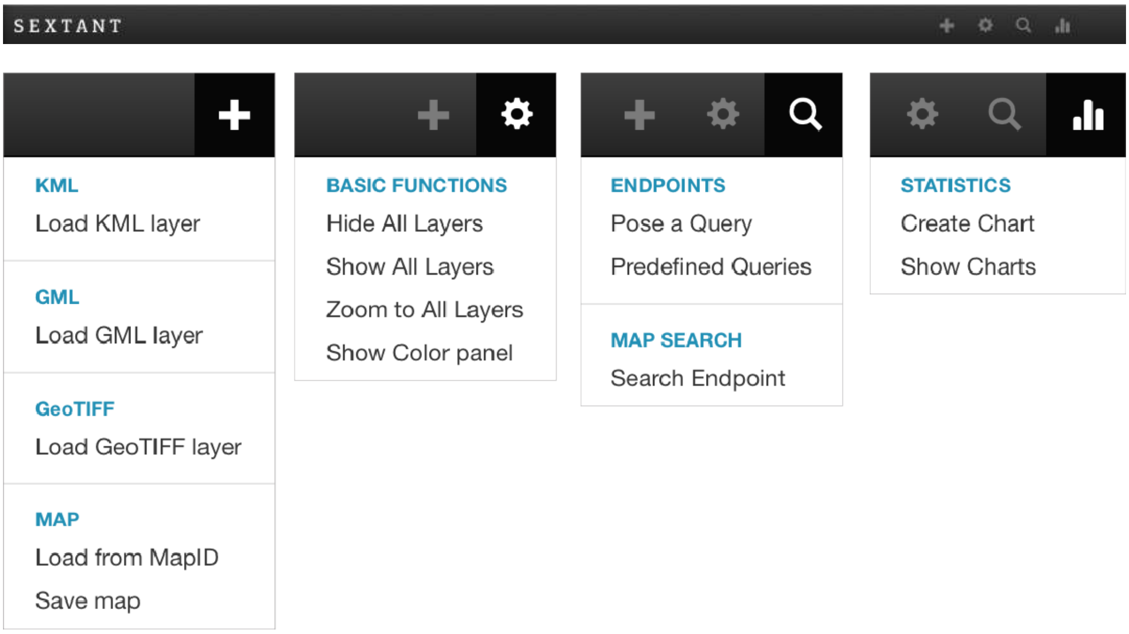


Figure 3.2: Sextant main menu

the red and blue bookmarks in the edge of the panels we can show or hide the Layers and Map Information panel and the timeline respectively.

3.2.2 Server

The server is build in Java 7 and is composed of the endpoint connector, the ontology manager, the map registry and the KML translator. The communication between the clients and the server is implemented using Jersey RESTful Web Services framework[7].

The endpoint connector is responsible for the communication between Sextant and various SPARQL endpoints that are publicly accessible on the web through a URI and are compliant with the specification documents of SPARQL 1.1 protocol and the SPARQL query results XML format.

As we described above, each map created with Sextant is represented using the map ontology. The ontology manager is the component responsible for maintaining the description of a map, and communicates with the map registry to store or retrieve the RDF description of a map.

The map registry is responsible for storing the RDF description of maps according to the map ontology. In Sextant we use Strabon Endpoints to act as map registries in order to offer search capabilities on spatial and temporal attributes.

Finally, the KML translator is used to translate the results of GeoSPARQL and stSPARQL queries from SPARQL query results XML format to the KML file format.

[7] https://jersey.java.net/

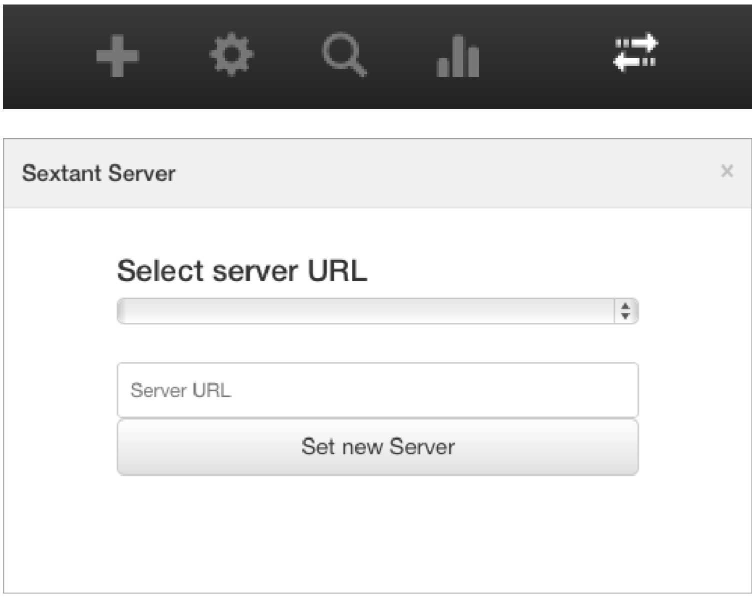
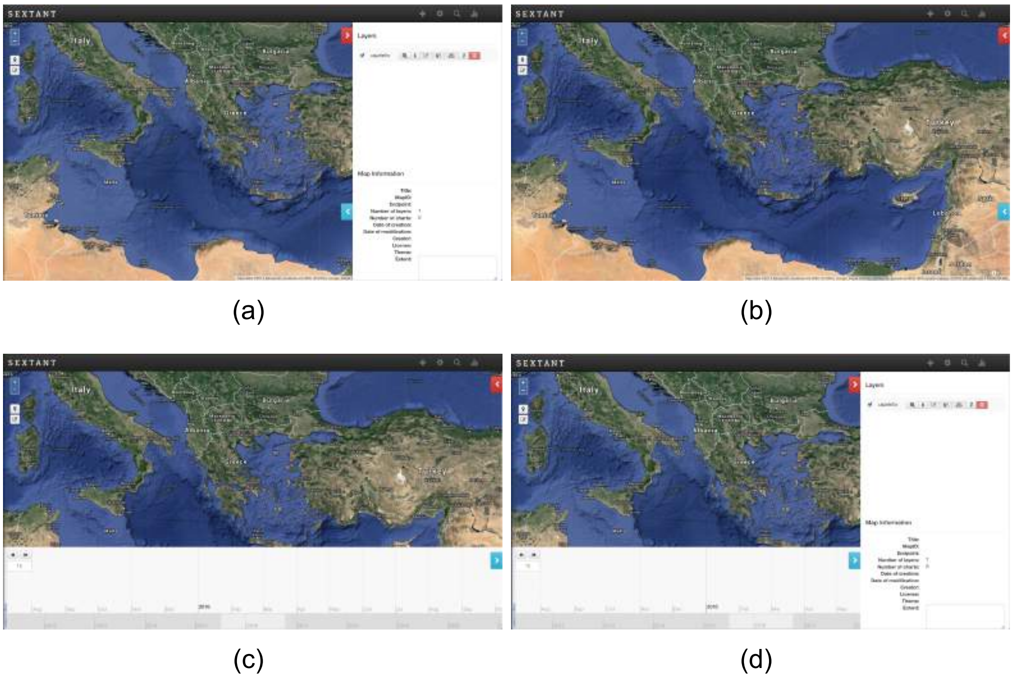


Figure 3.3: Sextant Android main menu extra tab

To achieve interoperability with other well-known GIS tools, Sextant is based on the OGC standards listed bellow:

- OpenGIS Simple Features Access (OGC-SFA) standard [2] that defines Well-Known Text (WKT) format for representing geometries.
- Geography Markup Language (GML) [1] for expressing geographical features.
- The KML file format [30] that was designed by Google and became on OGC standard in 2008, which is an XML notation for expressing geographic annotation and visualization on web-based and mobile two-dimensional maps or three-dimensional Earth browsers.
- The query language GeoSPARQL for representing and querying RDF data with geospatial information [29].



(a) (b) (c) (d)

Figure 3.4: Sextant User Interface(UI) (a), UI with hidden panels (b), UI with timeline (c) and UI with timeline and panels (d)

Chapter 4
Thematic Map

A thematic map is a type of map designed to show a particular theme connected with a specific geographic area. These maps can portray physical, social, political, cultural, economic, sociological, agricultural, or any other aspects of a city, state, region, nation, or continent.

Thematic maps emphasize spatial variation of one or a small number of geographic distributions. These distributions may be physical phenomena such as climate or human characteristics such as population density and health issues. While general reference maps show where something is in space, thematic maps tell a story about that place.

The core feature of Sextant is the ability to create thematic maps by combining geospatial and temporal information that exists in a number of heterogeneous data sources ranging from standard SPARQL endpoints, to SPARQL endpoints following the standard GeoSPARQL defined by the Open Geospatial Consortium (OGC), or well-adopted geospatial file formats, like KML. To create such maps we present the map ontology that helps to model our information into layers. Then we can save these maps in the form of RDF triples in a SPARQL endpoint creating a Map registry that allows for further data mining such as search mechanisms over the maps' metadata and an easy way for sharing the created maps.

4.1 Map Ontology

The base of this application's features is the creation, sharing, editing and searching for thematic maps. To do so we designed the map ontology of Figure 4.1(a) to model the content of a map. Each map consists of an ordered set of layers, the content of which may derive either from a standard file format for representing geospatial information, or from the evaluation of an stSPARQL or GeoSPARQL query to a SPARQL endpoint. stSPARQL is an extension of SPARQL 1.1 that can be used to query data represented in stRDF [22, 8], an extension of RDF, while GeoSPARQL is the Open Geospatial Consortium (OGC) standard for static geospatial data. Population of a map ontology results in a map that is also a web resource and can be shared for editing and viewing in Sextant. The map ontology also employs the temporal ontology depicted in Figure 4.1(c) that is dictated by the data model stRDF and the query language stSPARQL for modeling of valid time. This ontology enables the introduction of user-defined time and valid time of a triple in RDF data. User-defined time does not carry any special semantics. Valid time is the time an event takes place or a fact is true in the application domain. Time is modeled as instants or intervals and is represented using values of the datatypes `xsd:dateTime` and `strdf:period` respectively. One

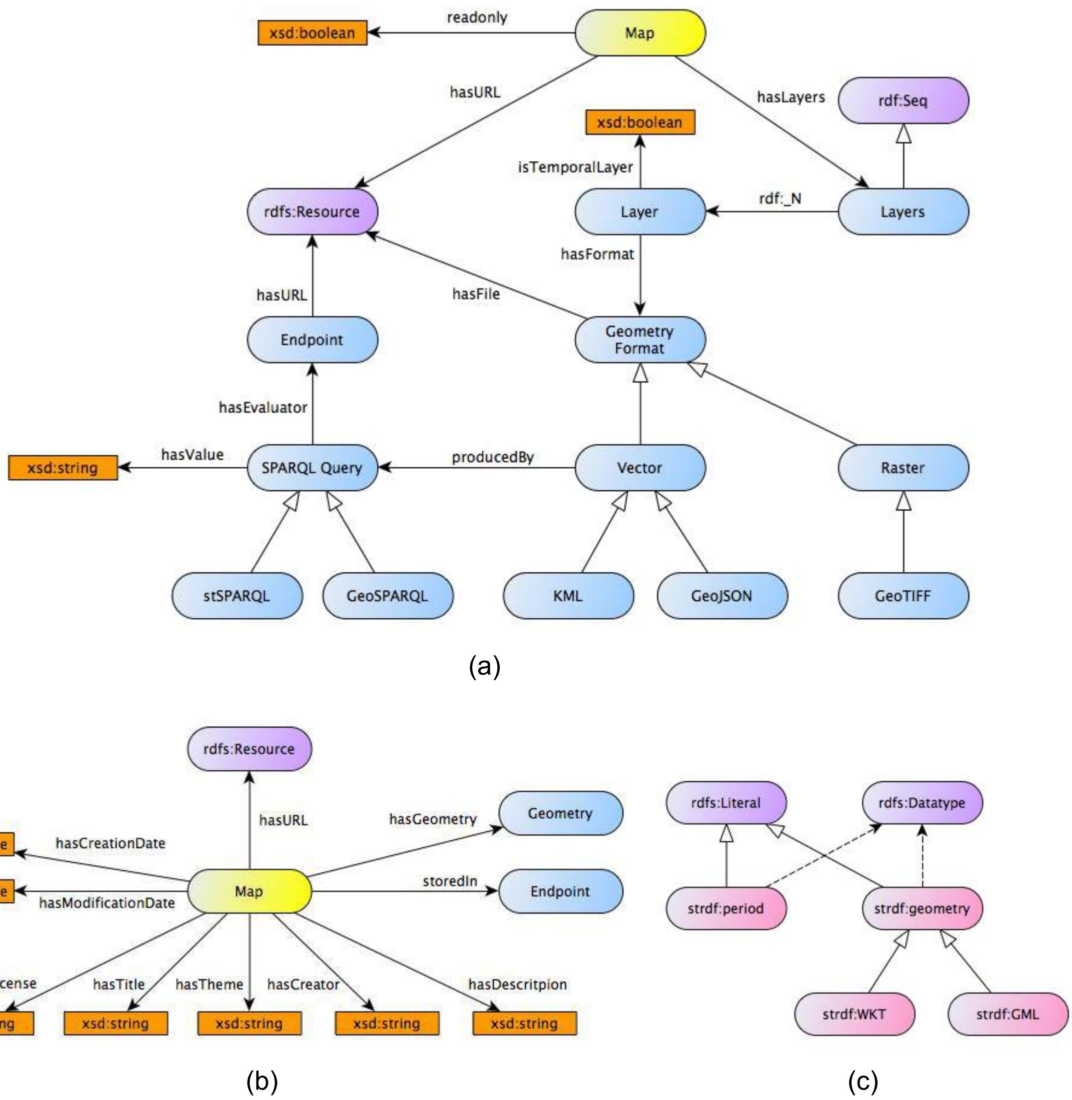


Figure 4.1: The Map Ontology (a), the Map Metadata (b) and the temporal Ontology (c)

may use all the temporal features of stSPARQL to query linked spatiotemporal data encoded in stRDF, such as the temporal extension functions corresponding to the thirteen temporal relations of Allen's interval algebra, the temporal aggregate functions, or the extension functions for constructing new temporal elements. In this way the full capabilities of SPARQL endpoints using the spatiotemporal RDF store Strabon can be exploited.

Each map also has a set of metadata as shown in Figure 4.1(b). These are the title of the map, the identifier that is used to share the map, the URI of the endpoint that it is saved in, the name of the creator, the license and the theme of the map, a small description for the content, the extent of the map that derives from the minimum bounty box of the geometries of the layers it consists of and two dates that specify the date of creation and last modification of the map. These metadata provide vital information for the map legend and are also used to fuel the search mechanism to allow users to search for maps that meet certain criteria.

