Querying Linked Geospatial Data with Incomplete Information

C. Nikolaou and M. Koubarakis

Department of Informatics and Telecommunications
National and Kapodistrian University of Athens, Greece
`charnik@di.uoa.gr`

Abstract. Linked geospatial data has recently received attention, as researchers and practitioners have started tapping the wealth of geospatial information available on the Web. Incomplete geospatial information, although appearing often in the applications captured by such datasets, is not represented and queried properly due to the lack of appropriate data models and query languages. We discuss our recent work on the model RDF^{i}, an extension of RDF with the ability to represent property values that exist, but are unknown or partially known, using constraints, and an extension of the query language SPARQL with qualitative and quantitative geospatial querying capabilities. We demonstrate the usefulness of RDF^{i} in geospatial Semantic Web applications by giving examples and comparing the modeling capabilities of RDF^{i} with the ones of related Semantic Web systems.

Keywords: linked geospatial data, incomplete information, RDF

1 Introduction

Linked data is a new research area which studies how one can make RDF data available on the Web, and interconnect it with other data with the aim of increasing its value for everybody [4]. The resulting "Web of data" has recently started being populated with geospatial data. A representative example of such efforts is LinkedGeoData[1] where OpenStreetMap data is made available as RDF and queried using the declarative query language SPARQL [2]. With the recent emphasis on open government data, some of it encoded already in RDF[2], portals such as LinkedGeoData demonstrate that the development of useful Web applications might be just a few SPARQL queries away. The recent paper [9] by our group addresses many research topics and relevant questions that deserve the attention of researchers in the area of linked geospatial data.

In the context of the research agenda presented in [9], we have developed stSPARQL [17], an extension of the query language SPARQL for querying linked

---

[1] `http://linkedgeodata.org/`

[2] `http://data.gov.uk/linked-data/`

geospatial data. The geospatial component of stSPARQL has been fully implemented in our open source system Strabon[3] which also supports GeoSPARQL, the recent proposed standard by OGC (Open Geospatial Consortium) for querying geospatial data expressed in RDF. Strabon is currently being used to query linked data describing sensors in the context of project SemsorGrid4Env[4] [16] and linked earth observation (EO) data in the context of project TELEIOS[5] [12].

A significant aspect of querying linked geospatial data that has not been addressed yet is querying linked geospatial data with incomplete information [11]. Incomplete information, although appearing often in applications captured by such datasets, is not represented or queried properly due to the lack of appropriate data models and query languages. For example, a wildfire monitoring and management application, developed by us in TELEIOS, requires the integration of multiple, heterogeneous data sources, some of them available on the Web, with data of varying quality and varying temporal and spatial scales. As a result, incomplete information needs to be represented in stRDF and queried by stSPARQL.

In this paper we address the problem of representing and querying *incomplete geospatial information* in RDF using the RDF^{i} framework that we have recently developed in [19]. RDF^{i} is a framework that extends RDF with the ability to represent property values that exist, but are unknown or partially known, using constraints. RDF^{i} is a general framework for the representation of incomplete information of this kind and it can be employed in various application domains, such as temporal and spatial. In this paper, we concentrate on the spatial domain only and demonstrate the modeling capabilities of RDF^{i} and the querying capabilities of our extension of SPARQL which is based on stSPARQL.

The organization of the paper is as follows. Section 2 introduces the RDF^{i} framework. Section 3 describes the kinds of linked geospatial data that we need to represent in the wildfire monitoring application of TELEIOS. Then, Section 4 demonstrates the RDF^{i} framework giving examples motivated from that application of TELEIOS. Finally, Section 5 compares the expressive power of RDF^{i} with related semantic web systems, while Section 6 concludes our work.

The paper is mostly informal and uses examples from the wildfire monitoring application of TELEIOS. Even in the places where the paper becomes formal, we do not give any detailed technical results for which the interested reader is directed to [13, 14, 19] and the survey paper [9].

