A Transparent Semantic Enablement Layer for the Geospatial Web *

Krzysztof Janowicz, Sven Schade, Arne Bröring,
Carsten Keßler, Christoph Stasch, Patrick Maué, Thorsten Diekhof

Institute for Geoinformatics, University of Münster, Germany
janowicz|sven.schade|arneb|carsten.kessler|staschc|pajoma|thorsten.diekhof
@uni-muenster.de

Abstract. Building on abstract reference models, the Open Geospatial Consortium (OGC) has established standards for storing, discovering, and processing geographical information. These standards act as basis for the implementation of specific services and Spatial Data Infrastructures (SDI). Research on geo-semantics plays an increasing role to support complex queries and retrieval across heterogeneous information sources, as well as for service orchestration, semantic translation, and on-the-fly integration. So far, this research targets individual solutions or focuses on the Semantic Web, leaving the integration into SDI aside. What is missing is a shared and transparent semantic enablement layer for Spatial Data Infrastructures which also integrates reasoning services known from the Semantic Web. Focusing on Sensor Web Enablement (SWE), we outline how Spatial Data Infrastructures in general can benefit from such a semantic enablement layer. Instead of developing new semantically enabled services from scratch, we propose to create profiles of existing services that implement a transparent mapping between the OGC and the Semantic Web world.

1 Motivation

Developing and deploying Spatial Data Infrastructures based on OGC services is attractive for two reasons. First, these services are well standardized and their implementations can be tested for conformity. Second, the OGC has defined a top-level interface standard called OWS Common [1] defining main aspects that are shared by most OGC web services. Frequent testbeds investigate, report on, and discuss the interoperability between specific services. Both aspects ease the integration of services into Spatial Data Infrastructures, make them more adaptable, and form the basis for their orchestration [2].

Services, however, are not built for their own sake but to encapsulate data or processing models. To exchange data between services, i.e., to make them interoperable, they have to share common schemas or translate between them. For

* This work is a substantially extended and rewritten version of the poster abstract: Janowicz, K., Keßler, C., Bröring, A., Stasch, C., and Schade, S.: *Towards Semantic Enablement for Spatial Data Infrastructures.* 4th European Conference on Smart Sensing and Context (EuroSSC 2009), Guildford, UK, September 16–18, 2009.

example, if one processing service requires a string representing wind direction as input and was developed with a *wind blows from* conceptualization in mind, a second service offering wind direction observations as strings, but based on a *wind blows to* conceptualization, can still act as input source [3]. The OGC standards guarantee interoperability on a syntactic level. Services can exchange data if they agree on names and types for their inputs, outputs, and operations. Whether data exchanged between services can be interpreted in a meaningful way is not covered by the specifications. For example, a Web Processing Service (WPS) [4] can be used to compute the dispersion of a gas plume caused by a factory fire based on wind direction observations delivered by a Sensor Observation Service (SOS) [5]. Both services need to share a common understanding of wind direction to compute meaningful results [3, 6]; otherwise the simulated dispersion plume would point in the opposite direction. Hence, the challenge is to establish semantic interoperability, i.e., the ability of services to exchange data in a meaningful[1] way and with a minimum of human intervention [7, 8]. For instance, mapping national geodata models to the specifications of the INSPIRE initiative for a European SDI requires formal specifications of the national data models (as well as the INSPIRE model) [9]. So far, the mappings between these models have to be specified and tested *outside* of OGC services. There is no way to check on-the-fly whether a dataset has been consistently mapped or whether it contains contradictions. To perform such checks, the application schemas and the contained feature type definitions have to be downloaded manually in order to perform common reasoning tasks.

The remaining paper is structured as follows. First, we introduce related work on SWE and geo-semantics related to OGC services and SDI. Next, we discuss the idea of a horizontal and vertical semantic enablement layer and its integration with OGC services. We stick to the gas plume dispersion example throughout the paper as a running scenario. We conclude our work by summarizing the proposed approach and point to further work.