4.2 Map Registry

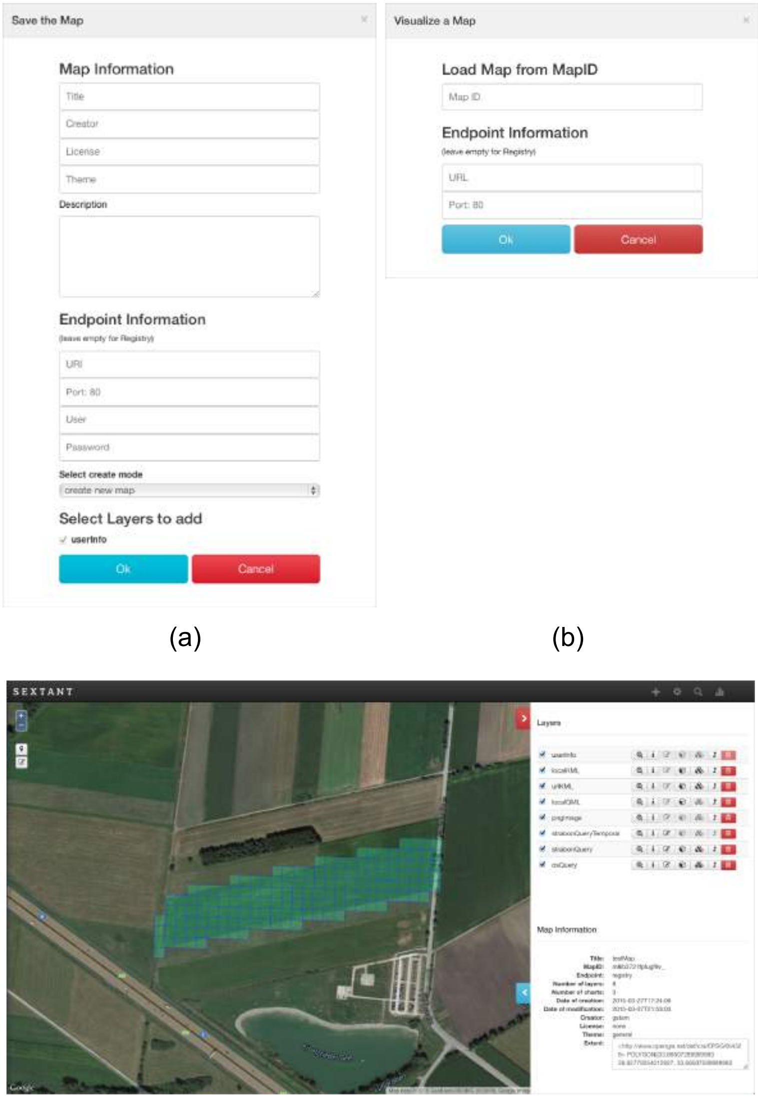
Apart from creating maps, we need a mechanism to save and load these maps. To accomplish that we can create map registries, that are SPARQL endpoints that hold all the map information and metadata to assist us in saving and retrieving the maps. As described above, each map is modeled with the map ontology and a set of RDF triples is produced that we can store using an RDF store. In this way we create map registries that hold all the needed information to reconstruct each map and thus provide a way to share and load the maps we create. Also as each registry is populated with maps and their metadata, we can use these registries to search for maps that meet our criteria and select one from the result list to load on the world map.

In Figure 4.2 we demonstrate how to save and load maps using Sextant. When choosing to save a map from the menu, the user is asked to provide certain information for the map he is going to create or update. This information consists of a title, a creator name, the license and theme of the map and also a brief description of what the map represents. Then we can choose the registry we want this map to be saved in and if the map will have a new identifier or we want to rewrite the existing map (in case the map was loaded first and after some changes should be saved again). In the end of the form we see all the available layers that are going to be saved. If we do not want some layers to be saved as part of the map we create, we can simply uncheck its checkbox. When saving the map, Sextant also calculates the extent of the map and adjusts the creation and last modification dates accordingly. Loading a map is simple if we know its identification string. We only need to provide that and also the registry URI that it is saved in if we use the load interface, or by accessing a URI with any browser as explained above.

After saving or loading a map in Sextant, the Layers panel and the Map Information panel are updated as depicted in Figure 4.3. These panels contain information about the layers in the map and the metadata of the map respectively. While the Layers panel gives us the content of the map providing a list of the layers it contains and all their metadata, the Map Information panel provides all the map metadata such as the title, the extent and others that together with the layers information act as the map legend.

4.3 Map Sharing

One of the key features we wanted to support is the ability to easily share the maps created by the application. Doing this we enable users to load existing maps, edit them if needed and save them again under a new identifier or by overwriting the existing map. This functionality supports both the collaborative editing that may be needed in the creation of a map, but also assist in the use of the map by third parties. When saving a map, Sextant creates a unique



(a) (b)

(c)

Figure 4.2: Save map modal (a), Load map modal (b) and a loaded Test map

identifier for this map and publishes it as a web resource under a specific URI of the form http://<domain>/Sextant/?mapid=<mapid>, where <domain> corresponds to the domain name the Sextant web application may be accessed and <mapid> is the unique identifier of the map we want to load. This URI can then be shared among users for collaborative viewing and editing of the map.

4.4 Map Search

Each map is accompanied by a set of metadata as shown in the map ontology, that helps to get a brief description of what the map contains. These metadata apart from being used to set the map legend can also be used to fuel a search mechanism over these map characteristics. Using all

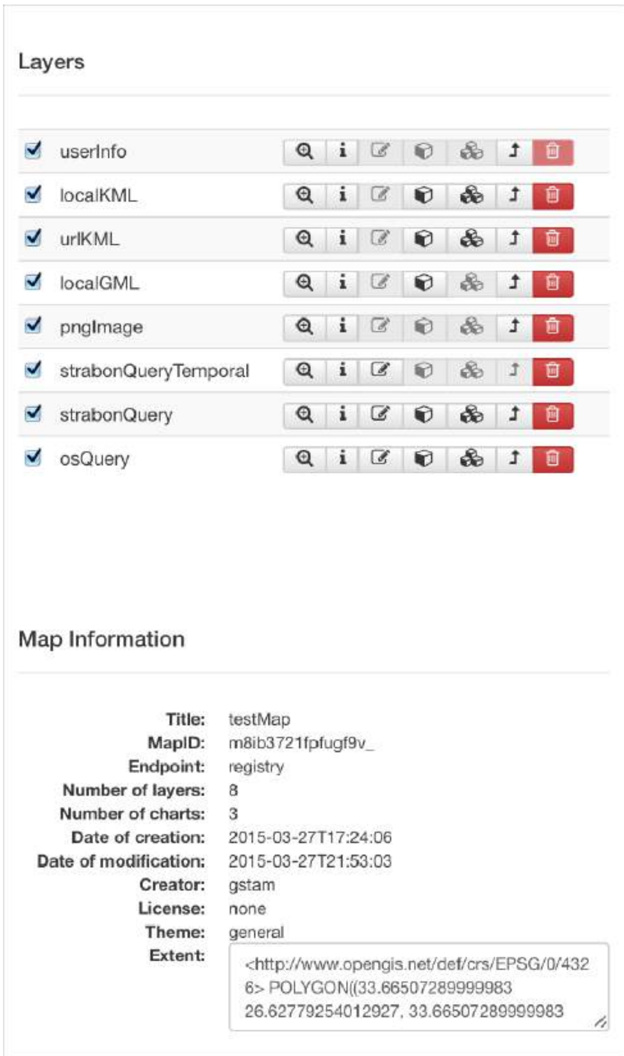
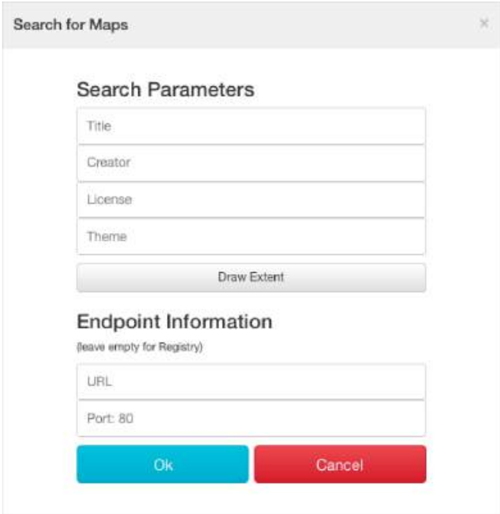


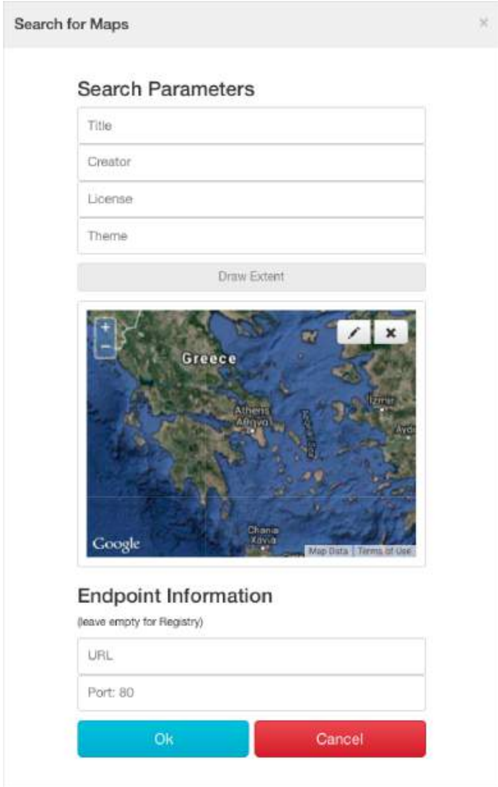
Figure 4.3: Legend Information for map

this metadata information a user can for example search in a map registry for maps that intersect with a given rectangle on the map, are created after 2012 and have information about fires. In this manner the application can become a powerful tool in the hands of domain experts from different fields that can have access on all these visualized data that are combined from different sources in a single map and can be easily retrieved by a single application.

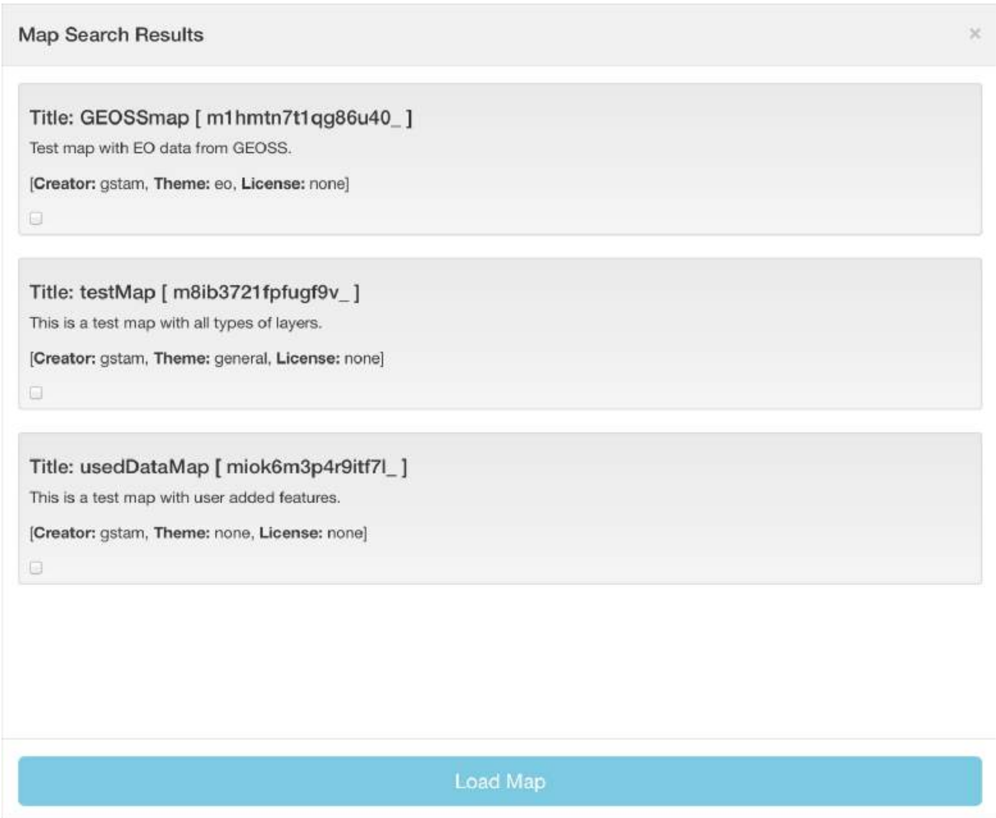
To search for maps, we provide a modal in the user interface as shown in Figure 4.4 where they can fill in a title, a creator name, the license and theme and also draw a rectangular on the map that represents an area of interest. In the end of the form they can also specify the registry that the search would take place. The results will be a list of maps with their description, that match the given key words in each field of the form and have an extent that intersects the given rectangular. The user can then choose one of the maps and load it on the world map.



(a)



(b)



(c)

Figure 4.4: Search map modal (a), extended Search map modal (b) and Search results

Chapter 5
Combine Data from Different Sources

A lot of effort was put the past years in Geographic Information Systems that resulted in creating a rich technological arsenal for the traditional GIS area. Many different GIS file formats were created by government mapping agencies or software developers and have become standards in encoding geographical information. Two are the main categories of these file formats according to the way the information is represented in the files. These are the raster and the vector file formats.

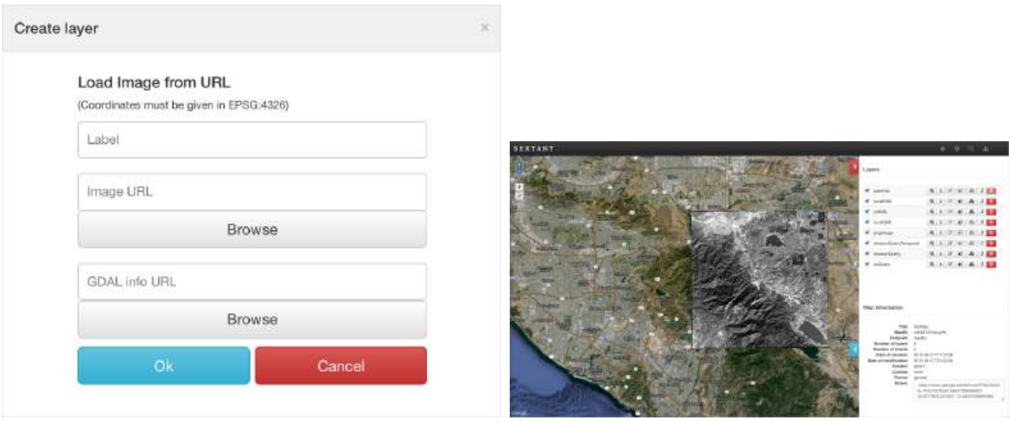
5.1 Raster Data

Raster Data rely on a spatial data model that defines space as an array of equally sized cells arranged in rows and columns, and composed of single or multiple bands. Each cell contains an attribute value and location coordinates. Unlike a vector structure, which stores coordinates explicitly, raster coordinates are contained in the ordering of the matrix. Groups of cells that share the same value represent the same type of geographic feature.

A raster data type is, in essence, any type of digital image represented by reducible and enlargeable grids. Anyone who is familiar with digital photography will recognize the Raster graphics pixel as the smallest individual grid unit building block of an image, usually not readily identified as an artifact shape until an image is produced on a very large scale. A combination of the pixels making up an image color formation scheme will compose details of an image, as is distinct from the commonly used points, lines, and polygon area location symbols of scalable vector graphics as the basis of the vector model of area attribute rendering. While a digital image is concerned with its output blending together its grid based details as an identifiable representation of reality, in a photograph or art image transferred into a computer, the raster data type will reflect a digitized abstraction of reality dealt with by grid populating tones or objects, quantities, cojoined or open boundaries, and map relief schemas. Aerial photos are one commonly used form of raster data, with one primary purpose in mind: to display a detailed image on a map area, or for the purposes of rendering its identifiable objects by digitization. Additional raster data sets used by a GIS will contain information regarding elevation, a digital elevation model, or reflectance of a particular wavelength of light, Landsat, or other electromagnetic spectrum indicators.