2 The RDF^{i} framework

The RDF^{i} framework developed by us in [19] (where "i" stands for "incomplete") is an extension of the RDF framework addressing an important kind of incomplete information that has so far been ignored in the context of RDF;

---

[3] `http://www.strabon.di.uoa.gr/`

[4] `http://www.semsorgrid4env.eu/`

[5] `http://www.earthobservatory.eu/`

representation of values that *exist but are unknown or partially known*. RDF^i extends RDF with the ability to define a new kind of literals for each datatype. These literals are called *e-literals* ("e" comes from the word "existential") and can be used to represent values of properties that *exist but are unknown or partially known*. Such information is abundant in recent applications where RDF is being used (e.g., sensor networks, the modeling of geospatial information, etc.). In RDF^i, e-literals are allowed to appear only in the object position of triples.

Previous research on incomplete information in databases and knowledge representation has shown that in many applications, having the ability to state *constraints* about values that are partially known is a very desirable feature and leads to the development of very expressive formalisms [5,8]. In the spirit of this tradition, RDF^i allows partial information regarding property values represented by e-literals to be expressed by a quantifier-free formula of a first-order *constraint language* $\mathcal{L}$. Thus, RDF^i extends the concept of an RDF graph to the concept of an RDF^i *database* which is a pair (G, ϕ) where G is an RDF graph possibly containing triples with e-literals in their object positions, and ϕ is a quantifier-free formula of $\mathcal{L}$.

The semantics for RDF^i databases and SPARQL query evaluation has been defined following ideas from the incomplete information literature [5, 6]. The semantics defines the set of possible RDF graphs corresponding to an RDF^i database and the fundamental concept of certain answer for SPARQL query evaluation over an RDF^i database.

The well-known concept of *representation system* from the seminal paper of [6] has been transferred to the case of RDF^i. It has been shown in [19] that CONSTRUCT queries without blank nodes in their templates and using only the operators AND, UNION, and FILTER or the restricted fragment of graph patterns corresponding to the well-designed patterns of [1] can be used to define a representation system for RDF^i. Last, [19] defines the fundamental concept of certain answer to SPARQL queries over RDF^i databases and presents an algorithm for its computation.

3 Linked geospatial data in the wildfire monitoring application of TELEIOS

The wildfire monitoring application of TELEIOS concentrates on the development of solutions for real time hotspot and active fire front detection, and burnt area mapping. Technological solutions to both of these cases require integration of multiple, heterogeneous data sources with data of varying quality and varying temporal and spatial scales. Some of the data sources are streams (e.g., streams of EO images) while others are static geo-information layers (e.g., land use/land cover maps) providing additional evidence on the underlying characteristics of the affected area.

In what follows, we briefly describe some of the datasets used by the National Observatory of Athens (NOA) that is leading the wildfire monitoring application of TELEIOS.

Hotspot maps. NOA operates a MSG/SEVIRI[6] acquisition station and receives raw satellite images every 15 minutes. These images are processed using image processing algorithms to detect the existence of hotspots. Information related to hotspots is stored in ESRI shapefiles and KML files. These files hold information about the date and time of image acquisition, cartographic X, Y coordinates of detected fire locations, the level of reliability in the observations, the fire radiative power assessed, and the observed fire area. NOA receives similar hotspot shapefiles covering the geographical area of Greece from the European project SAFER (Services and Applications for Emergency Response).

Burnt area maps. From project SAFER, NOA also receives ready-to-use accumulated burnt area mapping products in polygon format, projected to the EGSA87 reference system[7]. These products are derived daily using the MODIS satellite and cover the entire Greek territory. The data formats are ESRI shapefiles and KML files with information relating to date and time of image acquisition, and the mapped fire area.