2 Related work

The Open Geospatial Consortium provides standards to geospatially enable the Web with the goal of making spatial data and services accessible in an interoperable way for all kinds of applications. The Sensor Web Enablement [10] initiative as one focus area of OGC's specification program develops standards to integrate sensors into Spatial Data Infrastructures. SWE incorporates different data models for describing sensors (SensorML [11]) as well as gathered sensor data (Observations & Measurements [12]). Additionally, web service interfaces are defined which make these models available to enable the discovery and controlling of sensors, the retrieval of sensor data, as well as alerts in case of particular events within a sensor network. A central role in this framework plays the Sensor Observation Service (SOS) [5] which provides a standardized interface

[1] This is still a working definition as it does not define when a combination of data is considered to be *meaningful*.

for the pull-based access of both archived and near-realtime sensor observations and metadata.

Over the last years, work on semantics [13] and geo-ontologies has focused on several challenges towards establishing semantic interoperability between OGC services. This includes work on the role of ontology for spatio-temporal databases [14], the notion of semantic reference systems and the grounding of geographical categories [15, 16], semantics-based and context-aware retrieval of geographic information [17–20], ontology alignment [21], as well as work on Semantic Geospatial Web services [22] and their chaining [23]. This research has lead to several new services and tools such as ConceptVISTA[2] for ontology creation and visualization, the SWING Concept Repository[3], the SIM-DL similarity server and Protégé plug-in[4], or the semantically-enabled Sensor Observation Service SemSOS [24]. Opposed to work on Spatial Data Infrastructures, these services do not share common interfaces but are isolated solutions which lack a binding to each other and partially also to existing OGC services. For instance, while the SIM-DL server can compute the similarity of geographic feature types, it uses an extended version of the Description Logics Interface Group (DIG) protocol for communication and the Web Ontology Language (OWL) for knowledge representation[5]. In contrast, OGC services such as the Web Feature Service (WFS), which could be chained with the similarity server, use GetCapabilities requests and the Geographic Markup Language (GML).

3 Towards a Transparent Semantic Enablement Layer

To integrate Semantic Web services into SDI we propose a transparent Semantic Enablement Layer (SEL) for OGC services. It resides on top of recent standards and considers the following three challenges: (1) How to link data encodings and service protocols to formal specifications stored within ontologies? (2) How to manage and maintain these ontologies? (3) How to incorporate reasoning services known from the Semantic Web?

Based on these challenges we can derive functionalities, which should be provided by the SEL (see Table 1). For further structuring, we categorized atomic functionalities into four conformance classes. *Storage* groups functionalities which are required for ontology storage, evolution, and access. The functionality to connect elements of a specific resource, e.g., a GML or RDF data model, with concepts or instances from an ontology is provided by the *Lookup and Retrieval* conformance class. *Reasoning* groups operations about inferring hidden facts as well as adding new ones, while the *Deployment* functionality supports the deployment of OGC services if their data models have been encoded in ontologies. Such deployment includes the generation of a content description for an OGC web service which is advertised in its capabilities document, as well

[2] http://www.geovista.psu.edu/ConceptVISTA/
[3] http://purl.org/net/concepts/
[4] http://sim-dl.sourceforge.net/
[5] This is not only that case for SIM-DL but most reasoners on the Semantic Web.

as an automated creation of descriptions of resources such as feature type serializations using XSD. This functionality ensures explicit linkages between services and content descriptions

Table 1. Overview of SEL run-time functionalities.

Conformance Classes	Operation
Storage	• *createOntology*: creates a new ontology with all its classes and relations inside the repository. • *updateOntology*: registers a new version of an ontology to the repository. • *getConcept, getRelation, getOntology*: returns different types of elements from a registered ontology.
Lookup and Retrieval	• *getModelReference*: returns the appropriate ontology element ID for a given resource ID, e.g., GML Feature ID. • *retrieve*: executes semantic matchmaking between a goal/query and (a) available web service advertisements and (b) feature type definitions.
Reasoning	• *loadOntology*: loads a specific ontology into the reasoner. • *releaseOntology*: removes a specific ontology from the reasoner. • *tells*: inserts a new fact into the knowledge base. • *asks*: returns facts from the knowledge base.
Deployment	• *createCapabilities*: creates content-specific section of an OGC Capabilities Document. • *createFeatureTypeDescription*: creates a GML feature type in XSD format (created file may contain annotations).