Raster data type consists of rows and columns of cells, with each cell storing a single value. Raster data can be images (raster images) with each pixel (or cell) containing a color value. Additional values recorded for each cell may be a discrete value, such as land use, a continuous value, such as temperature, or a null value if no data is available. While a raster cell stores a single value, it can be extended by using raster bands to represent RGB (red, green, blue) colors, colormaps (a mapping between a thematic code and RGB value), or an extended attribute table with one row for each unique cell value. The resolution of the raster data set is its cell width in ground units. Raster data is stored in various formats; from a standard file-based structure of TIFF, JPEG, etc. to binary large object(BLOB) data stored directly in a relational database management system (RDBMS) similar to other vector-based feature classes. Database storage, when properly indexed, typically allows for quicker retrieval of the raster data but can require storage of millions of significantly sized records.

In Sextant we support GeoTIFF files and any king of image file as long as a text file with the georeference metadata is provided. To get the spatial data we use GDAL library[1]. As shown in Figure 5.1, to load a GeoTiff layer, the user must provide the URI of the image or search for it in his local file system and a text file with the results of the gdalinfo command for this image. If GDAL library is installed on the server side, the user can provide its location in the server properties file when installing the application and then Sextant handles the calls to the GDAL library to extract the image size and extent, so the user only needs to provide the image file.



(a) (b)

Figure 5.1: Load GeoTIFF modal (a) and an image layer (b)

5.2 Vector Data

Vector Data rely on a coordinate-based data model that represents geographic features as points, lines, and polygons. Each point feature is represented as a single coordinate pair, while line and polygon features are represented as ordered lists of vertices. Attributes are associated with each vector feature, as opposed to a raster data model, which associates attributes with grid cells. (esri definition)

[1] `http://www.gdal.org/`

In a GIS, geographical features are often expressed as vectors, by considering those features as geometrical shapes. Different geographical features are expressed by different types of geometry:

- **Points.** Zero-dimensional points are used for geographical features that can best be expressed by a single point reference—in other words, by simple location. Examples include wells, peaks, features of interest, and trailheads. Points convey the least amount of information of these file types. Points can also be used to represent areas when displayed at a small scale. For example, cities on a map of the world might be represented by points rather than polygons. No measurements are possible with point features.
- **Lines or polylines.** One-dimensional lines or polylines are used for linear features such as rivers, roads, railroads, trails, and topographic lines. Again, as with point features, linear features displayed at a small scale will be represented as linear features rather than as a polygon. Line features can measure distance.
- **Polygons.** Two-dimensional polygons are used for geographical features that cover a particular area of the earth's surface. Such features may include lakes, park boundaries, buildings, city boundaries, or land uses. Polygons convey the most amount of information of the file types. Polygon features can measure perimeter and area.

Each of these geometries are linked to a row in a database that describes their attributes. For example, a database that describes lakes may contain a lake's depth, water quality, pollution level. This information can be used to make a map to describe a particular attribute of the dataset. For example, lakes could be colored depending on level of pollution. Different geometries can also be compared. For example, the GIS could be used to identify all wells (point geometry) that are within one kilometer of a lake (polygon geometry) that has a high level of pollution. Vector features can be made to respect spatial integrity through the application of topology rules such as 'polygons must not overlap'. Vector data can also be used to represent continuously varying phenomena. Contour lines and triangulated irregular networks (TIN) are used to represent elevation or other continuously changing values. TINs record values at point locations, which are connected by lines to form an irregular mesh of triangles. The face of the triangles represent the terrain surface.

Sextant supports KML and GML vector file formats and in Figure 5.2 we present how we can use the interface to create such layers. The user must provide the URI of the KML or GML file or search for it in his local file system and also give a name to the layer.

The user added information is also saved in a KML file. Through Sextant, users can draw points and polygons on the map and accompany them with some metadata such as title, date of creation and a description. In Figure 5.3 we present the modals that allow the creation of these features and an example of a point and a polygon added on the map.

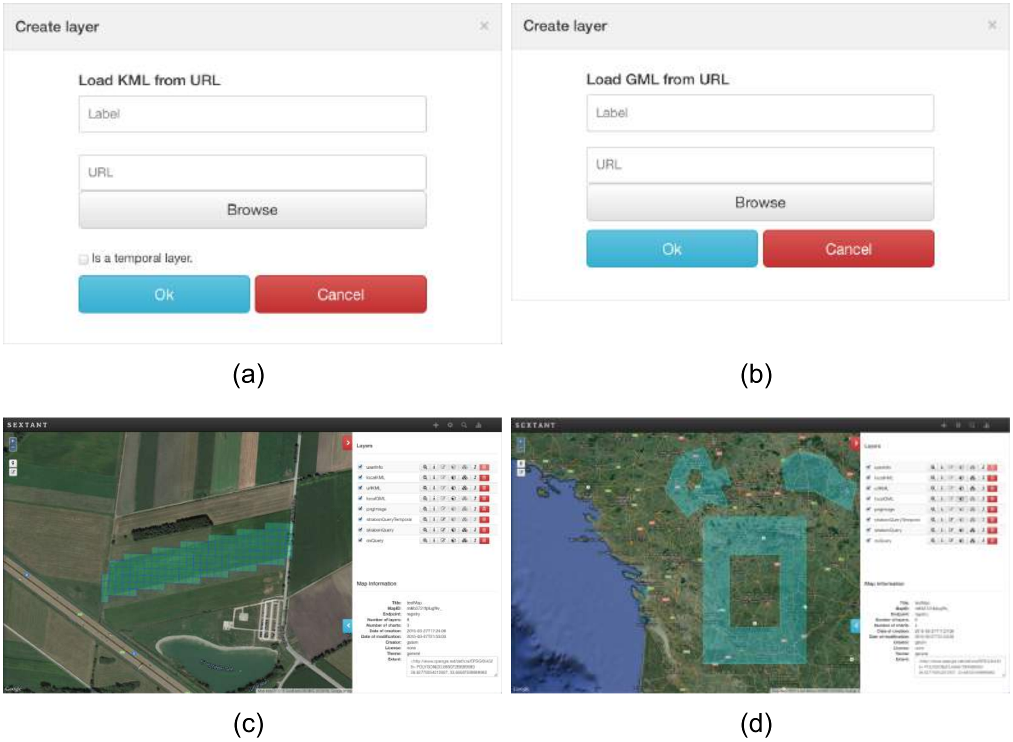


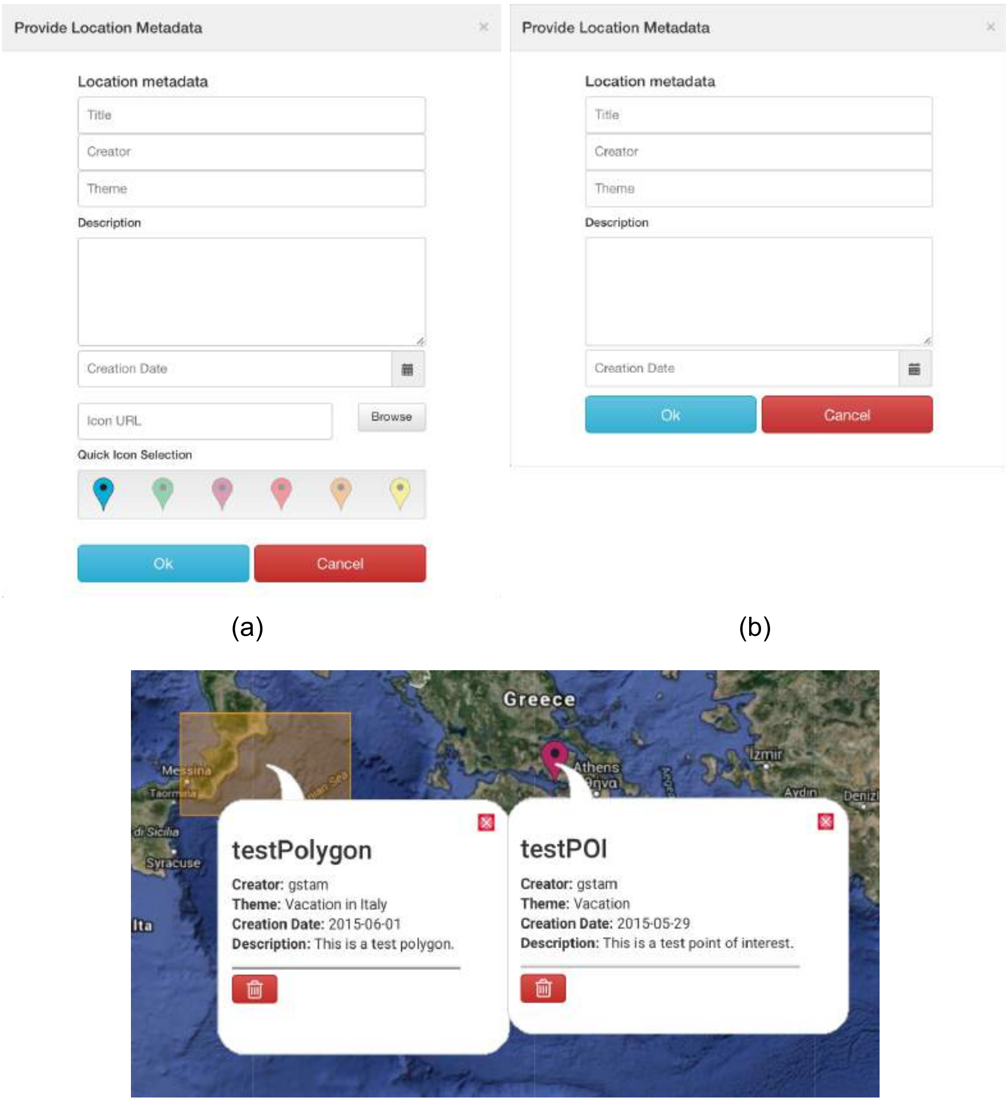
Figure 5.2: Load KML modal (a), load GML modal (b), a KML layer (c) and a GML layer (d)

These points and polygons are saved in the predefined userInfo layer and are saved and loaded along with the other layers of the map. To add these features we can use the respective buttons under the zoom buttons in the upper left corner of the UI. To delete one of these features we can use the red bin button in its pop-up window.

5.3 Layers in Sextant

After creating a layer the user can interact with it and graphically edit it. According to the file format of each layer there are a set of functions that Sextant provides for interaction. There are eight basic functions, as depicted in Figure 5.4.

- Checkbox that represents if the layer is shown or hidden on the map.
- Zoom function that zooms the map to the extent of the layer.
- An information button that shows to the user all the information regarding this layer.
- Query edit and update function that shows the query that resulted in the creation of this layer and the user can edit the query and update the current layer. This function is available only for layers that are created by querying a SPARQL endpoint.



(a) (b)

(c)

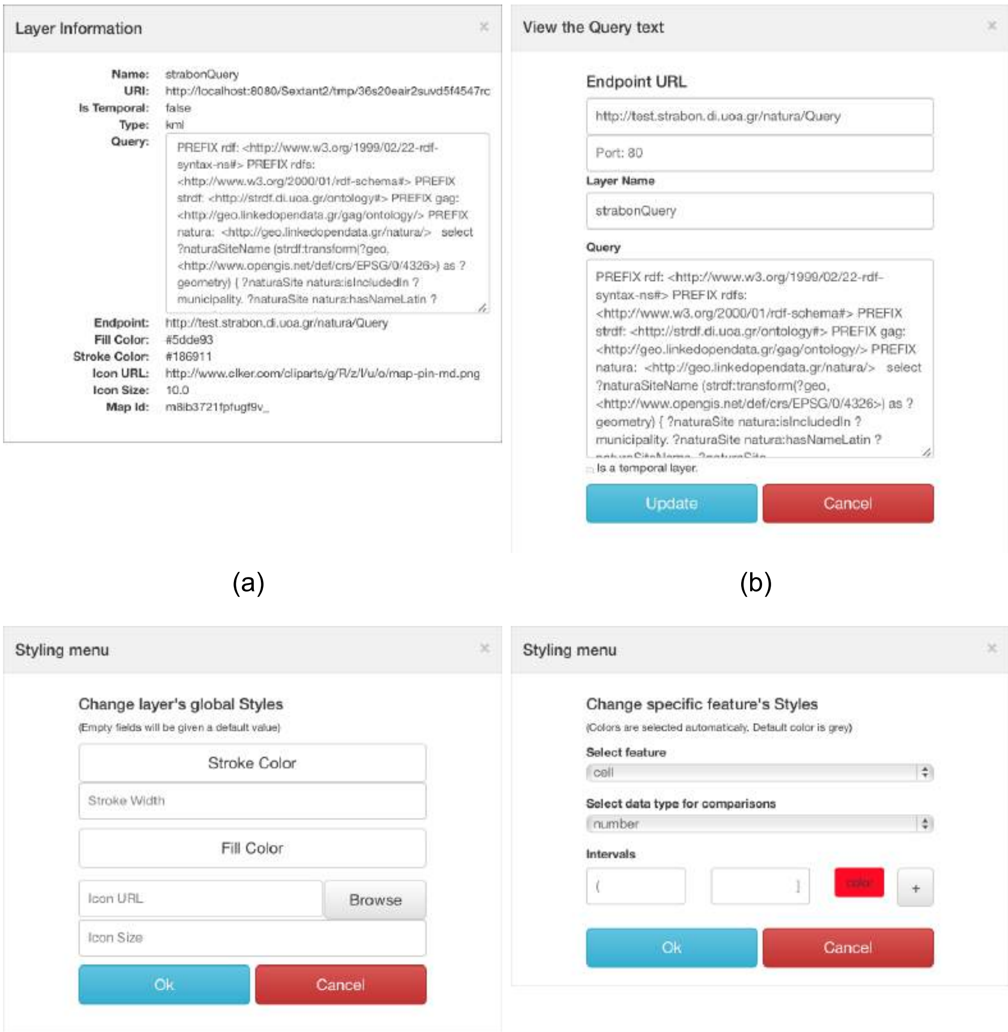
Figure 5.3: Add POI modal (a), add AOI modal (b) and an example map (c)

- Global styling function that provides an interface to chromatically edit vector layers.
- Feature styling function that provides an interface for changing the color of each placemark in a KML file according to the value of a selected feature.
- Move the layer on top of every other layer on the map.
- Delete button to erase the layer from the map.

In Figure 5.5 we demonstrate some of the basic functions we described. By selecting the information button a modal appears with all the metadata for the specific layer. The next modal is shown when we choose to update a layer that is the result of a SPARQL query. The endpoint URI and

Figure 5.4: The eight basic functions for layer manipulation

the text of the query are auto-filled and the user can alter both of them to update the layer with the new results. In the last two modals we show the UI forms that appear when we choose to change the global colors of a KML or GML layer and a feature's colors according to specified intervals in KML layers.