Corine Land Cover data. The Corine Land Cover project is an activity of the European Environment Agency which is collecting data regarding land cover (e.g., farmland, forest) of European countries. The Corine Land Cover nomenclature uses a hierarchical scheme with three levels to describe land cover:

- The first level consists of five items and indicates the major categories of land cover on the planet, e.g., forests and semi-natural areas.
- The second level consists of fifteen items and is intended for use on scales of 1:500,000 and 1:1,000,000 identifying more specific types of land cover, e.g., open spaces with little or no vegetation.
- The third level consists of forty-four items and is intended for use on a scale of 1:100,000, narrowing down the land use to a very specific geographic characterization, e.g., burnt areas.

The land cover of Greece is available as an ESRI shapefile that is based on the Corine Land Cover nomenclature.

Coastline geometry of Greece. An ESRI shapefile that describes the geometry of the coastline of Greece is available.

In [15,17] we discuss in great detail how we can query linked geospatial data such as the above using the model stRDF and the query language stSPARQL. In this work we concentrate on the representation and querying of linked geospatial data with incomplete information. This is presented in the following section by giving examples motivated from the wildfire monitoring application of TELEIOS.

---

[6] MSG refers to Meteosat Second Generation satellites, and SEVIRI is the instrument which is responsible for taking infrared images of the earth.

[7] EGSA87 is a 2-dimensional projected coordinate reference system that describes the area of Greece.

4 Incomplete geospatial information in the wildfire monitoring application of TELEIOS

This section motivates our approach towards extending RDF with the ability to represent and query incomplete information.

As mentioned in Section 3, NOA receives satellite images for the entire Greek fire season on a 15-minute basis from the SEVIRI infrared imager of a Meteosat Second Generation satellite. After the images are processed for georeferencing, they are analyzed by specialized image processing software to detect hotspots (i.e., regions of the image corresponding to geographic regions that are probably on fire). Processing of images results in the generation of shapefiles representing hotspots as point-vectors.

The following is a list of triples (namespaces are omitted) that gives an example of the kind of representation that is currently used by NOA for representing these hotspots and making them available as linked data to relevant public authorities.

```
hotspot1 type Hotspot .
fire1 type Fire .
hotspot1 correspondsTo fire1 .
fire1 occuredIn region1 .
region1 hasGeometry "x = 24.825668 ∧ y = 35.310643"^^SemiLinearPointSet .
```

The above list of triples is a graph in the model stRDF of [10] which extends RDF with the ability to represent geometries over $\mathbb{Q}^k$ that change over time following the paradigm of constraint databases [7]. In stRDF, geometries and valid times of triples are expressed using Boolean combinations of linear constraints that are given as literals of type `SemiLinearPointSet` defined in [10]. *Semi-linear point sets* are the subsets of $\mathbb{Q}^k$ defined by Boolean combinations of linear constraints. The above graph represents *definite* information; it states that there is a hotspot (`hotspot1`) and that the corresponding fire (`fire1`) takes place at the point $(24.825668, 35.310643) \in \mathbb{Q}^2$.

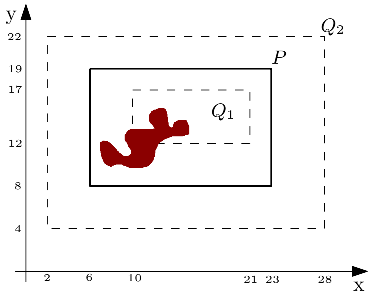

Fig. 1. Rectangles mentioned in the examples

In practice, due to the technical weaknesses of the instruments attached to satellites and inherent distortions of the algorithms applied on satellite images for knowledge extraction, the extracted spatial information can only be indefinite. For example, the SEVIRI imager has medium resolution and therefore each image pixel representing a hotspot corresponds to a 3km by 3km rectangle in geographic space. Accordingly, NOA represents hotspots as points in geographic space using the center of the corresponding rectangle.

In this case, another useful representation of the real world situation that corresponds to a hotspot would be to state that there is a geographic region with unknown exact coordinates where a fire is taking place, and that region is included in a known 3km by 3km rectangle. This real world situation can be represented by an RDF^i database as shown in the following example.