We propose to group the functionalities of the conformance classes in two services, the Web Ontology Service (WOS) for managing and accessing ontologies and the Web Reasoning Service (WRS) for providing reasoning functionality within SDIs. Instead of creating new services from scratch, the WOS is defined as a profile of the Web Catalog Service (CS-W) and the WRS as a profile of the Web Processing Service (WPS). This facilitates the integration with existing SDI technologies and simplifies the service orchestration. As WRS and WOS have to follow the OWS Common specification, a major challenge is the mapping between the protocols and representation languages used on the Semantic Web and the OGC world. Note that we do not propose to develop separate reasoners or ontology repositories for SDI but to transparently encapsulate existing Semantic Web solutions by the WOS and WRS.

Components, which support the creation of semantic annotations, i.e., which provide means to link elements from data model to concepts from an ontology, are not considered in this setting. Such components are not used during SEL run-time, but in a previous development phase. Examples of supporting tools are given in [25]. They can be used to generate content for successful lookup. The semantic annotation of OGC services as well as the WOS and the WRS are explained in more detail in the following subsections.

We illustrate the integration of the proposed services into SDI using the gas plume example. We assume that air pollutants peril an important European bird sanctuary, the so called Rieselfelder in Münster (Germany), as well as the surrounding natural reserves. For reasons of simplification, we further assume that a local Sensor Web is already set up and used by a disaster relief organization, i.e., mobile sensors are deployed to monitor air pollutants, wind speed, and wind direction.

3.1 Semantic Annotation

Service operations such as GetCapabilities and data encodings such as SensorML describe the functionality and data offered by a specific OGC service. This includes sensor inputs and outputs in case of SensorML, and a list of contained geographic feature types in case of a Web Feature Service's capabilities document. Currently, the descriptions of these resources are only syntactic, e.g., encoded in GML. If additional descriptions are contained, they mostly consist of plain-text. A first step towards semantic enablement is to annotate these elements of the General Feature Model [26]. Semantic annotation links them to the according classes specified within ontologies.

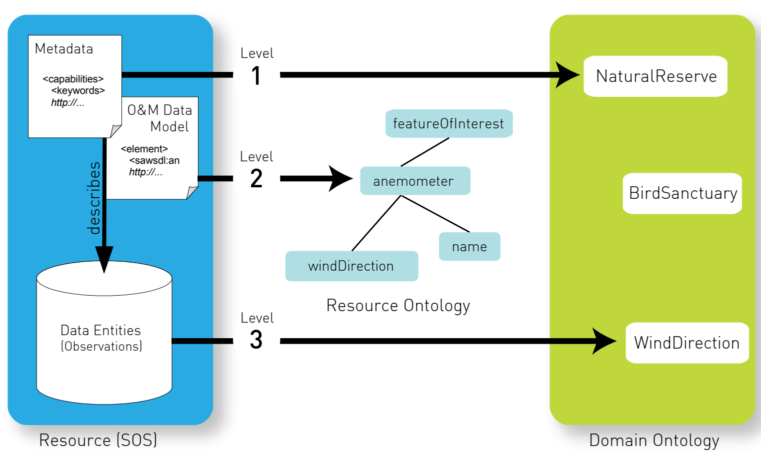

Fig. 1. Levels of annotations; adopted from [27].

Recently, Maué et al. [27] proposed a methodology for the annotation of OGC services. The authors suggest three different levels of (semantic) annotations. They distinguish between the annotation of resource metadata, e.g., an OWS Capabilities Document, of a data model, e.g. a GML Application Schema, and of data entities, such as a GML file. Annotations may point to information source ontologies, a specific form of application ontologies, or to shared domain ontologies. Figure 1 provides an example for annotating an OGC SOS. Level 1 links the keywords of a capabilities document to concepts from a domain ontology, *NaturalReserve* in this case. The connection between the application schema to an application ontology is illustrated in level 2, while linking features to concepts from a domain ontology is depicted at the bottom of the figure (level 3).

```
<InsertObservation service='SOS' version='1.0.0'>
  ...
  <om:Observation>
    ...
    <om:procedure sawsdl:modelReference='[nsKB]:Anemometer01'/>
    <om:observedProperty sawsdl:modelReference='[nsKB]:WindDirection'/>
    <om:featureOfInterest>
      <sa:SamplingPoint gml:id='sampling01'>
        <sa:sampledFeature sawsdl:modelReference='[nsKB]:Rieselfelder'/>
        <sa:position>
          <gml:Point>
            <gml:pos srsName= 'urn:ogc:def:crs:EPSG:4326'>
                7.645 52.034
              ...
      </sa:SamplingPoint>
    </om:featureOfInterest>
    <om:result uom='deg'>42.0</om:result>
...
```

Listing 1.1. Example annotation of a SOS InsertObservation request. *nsKB* is a placeholder for the namespace of the used (populated) ontology.