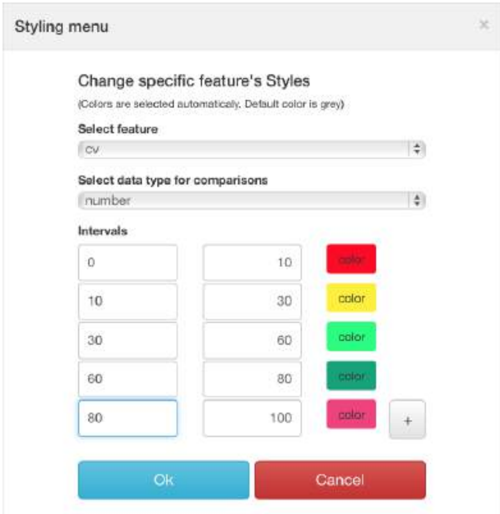


(a) (b)

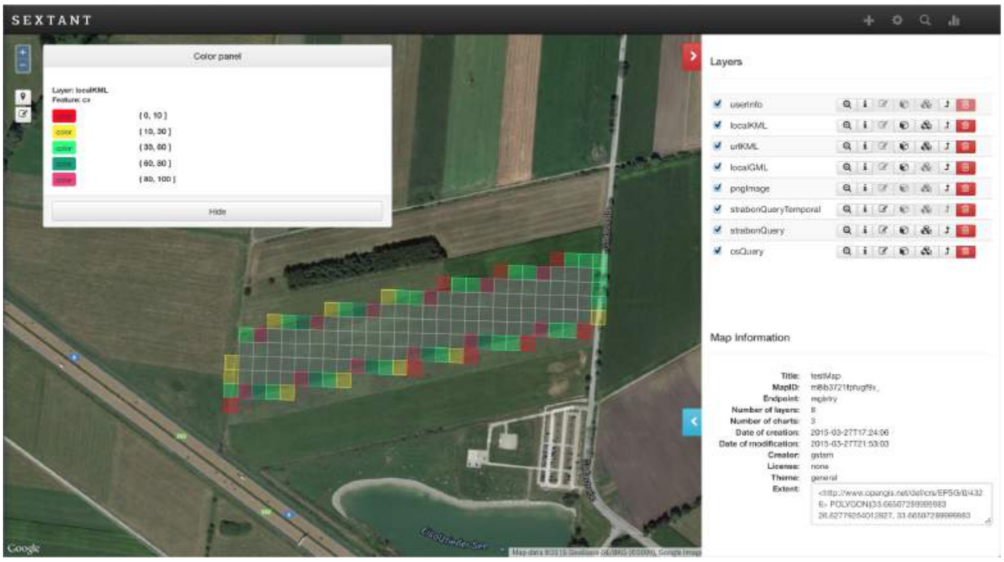
(c) (d)

Figure 5.5: Layer information modal (a), update query modal (b), global style modal (c) and feature style modal (d)

In Figure 5.6 we showcase how to use the feature's coloring to create color maps and have a more detailed and easy to understand representation of a specific KML layer. In this example we have a field that is represented by its raster cells. Each raster cell has certain values for the soil. By using Sextant and the feature coloring we select the numeric CV value of each raster cell and create the intervals that are interesting to us. Then we can see how the layer is updated by coloring each raster cell of the field with the respective color and the Color Panel appears that has a legend that presents the name of the layer, the feature that this colorization filters are applied to and the intervals we created with their respective colors.



(a)



(b)

Figure 5.6: Feature coloring

Chapter 6
Temporal Dimension

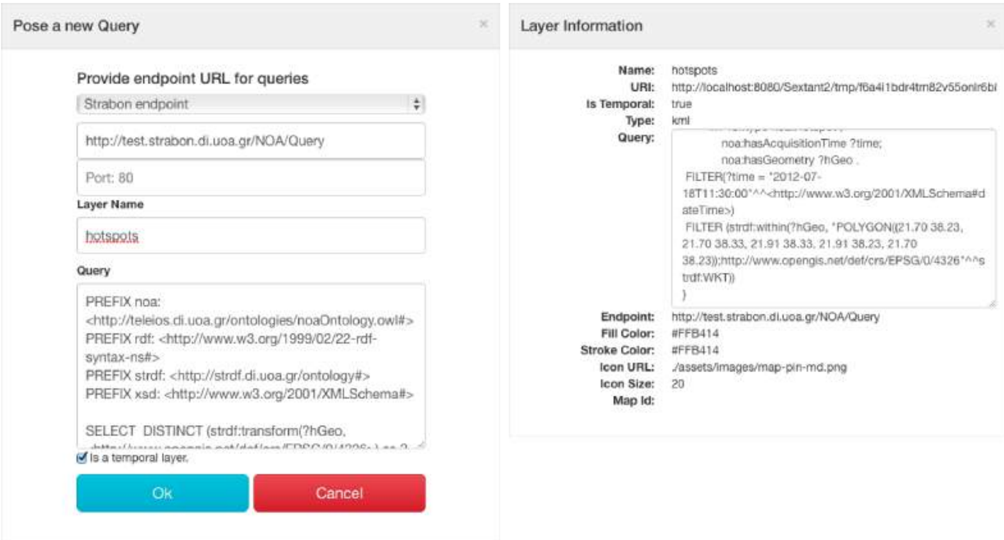
Time is an extra dimension that is often used in data sets to specify the valid time of an event. We wanted to give the users of Sextant the ability to bind the layers they create with the time dimension, so we added to the interface a timeline that can help visualize time events. To implement this feature we use the SIMILE Timeline widget that provides us with a timeline where we can add time events. To integrate this timeline with the layers we create, we used the Timemap JavaScript library that gives us the ability to show or hide temporal layers on the map according to the current position of the Timeline band. Since we use the system Strabon to store and query our data and Strabon is a spatiotemporal RDF store, temporal layers can be the result of a query written in stSPARQL which is an extension of SPARQL 1.1 that supports temporal functions.

Temporal layers can help us visualize the evolution of an observation through time, which is an important tool at the hands of specialists in a variety of domains such as climate, ecosystems, disasters, land use, agriculture and more. In the current version of Sextant temporal layers can be either the result of a query or KML files that contain temporal information, but we are working on providing the tools to add the temporal dimension to any type of layer by defining its valid time, even if this information is not available in the original data.

6.1 Time in Sextant

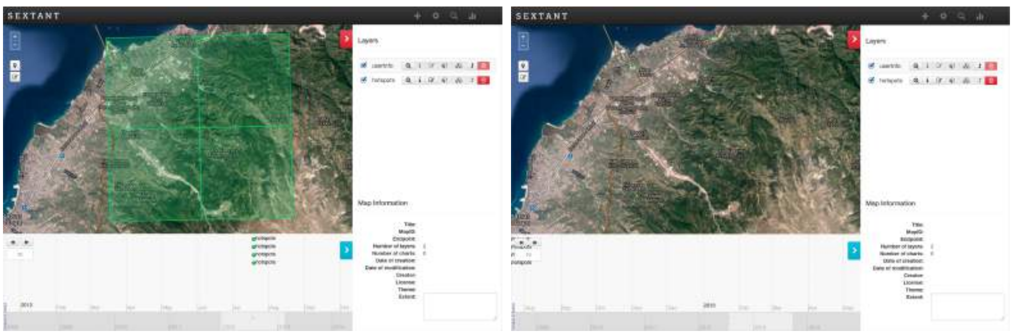
Implementation of he valid time component of stRDF and stSPARQL in system Strabon allows us to query both the spatial and the temporal dimension. Enriching our results with temporal information allows us to create layers with valid time. Using the SIMILE Timeline widget we can make these layers appear and disappear from the map according to their valid time. In this section we demonstrate how we can achieve this functionality using Sextant. To create a temporal layer the user can provide a KML enriched with temoral information or pose a query to a Strabon SPARQL endpoint and in both cases check the *"is temporal layer"* checkbox provided in the modal. An example of a query with temporal dimension that results in the creation of a temporal layer is depicted in Figure 6.1.

As we can see in this query we have applied a filter on the dateTime `2012-07-18T11:30:00` on the hotspots we want to get in the result set. Sextant uses this information with the assistance of Timemap JavaScript library and creates an event on the SIMILE Timeline for the created layer. As demonstrated in Figure 6.2, the layer is visible while the Timeline band intersects the event, but disappears from the map if the event is not within the main band.



(a) (b)

Figure 6.1: Query modal for temporal layer (a) and temporal layer information modal (b)



(a) (b)

Figure 6.2: Temporal layer shown (a) and temporal layer hidden (b), according to Timeline band position

Chapter 7
Discovering Earth Observation information in Sextant

Apart from visualizing standard GIS file formatted data, in many cases our data may lie in many different data bases. In order to use these data and produce layers over a map, one should first have an intuitive tool to discover all the information needed to be visualized. When using semantic web technologies to model and store our data, we can easily create catalogues with metadata over the underlying datasets and provide search mechanisms to the end user to allow the exploration of the data and visualization of the results. The key to using this infrastructure is the use of standards in both the modeling of the datasets and the metadata, so that the search mechanism can span across multiple catalogues and allow the linking of datasets. A lot of effort is put the last years in this direction with the support of OGC, in an attempt to encourage development and implementation of open standards for geospatial content and services, GIS data processing and data sharing, and W3C that offers a large palette of techniques to describe and define different forms of vocabularies in a standard format.

7.1 Catalogue Search for Earth Observation Data

Established in 2005, GEO[1] is a voluntary partnership of governments and organizations that envisions “a future wherein decisions and actions for the benefit of humankind are informed by coordinated, comprehensive and sustained Earth observations and information.” GEO Member governments include 96 nations and the European Commission, and 87 Participating Organizations comprised of international bodies with a mandate in Earth observations. Together, the GEO community is creating a Global Earth Observation System of Systems (GEOSS) that will link Earth observation resources world-wide across multiple Societal Benefit Areas - agriculture, biodiversity, climate, disasters, ecosystems, energy, health, water and weather - and make those resources available for better informed decision-making.

In an attempt to link Earth Observation (EO) resources, the GEO Discovery and Access Broker[2] (DAB) was created by ESSI lab[3]. DAB is a middleware component which is in charge of interconnecting the heterogeneous and distributed capacities contributing to GEOSS and became part of the GEOSS Common Infrastructure (GCI) since November 2011. The DAB provides three main functionalities:

[1] `http://www.earthobservations.org/index.php`
[2] `http://api.eurogeoss-broker.eu/docs/index.html`
[3] `http://www.uos-firenze.essi-lab.eu`

- Discovery of resources from brokered sources
- Semantics-enriched discovery
- Access of resources

DAB is used in GEOSS Portal[4] to provide an entry point to EO data from all over the world.

To support EO data discovery, exploration and visualization, we added the DAB component in Sextant. Users can use the modal depicted in Figure 7.1 ,where you can provide key words, a general theme, the valid time of the data and draw an extent for the area of interest, to search for EO data that meet the given criteria.

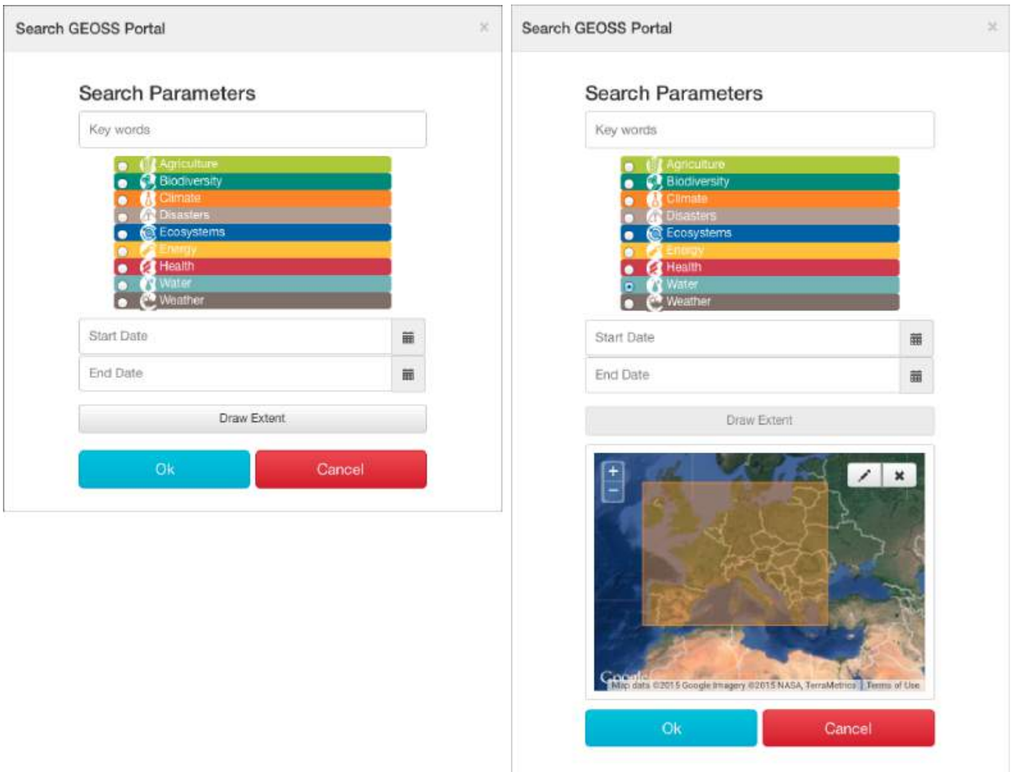


Figure 7.1: Search GEOSS Portal using Sextant

The results are presented using the reports DAB provides, that follow the ISO 19115[5] specification, which suggests formats for visualizing the results' metadata, such as KML, KML, WMS, GeoTIFF and more. In Figure 7.2 we demonstrate the results of a search query for water themed EO data in the area of Europe.

[4]http://www.geoportal.org/web/guest/geo_home_stp
[5]http://api.eurogeoss-broker.eu/docs/classes/Report.html

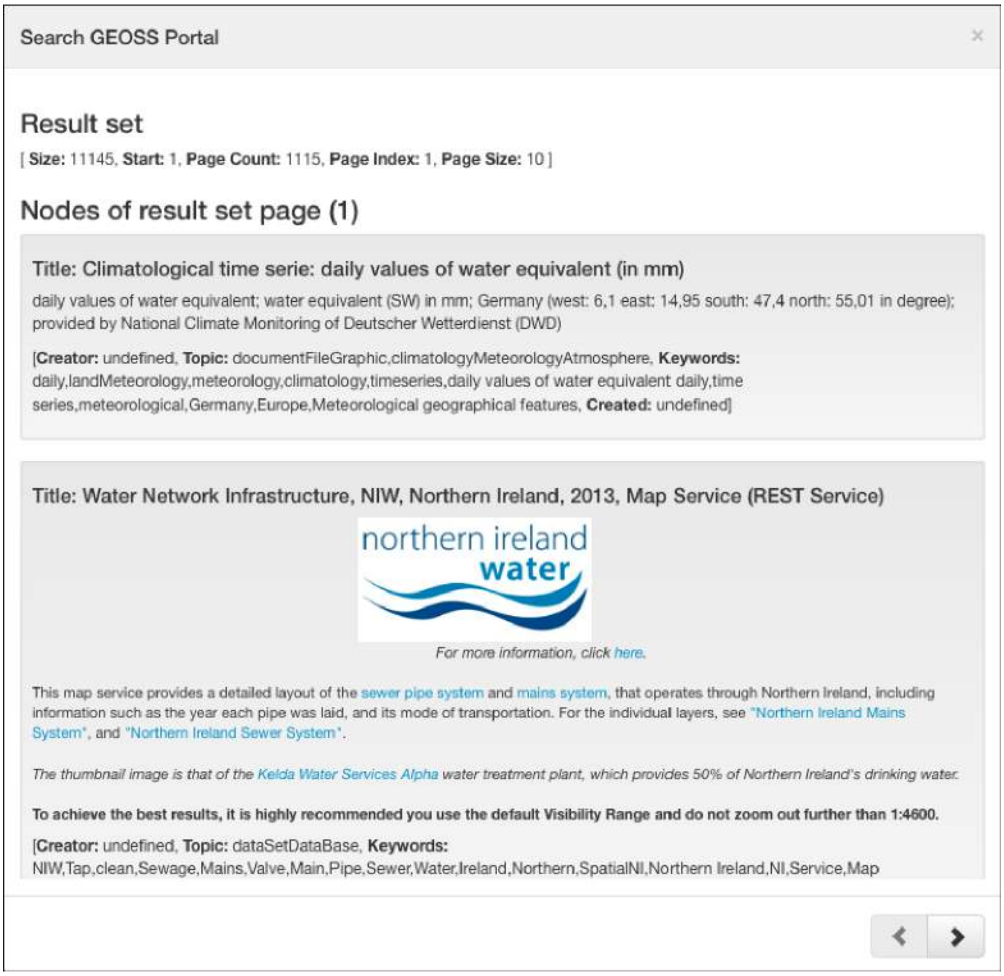


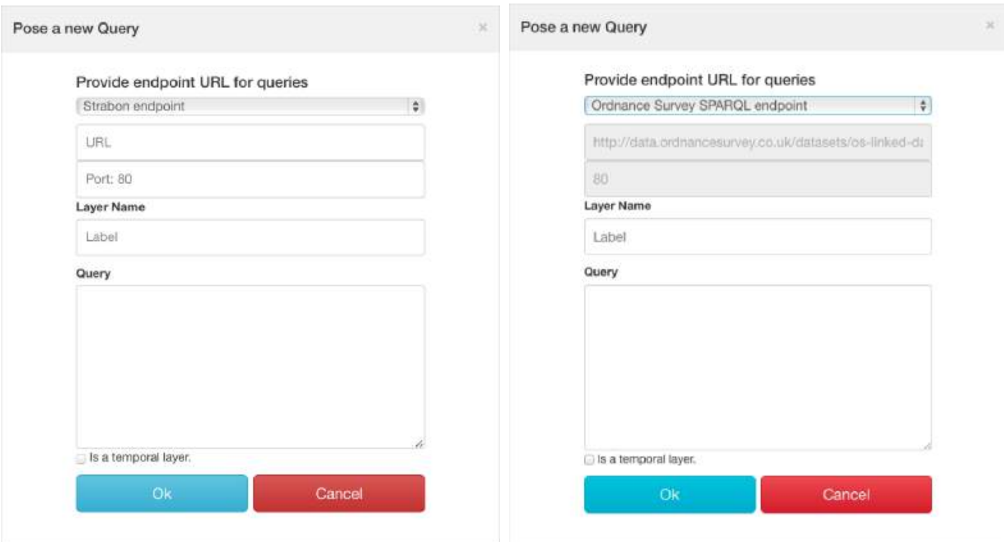
Figure 7.2: Search results from GEOSS Portal

7.2 Querying Endpoints

While providing search mechanisms in catalogues assists users in locating and visualizing information, most of the times the needed information is stored in a databases known to the users or even created by them. To gain access to this information and allow the visualization of the data, we offer a querying mechanism over RDF stores using the query languages stSPARQL and GeoSPARQL.