Example 1. The following is an RDF^i database encoding information about a detected hotspot.

```
hotspot1 type Hotspot .
fire1 type Fire .
hotspot1 correspondsTo fire1 .
fire1 occuredIn _R1 .
```

_R1 NTPP "$x \geq 6 \wedge x \leq 23 \wedge y \geq 8 \wedge y \leq 19$"

Fire `fire1` (red area of Figure 1) is asserted to have taken place inside region _R1. _R1 is an e-literal of datatype `SemiLinearPointSet` and is asserted to be inside the rectangle formed by the points $(6, 8)$ and $(23, 19)$ (rectangle P of Figure 1)[8]. This is stated with a constraint expressed in the language PCL (Polygon Constraint Language), a first-order constraint language that allows us to represent topological properties for polygons. NTPP is the "non-tangential-proper-part" relation of RCC-8 [24]. In general, constraints in PCL can be used to express qualitative and quantitative spatial information about regions in $\mathbb{Q}^2$.

The example shows that e-literals are like existentially quantified variables in first-order logic or Skolem constants. E-literals can be used to represent values of properties that *exist* but are *unknown* or *partially known* (e.g., by constraining the value of an e-literal).

RDF^i databases like the one of Example 1 consist of two parts: a graph (i.e., a set of triples) and a *global constraint*. Global constraints can in general be quantifier-free formulae of some first-order constraint language. RDF^i databases are syntactic devices for the representation of incomplete information. An RDF^i database is semantically equivalent to a set of possible RDF graphs that represent all the possible ways the domain of application could have been according to our incomplete information. One can find all the possible RDF graphs represented by an RDF^i database as follows: *a*) find an assignment to e-literals that satisfies the global constraint and *b*) substitute these values for the e-literals in the RDF^i database.

---

[8] For sake of readability of the examples, we chose to use small, integer numbers instead of real geographic coordinates.

For example, the RDF graph shown below is one of the possible RDF graphs corresponding to the RDF$^{\text{i}}$ database of Example 1.

```
hotspot1 type Hotspot .
fire1 type Fire .
hotspot1 correspondsTo fire1 .
fire1 occuredIn "x ≥ 10 ∧ x ≤ 21 ∧ y ≥ 12 ∧ y ≤ 17" .
```

RDF$^{\text{i}}$ databases can be queried using the well-known query language SPARQL. Example 2 below demonstrates a query in SPARQL.

Example 2. Let us consider the query "Find all fires that have occurred in a region which is a non-tangential proper part of rectangle Q_1 of Figure 1" over the database of Example 1. In the extension of SPARQL we consider, this query can be expressed as follows:

```
SELECT ?F
WHERE  {
            ?F type Fire .
            ?F occuredIn ?R .
            FILTER ( NTPP(?R, "x ≥ 10 ∧ x ≤ 21 ∧ y ≥ 12 ∧ y ≤ 17") )
}
```

The version of SPARQL we consider extends FILTER expressions of standard SPARQL [23] allowing also expressions of a first-order constraint language, such as PCL, for constraining the values of spatial variables. These expressions have a functional-like syntax and are interpreted in the underlying first-order language. For example, the global constraint of the RDF$^{\text{i}}$ database of Example 1 would be specified in a FILTER expression as

```
NTPP(?R, "x ≥ 6 ∧ x ≤ 23 ∧ y ≥ 8 ∧ y ≤ 19")
```

to constrain the value of the spatial variable `?R`.

What is the answer to the query of Example 2? If we examine the database of Example 1 (Figure 1), we can see that the answer should be *conditional* [6]. We cannot say for sure whether `fire1` satisfies the requirements of the query because the information in the database is indefinite (the exact geometry of `_R1` is not known). Fire `fire1` qualifies only in the possible graphs where `_R1` is a non-tangential proper part of the rectangle mentioned in the query. For every object that qualifies as an answer, the query answering procedure should also provide a *condition* characterizing this set of possible graphs. Following the ideas of conditional tables from [6], this answer can be represented by the following set of *conditional mappings* (see [19] for a formal definition):

?F	Condition
`fire1`	_R1 NTPP "$x \geq 10 \wedge x \leq 21 \wedge y \geq 12 \wedge y \leq 17$"

Conditional mappings are different from standard SPARQL mappings [22] in the sense that they map variables to constants only if a condition holds. Thus, they are reminiscent of conditional tuples in the conditional table model of [5]. In RDF$^{\mathrm{i}}$, the basic concept of triple is also defined to be conditional.