Listing 1.1 shows an example of an InsertObservation request offered by an SOS. By invoking this operation new sensor observations can be fed into the SOS. For example, the element of the observed property contains an attribute that links to an external resource. Following Maué et al. [27], the links are implemented as SAWSDL model references [28]. The model references point to concept (e.g. *WindDirection*) or instance (e.g., *Anemometer01* and *Rieselfelder*) identifiers from a populated ontology. The formal specifications of these concepts can then be used for tasks such as semantics-based information retrieval. Note that the annotations may refer to concepts or individuals from a resource or domain ontology.

3.2 Web Ontology Service – Managing and Accessing Ontologies

While annotations establish links between elements of data and service models to concepts, individuals, and relations in ontologies, these ontologies need to be stored and managed in repositories [29]. Typically, ontology repositories act as interfaces offering access and URI resolution as well as auxiliary services for querying, visualizing, versioning, and comparing the stored definitions[6]. With regards to SDIs built on OGC services, a decomposition of the functionality into separate services would be more appropriate (comparable to the separation of WFS and WMS). We therefore argue that a Web Ontology Service should provide access, lookup, and retrieval functionalities. A WOS could provide access to definitions from multiple domains ranging from geographic feature types over types of observations, to sensor types. In case of the gas plume example, a particular WOS could store feature types such as *Factory*, *NaturalReserve*, and *InhabitedPlace*, as well as sensor types such as *Anemometer*. By providing access to the formal specifications, a WOS could support a semantic mapping between sensor outputs and the properties of features of interest when registering new sensors or adding their observations to a Sensor Observation Service [6].

A WOS serves two requirements. First, it provides access to formal specifications for the elements annotated in data encodings and service operations. Second, and in conjunction with a Web Reasoning Service, a WOS can be used for semantics-based information retrieval. In this sense, a WOS is a semantically-enabled catalogue supporting information retrieval beyond simple keyword search [17, 18]. Therefore and in conformity with Lieberman et al. [30], we argue that a WOS should be designed as a profile of the OGC Catalogue Service (CS-W) [31]. Thereby, it abstracts from spatial and temporal search, while focusing on thematic aspects. As ontologies in general require specific querying, the Filter Encoding standard [32] requires an according profile. In conjunction with the CS-W it provides novel means to sophisticated retrieval of geospatial data and services. The enhanced filter encoding may even be re-used by other OGC service types, such as WFS or SOS in order to enhance data querying capabilities.

Using the gas plume scenario, Figure 2 illustrates how the WOS can support the transparent gathering of relevant data, e.g., sensor observations. A WOS is queried for all subtypes of *NaturalReserve* which are located within a particular bounding box, e.g., the greater Münster area. To process such a query, the WOS utilizes an associated Web Reasoning Service. The WOS response contains all feature types satisfying the input query, e.g., *Bird Sanctuary*. These types can be used in further discovery tasks to find features of interest affected by the fire and gas plume.

An OWL-Profile for CS-W was suggested recently by Stock et al. [33]. The authors suggest a non-transparent approach, which restricts the format (in their case OWL) and reasoning capabilities (in their case reasoning on description logics). In contrast, we propose a transparent approach which abstracts from a particular inference engine and from specific ontology languages such as OWL, OWL 2.0, Web Service Modelling Language (WSML), or Topic Maps.

3.3 Web Reasoning Service – Bringing Reasoning to SDI

While the Web Ontology Service encapsulates ontologies, a second service has to encapsulate the functionalities defined in the reasoning conformance class. This

[6] Examples of repositories and collaborative tools include work by the Open Ontology Repository Initiative, the NeON Cupboard, OwlSight, Web Protégé, or OWLDiff.