Sextant provides an interface for querying SPARQL endpoints using stSPARQL or GeoSPARQL and presenting the results as a layer on the world map. In the current version of Sextant, users can use the platform to pose queries to various SPARQL endpoints, such as Strabon endpoints and the Ordnance Survey SPARQL endpoint. Responsible for supporting these features are the endpoint connector and the KML translator server components. The endpoint connector undertakes the communication with SPARQL endpoints that are publically accessible through a URI, and are compliant with the specification documents of SPARQL 1.1 protocol and the SPARQL query results XML format. In order to visualize the results of a query on the map we use the KML translator component that translates the results of GeoSPARQL and stSPARQL queries from the SPARQL query results XML format to the KML file format. With respect to time representation, the KML standard defines the abstract element `kml:TimePrimitive` which is realized by two elements `kml:TimeStap` and `kml:TimeSpan` for representation of temporal instants and intervals respectively. The KML translator, translates any temporal instant expressed using the datatype `xml:dateTime` into the element `kml:TimeStamp` and any temporal interval expressed using datatype of stRDF `strdf:period` into the element `kml:TimeSpan`.

To create a layer by posing a query the modal depicted in Figure 7.3 is provided, where the user can select the appropriate type of the endpoint and provide a URI for it, a name for the layer and the query text. Also as we saw in the previous chapter the *"is temporal layer"* checkbox can be used for temporal layers. As a result, the query is evaluated in the specified endpoint and the returned results are visualized as a KML vector layer on the map.



(a) (b)

Figure 7.3: Query modal for Stabon Endpoints (a) and Ordnance Survey SPARQL Endpoint (b)

This querying feature was originally designed to be used by domain experts or users familiar with these query languages, but we wanted to provide a way for non-experts to use this feature too. To accomplish that we added the predefined queries feature. Predefined queries are a set of queries written by an expert that are made available through an endpoint to non-expert users of the platform. To implement predefined queries we introduce the ontology of Figure 7.4.

Each query is a resource that has a label that describes in plain text its functionality, a text that is the query expressed in stSPARQL or GeoSPARQL and a boolean attribute to verify if the evaluation of it, results in temporal layer. Using this new feature, non-expert users can now provide the URI of an endpoint that holds information they want to explore through Sextant user interface

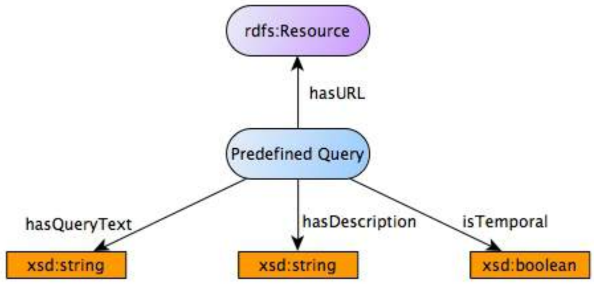


Figure 7.4: The predefined query ontology

and get a list of the predefined queries that are available in this endpoint. Then they can choose one of the results presented in plain text that they can easily understand and view the respective layer on the map. In this way the process of creating a layer from the results of a query is made transparent to user, as Sextant takes control of evaluating the selected query to the endpoint and creating the layer from the result information. An example to showcase this functionality is depicted in Figure 7.5.

Figure 7.5: Predefined Query modal (a), the predefined queries that are available (b) and the layer creation from the selected query (c)

Chapter 8
Visualizing statistical information in Sextant

Sextant also provides a feature for producing statistical charts from the data stored in SPARQL endpoints. Although charts can be produced by a SPARQL query, writing such a queries would require extensive use of SPARQL. To allow non-experts to use this functionality we provide an intuitive way for producing statistical charts without the need of writing such a query, making the process transparent to the end user.

8.1 The RDF Data Cube Vocabulary

There are many situations where it would be useful to be able to publish multi-dimensional data, such as statistics, on the web in such a way that it can be linked to related data sets and concepts. The Data Cube vocabulary[1] provides a means to do this using the W3C RDF standard. The model underpinning the Data Cube vocabulary is compatible with the cube model that underlies SDMX (Statistical Data and Metadata eXchange), an ISO standard for exchanging and sharing statistical data and metadata among organizations. The Data Cube vocabulary is a core foundation which supports extension vocabularies to enable publication of other aspects of statistical data flows or other multi-dimensional data sets.

Statistical data is a foundation for policy prediction, planning and adjustments and underpins many of the mash-ups and visualizations we see on the web. There is strong interest in being able to publish statistical data in a web-friendly format to enable it to be linked and combined with related information. At the heart of a statistical dataset is a set of observed values organized along a group of dimensions, together with associated metadata. The Data Cube vocabulary enables such information to be represented using the W3C RDF standard and published following the principles of linked data. The vocabulary is based upon the approach used by the SDMX ISO standard for statistical data exchange. This cube model is very general and so the Data Cube vocabulary can be used for other data sets such as survey data, spreadsheets and OLAP data cubes.

The Data Cube vocabulary is focused purely on the publication of multi-dimensional data on the web. They envisage a series of modular vocabularies being developed which extend this core foundation. Other extensions are possible to support metadata for surveys or publication of statistical reference metadata. The Data Cube in turn builds upon the following existing RDF vocabularies:

[1] `http://www.w3.org/TR/vocab-data-cube/`

- SKOS for concept schemes
- SCOVO for core statistical structures
- Dublin Core Terms for metadata
- VoiD for data access
- FOAF for agents
- ORG for organizations

Linked data is an approach to publishing data on the web, enabling datasets to be linked together through references to common concepts. The approach LOD[2] recommends use of HTTP URIs to name the entities and concepts so that consumers of the data can look-up those URIs to get more information, including links to other related URIs. RDF provides a standard for the representation of the information that describes those entities and concepts, and is returned by dereferencing the URIs. There are a number of benefits to being able to publish multi-dimensional data, such as statistics, using RDF and the linked data approach:

- The individual observations, and groups of observations, become (web) addressable. This allows publishers and third parties to annotate and link to this data; for example a report can reference the specific figures it is based on allowing for fine grained provenance trace-back.
- Data can be flexibly combined across datasets sets (for example find all Religious schools in census areas with high values for National Indicators pertaining to religious tolerance). The statistical data becomes an integral part of the broader web of linked data.
- For publishers who currently only offer static files then publishing as linked-data offers a flexible, non-proprietary, machine readable means of publication that supports an out-of-the-box web API for programmatic access.
- It enables reuse of standardized tools and components.

8.2 Statistical Data and Metadata Exchange (SDMX)

The Statistical Data and Metadata Exchange (SDMX) Initiative was organized in 2001 by seven international organizations (BIS, ECB, Eurostat, IMF, OECD, World Bank and the UN) to realize greater efficiencies in statistical practice. These organizations all collect significant amounts of data, mostly from the national level, to support policy. They also disseminate data at the supra-national and international levels. There have been a number of important results from this work:

[2] http://linkeddata.org

two versions of a set of technical specifications - ISO:TS 17369 (SDMX) - and the release of several recommendations for structuring and harmonizing cross-domain statistics, the SDMX Content-Oriented Guidelines. All of the products are available at `http://sdmx.org`. The standards are now being widely adopted around the world for the collection, exchange, processing, and dissemination of aggregate statistics by official statistical organizations. The UN Statistical Commission recommended SDMX as the preferred standard for statistics in 2007.

The SDMX specification defines a core information model which is reflected in concrete form in two syntaxes - SDMX-ML (an XML syntax) and SDMX-EDI. The RDF Data Cube vocabulary builds upon the core of the the SDMX 2.0 Information Model[3]. A key component of the SDMX standards package are the Content-Oriented Guidelines (COGs), a set of cross-domain concepts, code lists, and categories that support interoperability and comparability between datasets by providing a shared terminology between SDMX implementers. RDF versions of these terms are available separately for use along with the Data Cube vocabulary. These external resources do not form a normative part of the Data Cube Vocabulary specification.

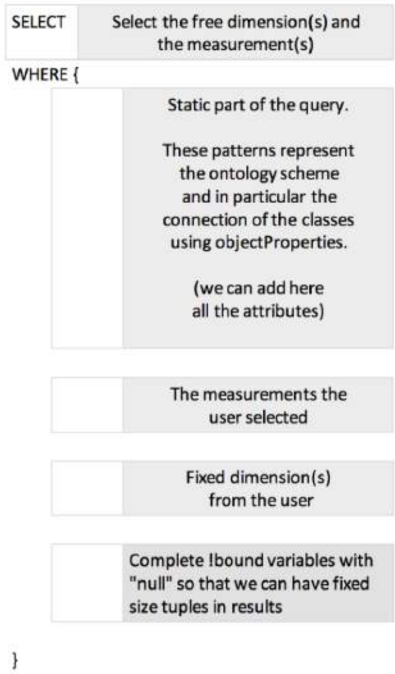
8.3 Statistical enhancement of datasets

The RDF Data Cube Vocabulary presents an ontology for publishing multi-dimensional data, such as statistics into RDF. In most cases though data sets are already published using another ontology and publishing the data again in the RDF Data Cube Vocabulary to provide statistics would be inefficient. In this thesis we present a process for statistical enhancement of an existing dataset published as RDF with an arbitrary ontology. To implement that we use the classes `qb:DimensionProperty`, `qb:MeasureProperty` and `qb:AttributeProperty` along with the property `qb:order` from the RDF Data Cube Vocabulary. What we want to achieve is to create a single query depicted in Figure 8.1(a) that will adjust itself according to the statistic information we want to extract from a data set.

Using the above vocabulary, to statistically enhance a data set we first need to define which properties represent the dimension and which represent our measures. A basic rule can be that object properties define dimensions and numeric datatype properties are the measures. We use the property qb:order to represent the hierarchical structure of the ontology and thus the dimensions' hierarchy. The next step is to create the static part of the general query. The static part is a graph pattern that would match all the data in our set. Since we want to extract statistical information, we need to run through the data as a whole and select the data measurements that are needed.

Using the introduced technique we can now define some parts of the query as optional and add filters to narrow down the results to our needs. A query produced with this technique is depicted in

[3] `http://sdmx.org/docs/2_0/SDMX_2_0%20SECTION_02_InformationModel.pdf`



(a)

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX tf: <http://data.linkedeodata.eu/talking-fields/ontology#>

SELECT DISTINCT ?RasterCell ?hasCV
WHERE {
  ?Farm rdf:type tf:Farm .
  ?Field rdf:type tf:Field .
  ?RasterCell rdf:type tf:RasterCell .

  ?Field tf:belongsToFarm ?Farm .
  ?RasterCell tf:belongsToField ?Field .

  OPTIONAL { ?RasterCell tf:hasCV ?hasCV. } .

  ?Field rdf:type ?typeField .
  FILTER (
    ?Field = </Field/id/1078> &&
    ?typeField = tf:Field ) .