Example 3. If we wanted to have an RDF$^{\mathrm{i}}$ database as the answer to a query like the one of Example 2, then we would have queried the RDF$^{\mathrm{i}}$ database of Example 1 using the CONSTRUCT query form of SPARQL as follows:

```
CONSTRUCT { ?F type Fire }
WHERE     {
             ?F type Fire .
             ?F occuredIn ?R .
             FILTER ( NTPP(?R, "x ≥ 10 ∧ x ≤ 21 ∧ y ≥ 12 ∧ y ≤ 17") )
}
```

The answer to this query would be an RDF$^{\mathrm{i}}$ database containing *conditional triples* adhering to the query template (i.e., {`?F type Fire`}). The template is instantiated for each conditional mapping from the evaluation of the graph pattern of the query, and the resulting triple together with the condition of the mapping form a conditional triple in the resulting database. Therefore, the answer to query of Example 3 consists of the following conditional triple:

```
fire1 type Fire [_R1 NTPP "x ≥ 10 ∧ x ≤ 21 ∧ y ≥ 12 ∧ y ≤ 17"] .
```

The e-literals in the above answer (i.e., _R1) are implicitly constrained by the global constraint of the original database, i.e., constraint

```
_R1 NTPP "x ≥ 6 ∧ x ≤ 23 ∧ y ≥ 8 ∧ y ≤ 19".
```

In some cases the user might know that the information in the database is incomplete. Thus, she might wish to find all values that *certainly* satisfy some qualification. This is the well-known notion of certain answer in the incomplete databases literature [5] and it is demonstrated in the following example.

Example 4. Let us consider the query of Example 3 again and rephrase it to "Find fires that have *certainly* occurred in a region which is a non-tangential proper part of rectangle Q_2 of Figure 1". In the version of SPARQL we consider, this query would be expressed as follows:

```
CERTAIN CONSTRUCT { ?F type Fire }
WHERE  {
           ?F type Fire .
           ?F occuredIn ?R .
           FILTER ( NTPP(?R, "x ≥ 2 ∧ x ≤ 28 ∧ y ≥ 4 ∧ y ≤ 22") )
}
```

Inspecting Figure 1, it is obvious that `fire1` satisfies the query unconditionally. Hence, the certain answer contains the following RDF triple

```
fire1 type Fire .
```

In contrast to Example 3 where the answer to the CONSTRUCT query is an RDF^{i} database, the answer to a CONSTRUCT query with a CERTAIN operator, like the one of Example 4 above, is an RDF graph. This is anticipated since a certain answer can not contain conditional information.

5 Expressive power of RDF^{i}: An informal comparison

In this paper we gave examples of the use of RDF^{i} in geospatial applications. Thus, it would be interesting to compare the expressive power that RDF^{i} gives us to other recent works that use Semantic Web data models and languages for geospatial applications.

When equipped with a constraint language like PCL (or TCL[9]) [19], RDF^{i} goes beyond the proposals of [10, 17] and [20] that cannot express incomplete geospatial information. Incomplete geospatial information as it is studied in this paper can also be expressed in spatial description logics [18, 21]. For efficiency reasons, spatial DL reasoners such as RacerPro[10] and PelletSpatial[11] have opted for separating spatial relations from standard DL axioms as we have done by separating graphs and constraints. Since RDF graphs can be seen as DL ABoxes with atomic concepts only, all the results of this paper can be trivially transferred to the relevant subsets of spatial DLs and their reasoners.

In the following we concentrate on the reasoner PelletSpatial since it is a more recent proposal than RacerPro and discuss how RDF^{i} is related to the recently proposed Semantic Web technologies of [3, 26].