  BIND (IF(!bound(?hasCV), "none", ?hasCV) AS ?hasCV) .
}
ORDER BY ?RasterCell
```

(b)

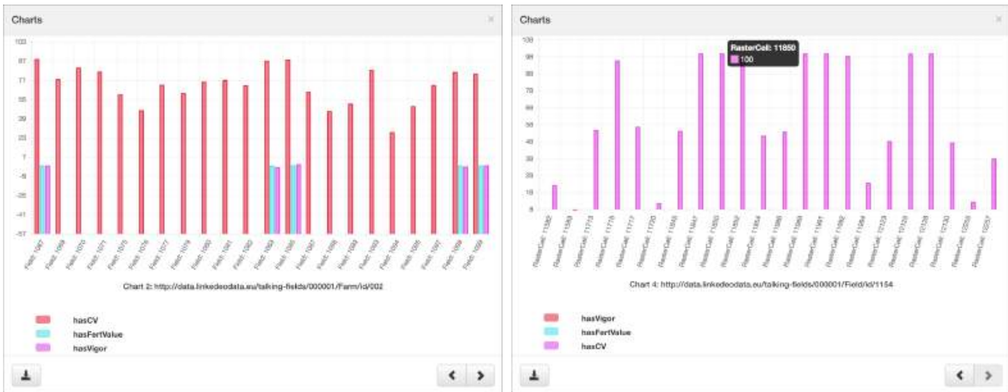
Figure 8.1: General Statistical Query (a) and a Statistical Query example (b)

Figure 8.1(b). In the selection clause we have all the free dimensions and the measurements we want in our results. The where clause consists of the static part along with the optional parts on the selected measures and the filtering on the fixed dimensions. Some bindings are also added to handle null values and ensure that we have fixed size tuples in the results. As with the predefined queries, an expert is needed to make the adjustments described in the ontology and provide the static part of the query as an RDF triplet. This procedure though is simple and needs to be done once for a data set to enhance it for statistical use. When these tasks are finished we can easily produce charts using Sextant interface.

8.4 Chart production using Sextant

In Figure 8.4 we present the modal from the user interface that will result in creating one or more charts according to the selected values. The user provides an endpoint URI with the statistically enhanced data. When connection is achieved, all the dimensions are presented and we can choose which dimensions to fix. As a result, all the instances of the selected fixed dimensions are presented to select one from each, along with the measures available. When the instances and the measures are selected Sextant uses the information given to create the final query and present the results as charts.

Each chart represents the variations of the selected fixed instances and measures over a free dimension, so according to the data set and the user selections more than one charts can derive from a single query. In Figure 8.2 we can see some examples of created charts. The charts are rendered using the Charts.js[4] JavaScript library that provides responsive and interactive charts. Using the button in the lower left corner of the modal that holds the charts associated to our map, we can download the chart as an image.



(a) (b)

Figure 8.2: Various Charts created with Sextant

As with map layers, charts are also saved when we choose to save a map. Charts are also represented in RDF using the ontology depicted in Figure 8.3. Each map can have a number of charts that have a specific type and are produced by a SPARQL query. Also each chart has a list of instances that are the instances of the fixed dimensions, a list of measures that are the measures we selected to visualize and a list of the free dimensions in our chart.

[4] http://www.chartjs.org

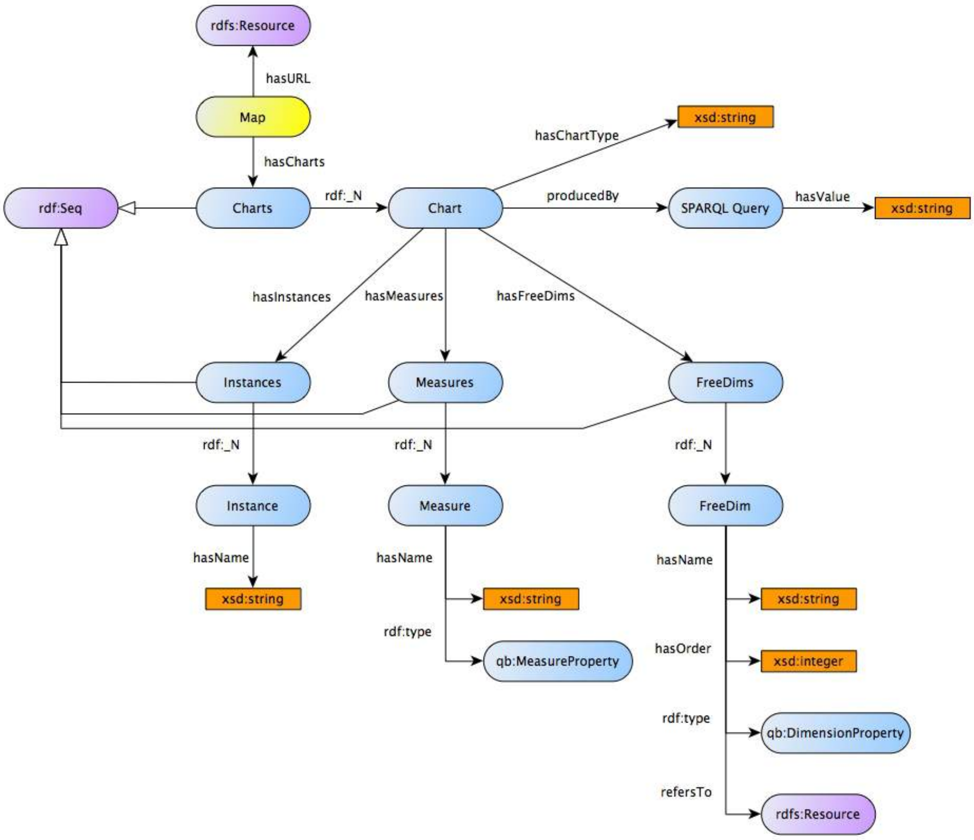


Figure 8.3: Charts Ontology

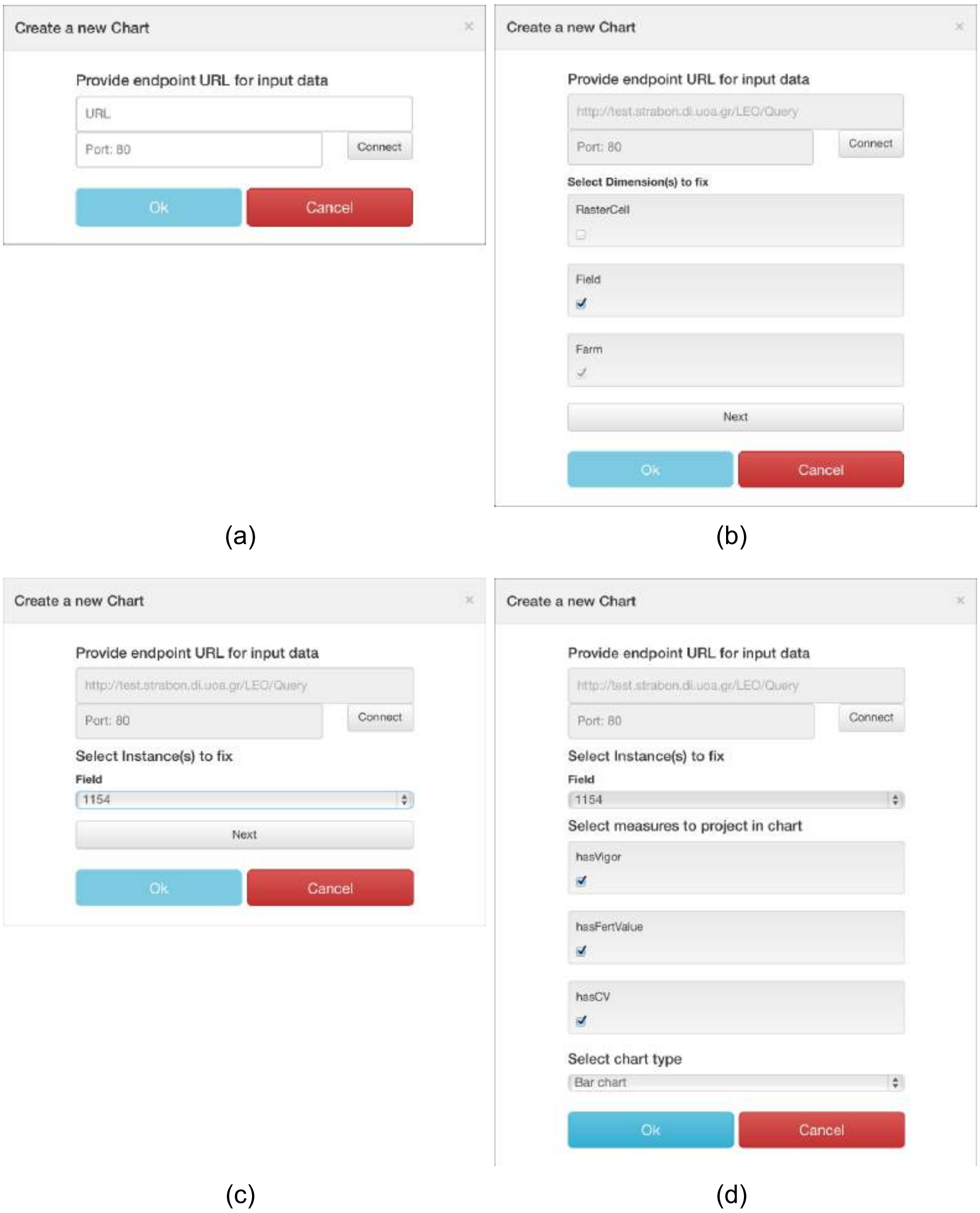


Figure 8.4: Create chart(s)

Chapter 9
Precision farming application in Sextant

In this chapter we will demonstrate how Sextant is used as one of the visualization and exploration tools in the FP7 project LEO[1] (611141). The main purpose of the project is to develop tools to aid precision farming[2], in an attempt to improve agriculture and cope with the increasing needs for various agricultural products in the next years.

9.1 FP7 project LEO

Lots of Earth Observation data has become available at no charge in Europe and the US recently and there is a strong push for more open EO data. For example, a recent paper on Landsat data use and charges by the US National Geospatial Advisory Committee – Landsat Advisory Group starts with the following overarching recommendation: “Landsat data must continue to be distributed at no cost”. Similarly, the five ESA Sentinel satellites that would soon go into orbit, starting with Sentinel-1 in 2013, have already adopted a fully open and free data access policy.

Open EO data that are currently made available by space agencies such as ESA and NASA are not following the linked data paradigm. Therefore, from the perspective of a user, the EO data and other kinds of geospatial data necessary to satisfy his or her information need can only be found in different data silos, where each silo may contain only part of the needed data. Opening up these silos by publishing their contents as RDF and interlinking them with semantic connections will allow the development of data analytics applications with great environmental and financial value.

In LEO, the core academic partners of TELEIOS[3] (UoA[4] and CWI[5]) join forces with two SMEs (SpaceApps[6], VISTA[7]) and one industrial partner (PCA[8]) with relevant experience to develop software tools that support the whole life cycle of reuse of linked open EO data and related linked geospatial data. Finally, to demonstrate the benefits of linked open EO data and its combination with linked geospatial to the European economy, a precision farming application is developed that is heavily based on such data.

[1] http://linkedeodata.eu
[2] http://earthobservatory.nasa.gov/Features/PrecisionFarming/
[3] http://www.earthobservatory.eu
[4] http://www.uoa.gr
[5] http://www.cwi.nl
[6] http://www.spaceapplications.com
[7] http://www.vista-geo.de/vista/eng/home/home.php
[8] http://www.pc-agrar.de

9.2 Precision Farming

Precision agriculture (PA) or satellite farming or site specific crop management (SSCM) is a farming management concept based on observing, measuring and responding to inter and intra-field variability in crops. Crop variability typically has both a spatial and temporal component which makes statistical/computational treatments quite involved. The holy grail of precision agriculture research will be the ability to define a Decision Support System (DSS) for whole farm management with the goal of optimizing returns on inputs while preserving resources. The reality today is that seemingly simple concepts such as the ability to define management zones, areas where different management practices will apply, for a single crop type on a single field over time are difficult to define [26, 36]. Whelan and McBratney (2003) articulate a number of approaches that are currently being used to define management zones (mostly by the research community), these include hand drawn polygons on yield maps, supervised and unsupervised classification procedures on satellite or aerial imagery, identification of yield stability patterns across seasons, etc. Among these many approaches is a phytogeomorphological approach which ties multi-year crop growth stability/characteristics to topological terrain attributes. The interest in the phytogeomorphological approach stems from the fact that the geomorphology component typically dictates the hydrology of the farm field. Multi-year datasets are now becoming available that show this stability and these effects, however, there is a lot of work remaining to create an actual DSS system that could universally help farmers.

It can be said that the practice of precision agriculture was enabled by the advent of GPS and GNSS. The farmer's and/or researcher's ability to locate their precise position in a field allows for the creation of maps of the spatial variability of as many variables as can be measured (e.g. crop yield, terrain features/topography, organic matter content, moisture levels, nitrogen levels, pH, EC, Mg, K, etc.). Further, these maps can be interpolated onto a common grid for comparison. Spatial and temporal variability of crop variables are at the heart of PA, while the spatial and temporal behaviors of that variability are key to defining amendment strategies, or *recipe maps*. Recipe maps would be the output of any generalized decision support system that could be defined for farm use. Precision agriculture has also been enabled by technologies like crop yield monitors mounted on GPS equipped combines, the development of variable rate technology (VRT) like seeders, sprayers, etc., the development of an array of real-time vehicle mountable sensors that measure everything from chlorophyll levels to plant water status, multi- and hyper-spectral aerial and satellite imagery, from which products like NDVI maps can be made, although the costs of these are high.

9.3 The role of Sextant

In this section we demonstrate the use of Sextant as part of the precision farming application. After the processing chain of EO data is completed, we have the final EO products published in RDF and linked to related geospatial data. Sextant can now be used to provide visualization and exploration over these datasets by communicating with the SPARQL endpoints that hold the data, and creating thematic maps to assist the precision farming. Each thematic map in our case consists of a set of fields that belong to one farmer, along with the water bodies in close proximity, that are taken from OSM water-bodies[9] dataset, and the protected areas that either intersect the fields or are close to the fields, that are retrieved by the Natura2000[10] dataset. Each field consists of raster cells, which are square polygons that act as building blocks and have all the measures (fertilization, vigor) for the specific part of the field. Providing such a map is crucial in precision farming for the following reasons:

- Highlight the parts of fields close to water bodies or protected areas that follow specific legislation for fertilization.
- Provide color maps for each field in respect to specific measure values, to monitor the progression of the crops.
- Provide statistical charts over each field and the farm as a whole, to assist in management decisions.

The Android client of Sextant is fully utilized in LEO, as it was designed to run on any Android device, such as tablets and mobile phones. This is important for promoting precision farming techniques and providing low cost solutions to small farms, that cannot afford the fully automated vehicles mounted with sensors, which have build-in software to assist precision farming.

9.4 Demonstration

In this section we will analyze one thematic map that is created with Sextant to assist in precision farming for a demo farm from the LEO project, that consists of eleven fields. We have created a Strabon endpoint that holds the three datasets mentioned in the previous section, that were published using the tool GeoTriples[23], and links between these datasets that were created with the use of Silk[27, 33]. This demo map consists of 13 layers and one chart, which are the following:

- Layer 1: user input data (POIs and AOIs).

[9] `http://openstreetmapdata.com/data/water-polygons`
[10] `http://natura2000.eea.europa.eu`

- Layer 2-12: the fields of the demo farm that consist of their respective raster cells with the measurements.
- Layer 13: this is the "alert" layer that shows which fields are close to water bodies or protected areas.
- Chart: a chart that depicts the average "vigor" values for each field of the demo farm.

In Figure 9.1 we present the overview of the map. When zoomed out the map depicts two pink polygons that represent the boundaries of the fields that are close to water bodies or protected areas, and eleven pins that represent the fields. The pins are the result of clustering strategies that apply to raster cells when we are zoomed out of the detail level they represent. Zooming in has the opposite result, as the big clusters start to decompose into smaller ones and when the zoom level is appropriate to depict the actual raster cells, then no clustering is applied and we can view the original data.

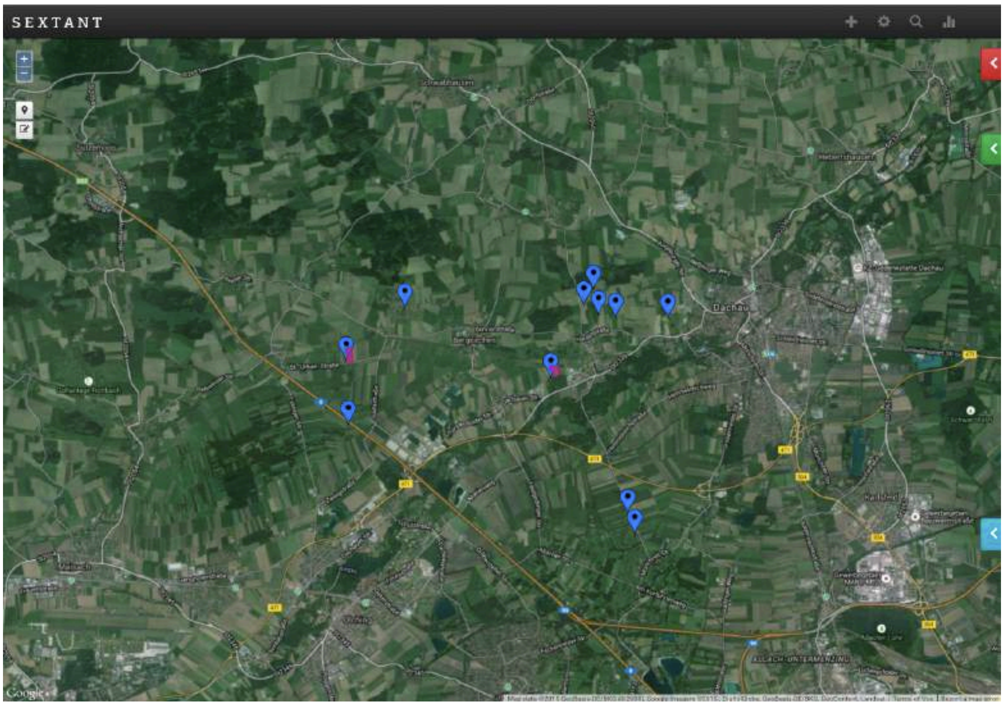


Figure 9.1: The demo farm map

Zooming to "field37" results in a detail level appropriate for depicting raster cells and thus no clustering is applied to the data. In Figure 9.2(a) we have a view of "field37" and its raster cells. If we click in one of the raster cells we have the pup-up with that holds the detailed information for the specific part of the field. As shown in the Figure 9.2(b), each raster cell belongs to a specific field (title of the pop-up), has an identifier and two measurements. The vigor measure is the result of analyzing EO data and crop models to produce a value for this part of the field that shows the condition of the crop. Since raster cells are square polygons we do not have an actual representation of the field perimeter. The CV value is a measure to value what percentage of the raster cell is actually part of the field.

Figure 9.2: The "field37" from the demo farm map

When we zoom to one of the two pink polygons, which represent field boundaries that are close to water bodies of protected areas, we can see the polygon along with the underlying layer that represents the actual field. In Figure 9.3 we show that "field58" is one of the fields that the farmer must take into consideration because of the fertilization legislation that applies for areas near water bodies and protected areas. If we click on the pink polygon that represent the boundary of "field58" we can view a pop-up window with all the available information for the field and the OSM feature or Natura2000 area close by. In this case we can see that the identifier, the label and the name of the field along with the close to water body that has an OSM identifier and the name "Maisach", which is the river that passes by the north edge of the field.

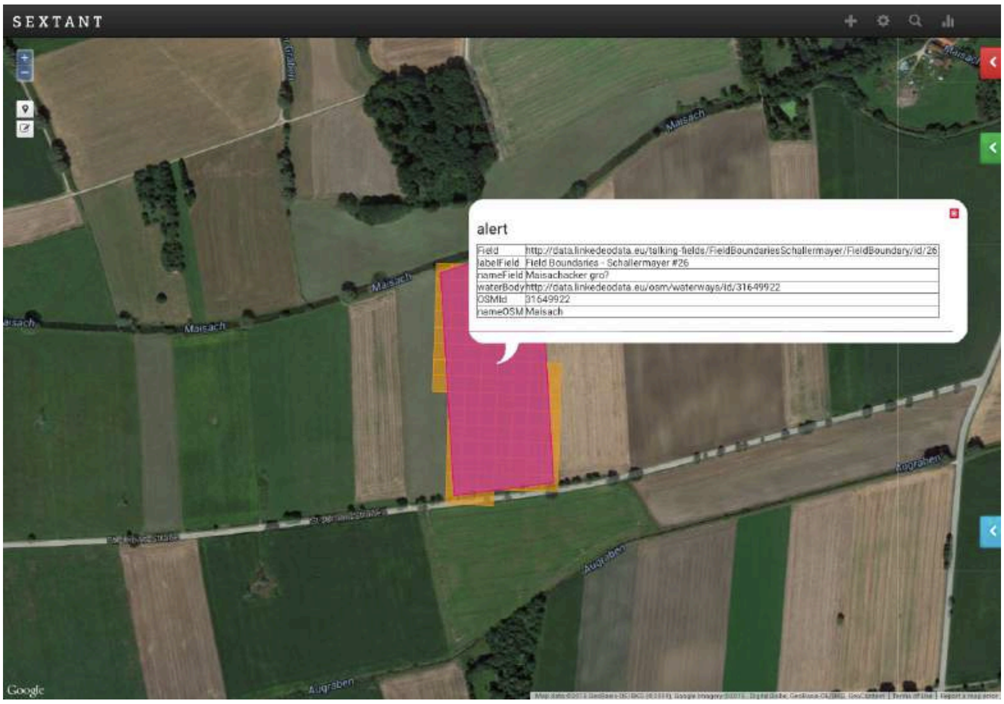


Figure 9.3: The "alert" layer

One of Sextant's more powerful tools is the colorization of a layer's features according to the value of a measurement. This tool allows us to create color maps to visualize measurement variations in one layer. We will use the tool for this demonstration to show how we can create *color fields* to depict the variation of the measurements between raster cells in the field. This feature is enabled for KML layers. In our case each field is the result of a SPARQL query, but the KML translator is responsible for creating a KML file with the results of the SPARQL query, allowing the use of this feature also for query layers. To create this colored layer we click the respective button (fifth in the row for each layer) in the layers panel. In the modal that appears we select the name of the feature we want to focus on, and then we create five intervals, each with a respective color representation, as shown in Figure 9.4.

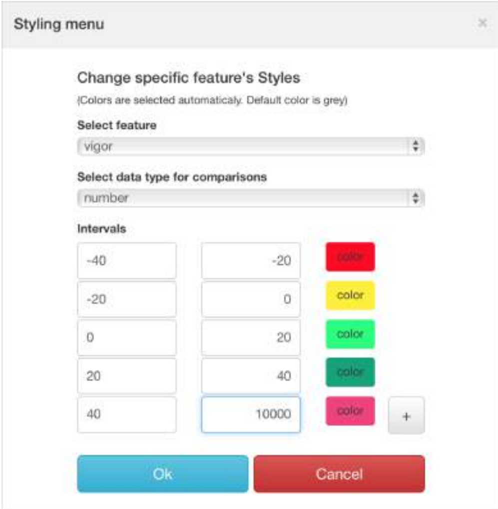


Figure 9.4: Feature coloring modal for "field37"

In Figure 9.5 we demonstrate how we use this tool in the demo farm scenario. We select again "field37", but now we add color filtering for the measure "vigor". The result is colorization of each raster cell of the field, with a color that represents the interval in which the actual value of "vigor" for this raster cell belongs. The color panel in the upper left corner of the map, shows all the coloring information to help in the realization of the color map.

Figure 9.5: Feature coloring for "field37"

This *color field* shows vital information for the crop growth and assists the farmer on identifying which parts of the field are in good or bad condition to deal with them accordingly. For example raster cells with low values of "vigor" will need more fertilizer than the others. In this manner the farmer can make plans for the amount of fertilizer he will need and how to spread it throughout the field.

The demo farm map also consists of one chart. The chart depicts the average "vigor" values for each field of the farm, as depicted in Figure 9.6. This chart can be used to compare the average "vigor" value of the fields to the respective average values of previous years for the same field, or between fields. We can use this information to determine for example if a new fertilization technique that we applied this year has better results than the one from previous years, or compare two fields with different crops and determine if some types of crop are better suited for the geographic region of the fields.

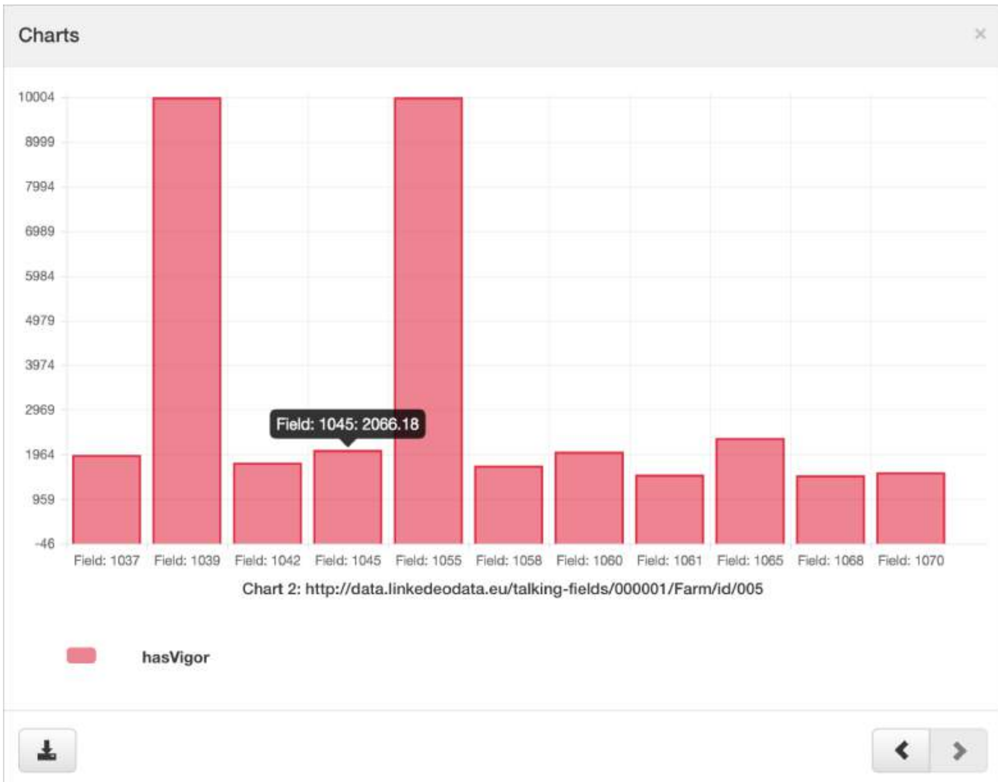


Figure 9.6: Chart with the average "vigor" values for the fields of the demo farm

To produce this chart we select to create a new chart from the menu bar and then follow the steps as presented in Figure 9.7. We provide the URI of the Strabon endpoint that holds the data, fixate the "Farm" dimension, select the farm with identifier "005", which is the demo farm, select the measure "vigor" and create the bar chart.

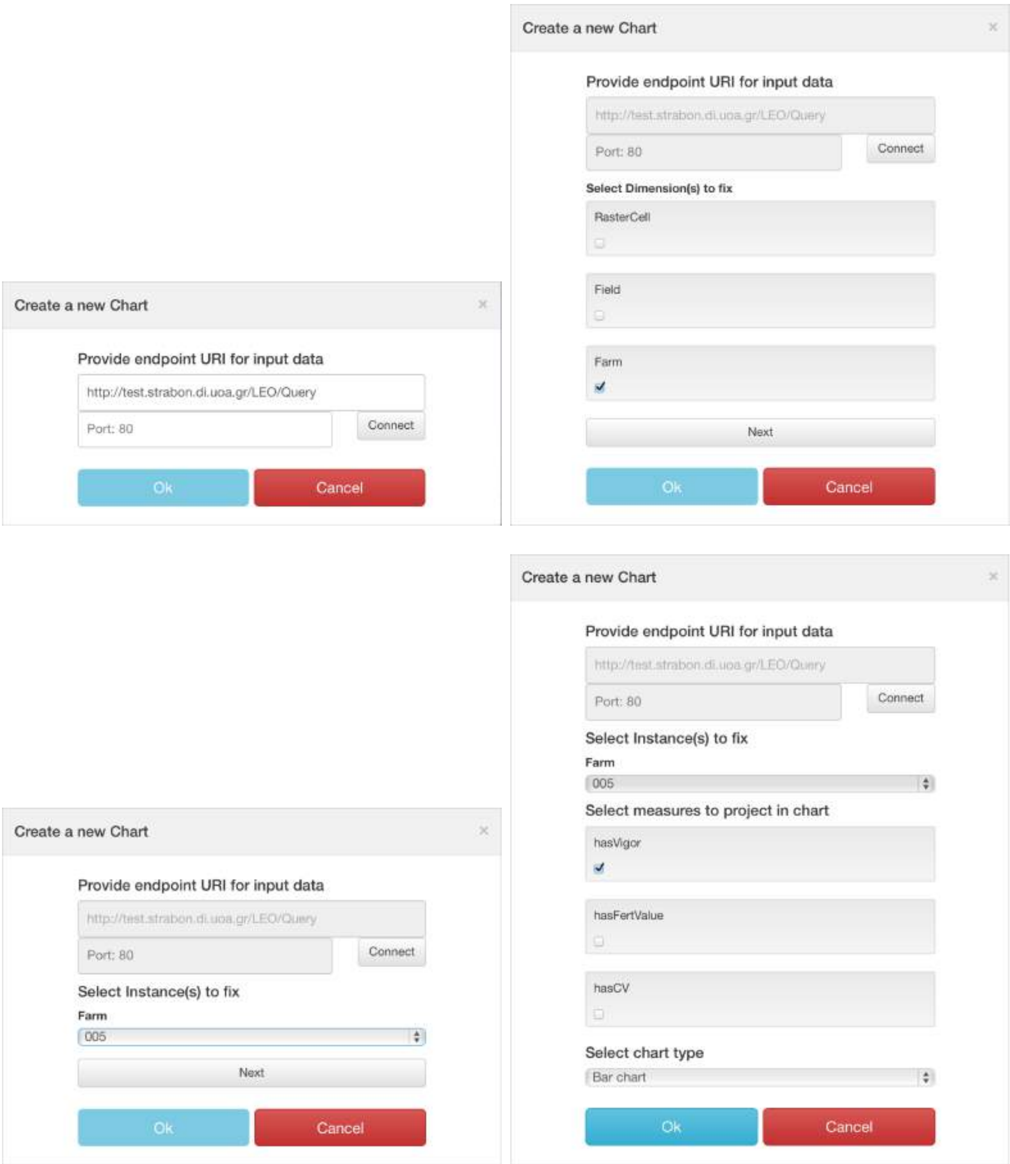


Figure 9.7: Steps for creating the chart for the demo farm map

Chapter 10
Conclusions and Future Work

In this thesis we presented Sextant, an application for visualizing and exploring linked geospatial and temporal data. We tried to remedy the shortcomings of the original version and created a user-friendly interface that assists non-expert users on using semantic web technologies.

We implemented two clients in parallel, the web-client that is designed to be cross-platform and cross-browser, and a mobile-client that runs on Android 4.3 and up. The main objective of the clients was to hide all the communication with the server through SPARQL queries, under smart features in the interface that undertake the communication in a transparent way to the user. The second crucial objective was to design an interface that can convince users to adopt the semantic web technologies by presenting the benefits of the linked open geospatial web. To achieve that we designed a smart, interactive, responsive and modern interface that assists the user in every step on using all the available functionality and make the experience of using Sextant a pleasant journey in the world of semantic web.

Thematic maps are enhanced with metadata that provide a complete legend for the map and also allow for map search implementations. We also presented how we can use this application to combine data from different sources into one map and shown how we can take advantage of the time dimension to adjust our maps using the Timeline.

Apart from using known data sources, we demonstrated how we can use Sextant to search in catalogues, and use the results to create layers. In this manner users are not limited to the resources of datasets they are familiar with, but can discover new datasets that are useful to their domain and use them to create thematic maps. From our experience with the GEOSS data core, this functionality has not yet reached its full potential, but we hope to have better support for the datasets in these catalogues in the future.

Moreover, in an attempt to close the gap between the currently available tools for visualizing linked geospatial data and the GIS platforms we added support for the most promising GIS file formats and introduced data analyzing techniques such as color maps and statistical charts.

In the future we would like to make Sextant more robust and versatile by improving existing aspects and adding more functionality.

To assist non-expert users even more we plan to integrate a new panel that would present the data of a given SPARQL endpoint and allow users to explore its contents through a tree structure, select specific instances with their properties and create layers out of them on the map. This form of query builder gives the power of creating detailed layers without the requirement of knowledge of any semantic web technology.

In the direction of statistical maps, we would like to support more aggregate functions when we need to combine low level measurements into higher dimension charts, and also allow the creation of charts on the map. Production of statistical maps is a key feature for interpreting the underlying elementary data in a qualitative manner. Statistical maps are heavily used by agencies, such as Eurostat[1], the leading provider of high quality statistics in Europe, and the United States Census Bureau[2], the leading provider of quality data about the nation's people and economy.

Map or cartographic generalization[35] is another very desirable and helpful feature for visual analytics, which is essentially a process of abstraction and reduction of the complexity of a map. Map generalization is the process of selecting and simplifying a feature's representation in a way that is appropriate to the scale and the purpose of a map. For example, having a view of the world, it would be undesirable to display villages, that would be otherwise useful to display (e.g., when one zooms into a very specific area). As another example, a map for bicyclists will emphasize a different selection of the roads as opposed to a map targeted at drivers of trucks or lorries. In this case, it is also useful to support clustering techniques aiming at computing groups of features of the same type and displaying an aggregated view of them while losing with respect to location precision.

It is also very important for a tool for visualizing linked spatiotemporal data to offer the ability to combine spatiotemporal information present in different SPARQL endpoints into a single layer of a map. This requirement calls for the development of federated query processing systems for query languages such as GeoSPARQL and stSPARQL. The availability of such a system would turn Sextant into an even more powerful tool for leveraging the linked spatiotemporal data cloud.

[1] `http://ec.europa.eu/eurostat/`
[2] `http://www.census.gov/`

Appendix A
Queries used for Sextant functionality

To support Sextant functionalities the communication with the SPARQL endpoints is made in the final step using various SPARQL queries that provide results used by the client of the application accordingly. In this appendix we provide all the queries that are being used for the client-server communication to support the functionality for loading a map, predefined queries, search for maps, and chart creation.

A.1 Queries to support map load

To load an existing map in Sextant a number of queries are posed to the map registry to retrieve all the information needed to reconstruct the map. First we need to retrieve all the metadata and then the list of the layers of the map and for each layer retrieve all the metadata.