PelletSpatial [25] is a hybrid spatial reasoner that provides RCC-8 and OWL 2 reasoning and querying capabilities. In PelletSpatial, spatial relations are separated from OWL 2 relations providing a hybrid reasoner for both spatial and thematic data. Spatial relations are managed as an RCC-8 constraint network. Conjunctive query answering in PelletSpatial requires two phases: *a*) evaluating spatial query atoms over the constraint network by employing a path-consistency algorithm, and *b*) further constraining the set of bindings such that the non-spatial query atoms are satisfied.

Compared to the RDF^{i} framework, PelletSpatial corresponds to RDF^{i} databases with a conjunction of TCL-constraints as a global constraint. Compared to our extension of SPARQL, the query language of PelletSpatial computes certain answers for SPARQL queries using only the operators AND and FILTER with conjunctions of TCL-constraints allowed as expressions in FILTER graph patterns. The representational and querying power of RDF^{i} when $\mathcal{L}$ is PCL is greater than the one of PelletSpatial since PCL is a language more expressive than TCL. However, PelletSpatial offers OWL representation and reasoning that is not offered by RDF^{i}.

---

[9] TCL is like PCL but without constants, that is, TCL can express topological constraints only between variables.

[10] `http://www.racer-systems.com/`

[11] `http://clarkparsia.com/pellet/spatial/`

A more general and formal approach to modeling spatial information is [26] that proposes an abstracted graph-based data model and query language with which any subset of first-order predicate logic (FOPL) (e.g., modal, description logic) can be associated. For the case of spatial information, a substrate can play the role of a geometric substrate, called SBox. SBox deals with spatial datatypes (e.g., polygons) the geometry of which can be described using an appropriate FOPL, inheriting also its formal semantics for satisfiability, entailment, etc. The authors investigate four options for representing and querying spatial information: use (i) an ABox, (ii) a map substrate, (iii) a spatial ABox, (iv) an ABox and RCC substrate.

Finally, [3] proposes SOWL, an extension of OWL, to represent spatial qualitative and quantitative information employing the RCC-8 topological relations, cardinal direction relations, and distance relations. To reason about spatial relations, a set of SWRL rules are implemented in the Pellet reasoner.

6 Conclusions

This work stressed the inability of semantic web data models and query languages to manage linked geospatial data with incomplete information. Motivated by a real application in which representation and querying of incomplete information is inherent, it demonstrated through the use of many examples how the RDF^i framework and an extension of the query language SPARQL [19] can be employed for active fire front detection and burnt area mapping in the context of the EU project TELEIOS.

Acknowledgments

This work has been funded by the FP7 project TELEIOS (257662).

References

1. M. Arenas and J. Pérez. Querying semantic web data with SPARQL. In *PODS*, pages 305–316, 2011.
2. S. Auer, J. Lehmann, and S. Hellmann. Linkedgeodata: Adding a spatial dimension to the web of data. In *ISWC'09*, pages 731–746, 2009.
3. S. Batsakis and E. Petrakis. SOWL: spatio-temporal representation, reasoning and querying over the semantic web. In *I-SEMANTICS*, 2010.
4. C. Bizer, T. Heath, and T. Berners-Lee. Linked data-the story so far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22, 2009.
5. G. Grahne. *The Problem of Incomplete Information in Relational Databases*, volume 554 of *LNCS*. Springer Verlag, 1991.
6. T. Imielinski and W. Lipski. Incomplete Information in Relational Databases. *JACM*, 31(4):761–791, 1984.
7. P. C. Kanellakis, G. M. Kuper, and P. Z. Revesz. Constraint Query Languages. In *PODS*, pages 299–313, 1990.

8. M. Koubarakis. Database models for infinite and indefinite temporal information. *Inf. Syst.*, 19(2):141–173, 1994.
9. M. Koubarakis, M. Karpathiotakis, K. Kyzirakos, C. Nikolaou, and M. Sioutis. Data Models and Query Languages for Linked Geospatial Data. In T. Eiter and T. Krennwallner, editors, *Reasoning Web. Semantic Technologies for Advanced Query Answering*, volume 7487 of *Lecture Notes in Computer Science*, pages 290–328. Springer Berlin / Heidelberg, 2012.
10. M. Koubarakis and K. Kyzirakos. Modeling and querying metadata in the semantic sensor web: The model stRDF and the query language stSPARQL. In *ESWC*, 2010.
11. M. Koubarakis, K. Kyzirakos, M. Karpathiotakis, C. Nikolaou, M. Sioutis, S. Vassos, D. Michail, T. t. Herekakis, C. Kontoes, and I. Papoutsis. Challenges for Qualitative Spatial Reasoning in Linked Geospatial Data. In *Proceedings of IJCAI 2011 Workshop on Benchmarks and Applications of Spatial Reasoning*, 2011.
12. M. Koubarakis, K. Kyzirakos, M. Karpathiotakis, C. Nikolaou, S. Vassos, G. Garbis, M. Sioutis, K. Bereta, D. Michail, C. Kontoes, I. Papoutsis, T. Herekakis, S. Manegold, M. Kersten, M. Ivanova, H. Pirk, Y. Zhang, M. Datcu, G. Schwarz, C. Dumitru, D. E. Molina, K. Molch, U. D. Giammatteo, M. Sagona, S. Perelli, T. Reitz, E. Klien, and R. Gregor. TELEIOS: A Database-Powered Virtual Earth Observatory. *PVLDB*, 5(12):2010–2013, 2012.
13. M. Koubarakis, C. Nikolaou, and V. Fisikopoulos. Theoretical results on query processing for RDF/SPARQL with time and space. Del. 2.3, TELEIOS, 2011.
14. M. Koubarakis et al. A data model and query language for an extension of RDF with time and space. Del. 2.1, TELEIOS project, 2011.
15. K. Kyzirakos, M. Karpathiotakis, G. Garbis, C. Nikolaou, K. Bereta, M. Sioutis, I. Papoutsis, T. Herekakis, D. Michail, M. Koubarakis, and H. Kontoes. Real Time Fire Monitoring Using Semantic Web and Linked Data Technologies. In *ISWC'12*, 2012.
16. K. Kyzirakos, M. Karpathiotakis, and M. Koubarakis. Developing Registries for the Semantic Sensor Web using stRDF and stSPARQL. In *SSN*, volume 668, 2010.
17. K. Kyzirakos, M. Karpathiotakis, and M. Koubarakis. Strabon: A Semantic Geospatial DBMS. In *ISWC'12*, 2012.
18. C. Lutz and M. Miličić. A tableau algorithm for description logics with concrete domains and general tboxes. *J. Autom. Reason.*, 38:227–259, April 2007.
19. C. Nikolaou and M. Koubarakis. Incomplete Information in the Semantic Web. 2012 (submitted to a conference).
20. Open Geospatial Consortium Inc. GeoSPARQL - A geographic query language for RDF data. OGC, 2010.
21. Ö. L. Özçep and R. Möller. Combining DL-Lite with Spatial Calculi for Feasible Geo-thematic Query Answering. In *Description Logics*, 2012.
22. J. Pérez, M. Arenas, and C. Gutierrez. Semantics and complexity of SPARQL. *ACM Trans. Database Syst.*, 34(3):1–45, 2009.
23. E. Prud'hommeaux and A. Seaborne. SPARQL Query Language for RDF. W3C Recommendation 15 Jan. 2008.
24. D. A. Randell, Z. Cui, and A. G. Cohn. A spatial logic based on regions and connection. In *KR*, pages 165–176, 1992.
25. M. Stocker and E. Sirin. PelletSpatial: A hybrid RCC-8 and RDF/OWL reasoning and query engine. In *OWLED*, 2009.
26. M. Wessel and R. Moller. Flexible software architectures for ontology-based information systems. *JAL*, 7(1):75–99, 2009.