```
Map Metadata:
-------------
SELECT ?title ?creator ?license ?theme ?createDate ?modifyDate ?bbox
WHERE {
    GRAPH <http://geo.linkedopendata.gr/map/mapId> {
        <map>   <hasTitle>      ?title .
        <map>   <hasCreator>    ?creator .
        <map>   <hasLicense>    ?license .
        <map>   <hasTheme>      ?theme .
        <map>   <hasCreateDate> ?createDate .
        <map>   <hasModifyDate> ?modifyDate .
        <map>   <hasGeometry>   ?bbox .
    }
}

List of layers:
---------------
SELECT ?list
WHERE {
    GRAPH <http://geo.linkedopendata.gr/map/mapId> {
        <map>   <hasOrderedList> ?list .
```

```
    }
}

Layers' metadata:
-----------------
SELECT ?layerId
WHERE {
    GRAPH <http://geo.linkedopendata.gr/map/mapId> {
        <layer> <hasId> ?layerId .
    }
}

SELECT ?name ?query ?endpoint ?file ?styleP ?styleL
       ?styleIcon ?scaleIcon ?temporal ?imageBox
WHERE {
    GRAPH <http://geo.linkedopendata.gr/map/mapId> {
        <layerId>   <hasName>           ?name .
        <layerId>   <producedByQuery>   ?q .
        ?q          <hasValue>          ?query .
        ?q          <derivedBy>         ?e .
        ?e          <hasURI>            ?endpoint .
        <layerId>   <hasKML>            ?file .
        <layerId>   <hasPolyStyle>      ?styleP .
        <layerId>   <hasLineStyle>      ?styleL .
        <layerId>   <hasIconStyle>      ?styleIcon .
        <layerId>   <hasIconScale>      ?scaleIcon .
        <layerId>   <isTemporal>        ?temporal .
        <layerId>   <hasImageBox>       ?imageBox .
    }
}
```

Then we retrieve for each chart their metadata.

```
Charts' metadata:
-----------------
SELECT ?type ?query ?endpoint
WHERE {
    GRAPH <http://geo.linkedopendata.gr/map/mapId> {
        <chartId>   <hasType>           ?type .
        <chartId>   <producedByQuery>   ?q .
```

```
        ?q          <hasValue>          ?query .
        ?q          <derivedBy>         ?e .
        ?e          <hasURI>            ?endpoint .
    }
}
```

The results of all the queries are combined and parsed on the server side and a final report is sent to the client to reconstruct the map.

A.2 Queries to support predefined queries feature

To retrieve the list of predefined queries in a SPARQL endpoint and present the list of descriptions to the user, the following query is posed.

```
SELECT ?query ?description ?temporal
WHERE {
    ?x    rdf:type         <predefinedQuery> .
    ?x    <hasText>        ?query .
    ?x    <hasDescription> ?description .
    ?x    <isTemporal>     ?temporal .
}
```

A.3 Queries to support map search

To search for maps in a given registry, we pose the following query to the SPARQL endpoint.

```
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
SELECT ?mapId ?title ?creator ?license ?theme ?description
WHERE {
    ?mapId    <hasTitle>       ?title .
    ?mapId    <hasCreator>     ?creator .
    ?mapId    <hasLicense>     ?license .
    ?mapId    <hasTheme>       ?theme .
    ?mapId    <hasDescritpion> ?description .
    ?mapId    <hasGeometry>    ?geom .

    FILTER regex(?title, "givenTitle", "i") .
```

```
    FILTER regex(?creator, "givenCreator", "i") .
    FILTER regex(?license, "givenLicense", "i") .
    FILTER regex(?theme, "givenTheme", "i") .
    FILTER (geof:sfIntersects(?geom,
            "givenExtent"^^<http://www.opengis.net/ont/geosparql#wktLiteral>)) .
}
```

A.4 Queries to support statistical chart creation

To support the creation of statistical charts over an enabled dataset we have seen that Sextant provides an intuitive interface that assists even non-expert users. To achieve that, we hide all the SPARQL queries that support the communication with the underlying SPARQL endpoint. In this section we show the queries that are automatically created, according to the user input, in order to create a chart.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX qb: <http://purl.org/linked-data/cube#>
PREFIX exqb: <http://geo.linkedopendata.gr/statistics/ontology/>

Retrieve all the Dimensions in the dataset:
-------------------------------------------
SELECT ?dim ?order ?c
WHERE {
    ?dim rdf:type qb:DimensionProperty.
    ?dim qb:order ?order.
    ?dim exqb:position ?pos.

    ?s ?dim ?o.
    BIND (
        IF (?pos = "subject",  ?s , ?o) AS ?instance
    ).
    ?instance rdf:type ?c.
}

Retrieve Free Dimensions:
-------------------------
SELECT ?dim ?c
WHERE {
    ?dim rdf:type qb:DimensionProperty.
```

```
    ?dim qb:order ?order.

    ?s ?dim ?o.
    BIND (
        IF (?pos = "subject",  ?s , ?o) AS ?instance
    ).
    ?instance rdf:type ?c.
    ?instance ?dim+ ?u.
    ?u rdf:type ?uC.
    FILTER (?order >= maxOrder_from_user_fixed_dimensions &&
            ?c != fixedRepresentedType && ?c != ?uC).

}

Retrieve Fixed Dimensions:
--------------------------
SELECT ?dim ?c
WHERE {
    ?dim rdf:type qb:DimensionProperty.
    ?dim qb:order ?order.

    ?s ?dim ?o.
    BIND (
        IF (?pos = "subject",  ?s , ?o) AS ?instance
    ).
    ?instance rdf:type ?c.
    ?instance ?dim+ ?u.
    ?u rdf:type ?uC.
    FILTER (?order <= maxOrder_from_user_fixed_dimensions &&
           (?c = fixedRepresentedType || ?c = ?uC ) ).
}

Retrieve all Measures:
----------------------
SELECT ?measure
WHERE {
    ?measure rdf:type qb:MeasureProperty.
}
```

```
Retrieve the instances of o Fixed Dimension:
--------------------------------------------
SELECT ?instance
WHERE {
    ?s fixedDimensionProperty  ?o.
    fixedDimensionProperty  exqb:position ?pos.
    BIND (
        IF (?pos = "subject",  ?s , ?o) AS ?instance
    ).
    ?instance rdf:type fixedRepresentedType.
}

Retrieve Static Part of the self-adjusting query:
-------------------------------------------------
SELECT DISTINCT ?static
WHERE {
    ?s <hasStaticPart> ?static .
}
```

Bibliography

[1] Open Geospatial Consortium. Geography Markup Language (GML) Encoding Standard. OpenGIS Implementation Standard, 2007. Available from: http://portal.opengeospatial.org/files/?artifact_id=20509.

[2] Open Geospatial Consortium. OpenGIS Implementation Standard for Geographic information - Simple f eature access - Part 1: Common Architecture. OpenGIS Implementation Standard, 2010. Available from: http://portal.opengeospatial.org/files/?artifact_id=25355.

[3] Open Geospatial Consortium. OGC GeoSPARQL - A geographic query language for RDF data. OGC Candidate Implementation Standard, 02 2012.

[4] G. A. Atemezing and R. Troncy. Towards a linked-data based visualization wizard, 2014.

[5] Sören Auer, Jens Lehmann, and Sebastian Hellmann. LinkedGeoData: Adding a Spatial Dimension to the Web of Data. In *International Semantic Web Conference*, pages 731--746, 2009.

[6] BBN Technologies Inc. Parliament$^{\mathrm{TM}}$: A High-Performance Triple Store, SPARQL Endpoint, and Reasoner, 2009.

[7] Konstantina Bereta, Charalampos Nikolaou, Manos Karpathiotakis, Kostis Kyzirakos, and Manolis Koubarakis. SexTant: Visualizing Time-Evolving Linked Geospatial Data. In *International Semantic Web Conference (Posters & Demos)*, pages 177--180, 2013.

[8] Konstantina Bereta, Panayiotis Smeros, and Manolis Koubarakis. Representation and querying of valid time of triples in linked geospatial data. In *ESWC*, volume 7882 of *LNCS*, pages 259--274, 2013.

[9] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems*, 5(3):1--22, 2009.

[10] Chris J. Date, Hugh Darwen, and Nikos A. Lorentzos. *Temporal data and the relational model*. Elsevier, 2002.

[11] Josep Maria Brunetti Fernéandez, Sören Auer, and Roberto Garcia. The linked data visualization model. In *International Semantic Web Conference (Posters & Demos)*, 2012.

[12] George Garbis, Kostis Kyzirakos, and Manolis Koubarakis. Geographica: A benchmark for geospatial rdf stores (long version). In Harith Alani, Lalana Kagal, Achille Fokoue, Paul Groth, Chris Biemann, JosianeXavier Parreira, Lora Aroyo, Natasha Noy, Chris Welty, and Krzysztof

Janowicz, editors, *The Semantic Web – ISWC 2013*, volume 8219 of *Lecture Notes in Computer Science*, pages 343--359. Springer Berlin Heidelberg, 2013.

[13] Claudio Gutierrez, Carlos Hurtado, and Ro Vaisman. Temporal RDF. In *In European Conference on the Semantic Web (ESWC'05)*, pages 93--107, 2005.

[14] Claudio Gutierrez, Carlos A. Hurtado, and Alejandro Vaisman. Introducing time into RDF. *IEEE Transactions on Knowledge and Data Engineering*, 19(2):207 --218, Feb. 2007.

[15] Charalampos Kontoes, Iphigenia Keramitsoglou, Ioannis Papoutsis, Nicolas I. Sifakis, and Panteleimon Xofis. National scale operational mapping of burnt areas as a tool for the better understanding of contemporary wildfire patterns and regimes. *Sensors*, 13(8):11146--11166, 2013.

[16] Manolis Koubarakis. Linked Open Earth Observation Data: The LEO Project, 2014.

[17] Manolis Koubarakis, Manos Karpathiotakis, Kostis Kyzirakos, Charalampos Nikolaou, and Michael Sioutis. Data Models and Query Languages for Linked Geospatial Data. In Thomas Eiter and Thomas 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.

[18] Manolis Koubarakis, Charalambos Kontoes, Stefan Manegold, Manos Karpathiotakis, Kostis Kyzirakos, Konstantina Bereta, George Garbis, Charalampos Nikolaou, Dimitrios Michail, Ioannis Papoutsis, Themistoklis Herekakis, Milena Ivanova, Ying Zhang, Holger Pirk, Martin Kersten, Kallirroi Dogani, Stella Giannakopoulou, and Panayiotis Smeros. Real-time wildfire monitoring using scientific database and linked data technologies. In *EDBT*, 2013.

[19] Manolis Koubarakis and Kostis Kyzirakos. Modeling and querying metadata in the semantic sensor web: The model stRDF and the query language stSPARQL. In *ESWC*, 2010.

[20] Manolis Koubarakis and Kostis Kyzirakos. Modeling and Querying Metadata in the Semantic Sensor Web: The Model stRDF and the Query Language stSPARQL. In *ESWC*, pages 425--439, 2010.

[21] Kostis Kyzirakos, Manos Karpathiotakis, and Manolis Koubarakis. Strabon: a semantic geospatial dbms. In *The Semantic Web--ISWC 2012*, pages 295--311. Springer, 2012.

[22] Kostis Kyzirakos, Manos Karpathiotakis, and Manolis Koubarakis. Strabon: A Semantic Geospatial DBMS. In *ISWC*, volume 7649 of *LNCS*, pages 295--311. Springer, 2012.

[23] Kostis Kyzirakos, Ioannis Vlachopoulos, Dimitrianos Savva, Stefan Manegold, and Manolis Koubarakis. GeoTriples: a Tool for Publishing Geospatial Data as RDF Graphs Using R2RML Mappings. In *6th International Workshop on the Foundations, Technologies and Applications of the Geospatial Web*, 2014.

[24] Alexander de León, Victor Saquicela, Luis M. Vilches, Boris Villazón-Terrazas, Freddy Priyatna, and Oscar Corcho. Geographical Linked Data: a Spanish Use Case. In *I-SEMANTICS*. ACM, 2010.

[25] Alexander de León, Filip Wisniewki, Boris Villazón-Terrazas, and Oscar Corcho. Map4rdf - Faceted Browser for Geospatial Datasets. PMOD workshop, 2012.

[26] A. McBratney, B. Whelan, and T. Ancev. Future Directions of Precision Agriculture, 2015.

[27] Axel-Cyrille Ngonga Ngomo. Orchid - reduction-ratio-optimal computation of geo-spatial distances for link discovery. In *Proceedings of ISWC 2013*, 2013.

[28] Charalampos Nikolaou, Kallirroi Dogani, Kostis Kyzirakos, and Manolis Koubarakis. Sextant: Browsing and Mapping the Ocean of Linked Geospatial Data. In *ESWC (Satellite Events)*, pages 209--213, 2013.

[29] OGC. GeoSPARQL - A geographic query language for RDF data, November 2010.

[30] Open Geospatial Consortium. OGC KML. OGC® Implementation Standard, 2008. Available from: `http://portal.opengeospatial.org/files/?artifact_id=27810`.

[31] Open Sahara. uSeekM library.

[32] Matthew Perry. *A Framework to Support Spatial, Temporal and Thematic Analytics over Semantic Web Data*. PhD thesis, Wright State University, 2008.

[33] Panayiotis Smeros and Manolis Koubarakis. Tools for interlinking EO data and their evaluation. Del. 3.2, FP7 project LEO, 2015.

[34] Richard T. Snodgrass, editor. *The TSQL2 Temporal Query Language*. Kluwer, 1995.

[35] Robert Weibel. Generalization of spatial data: Principles and selected algorithms. In Marc Kreveld, Jürg Nievergelt, Thomas Roos, and Peter Widmayer, editors, *Algorithmic Foundations of Geographic Information Systems*, volume 1340 of *Lecture Notes in Computer Science*, pages 99--152. Springer Berlin Heidelberg, 1997.

[36] B.M. Whelan and A.B. McBratney. Definition and Interpretation of potential management zones in Australia. In *11th Australian Agronomy Conference*, 2003.