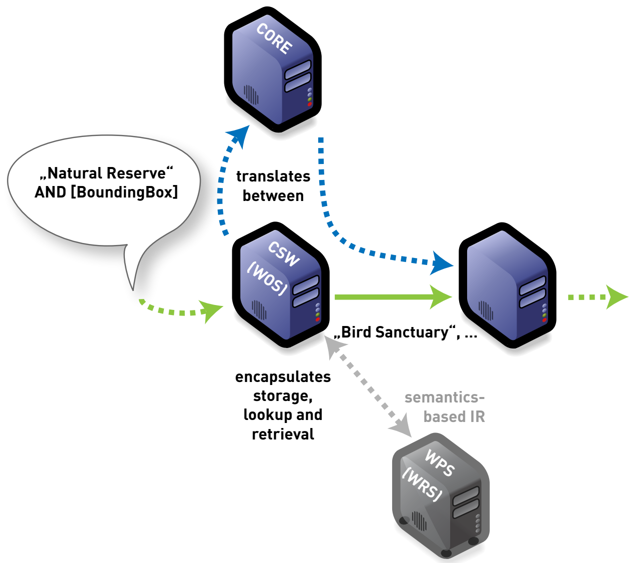

Fig. 2. Transparent integration of existing ontology repositories in to an SDI to support semantics-based lookup and retrieval using the Web Ontology Service as a CSW profile.

service has to bridge between the inference engines as key components of the Semantic (Geospatial) Web and the OGC world. Reasoners are not restricted to subsumption reasoning, but include non-standard inference such as finding the most specific concept, least common subsumer, similarity reasoning [17], as well as context-aware instantiation based on SWRL rules and built-ins [19]. We argue that such a Web Reasoning Service should be developed as a profile of the Web Processing Service specification [4]. Since the WRS should encapsulate semantic web reasoners and make them accessible for SDIs, it has to map in both directions between *DIG tells and asks* calls on the one side and GetCapabilities request and GML on the other side[7].

With respect to Sensor Web Enablement, a WRS could be used to discover appropriate sensors using a feature of interest as query [6]. For instance, a semantically-enabled SDI could automatically choose and register sonic anemometers if the user is interested in data on the dispersion of a gas plume. In case of semantics-based retrieval of feature types [18] as depicted in figure 2, the WRS gives the necessary reasoning power to the WOS.

Figure 3 illustrates how the WRS can be used to incorporate reasoning services into an OGC service chain. With respect to the gas plume example, the WRS encapsulated the SIM-DL similarity server to select features similar to the *Rieselfelder* in the greater Münster area, e.g., the *Wienburgpark*. Next, a SOS is

[7] If the WRS should also encapsulate other ontology languages and their reasoning services, such as WSML and IRIS, it has to implement additional mappings.

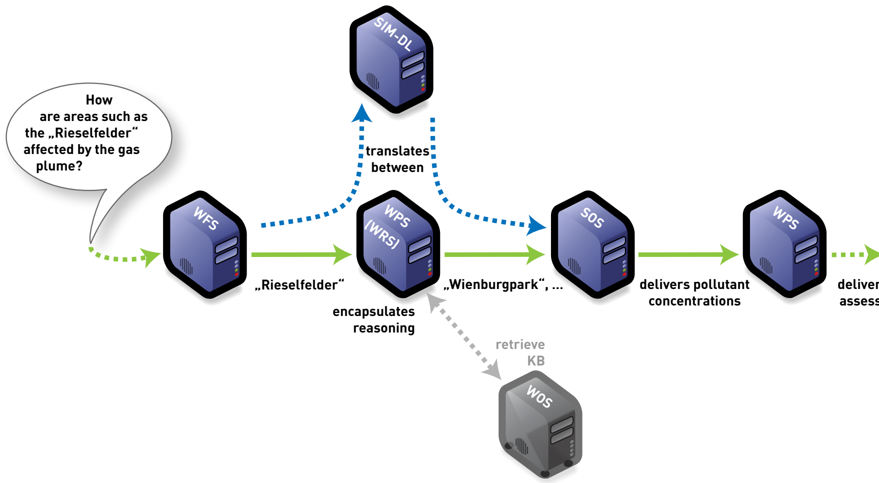

Fig. 3. Transparent integration of existing reasoners in to an SDI using the Web Reasoning Service as a WPS profile.

used to access sensor observations about the potential pollution of these features, and, finally, a web processing service delivers a risk analysis.

4 Conclusions and Further Work

In this paper, we outlined the need for a Semantic Enablement Layer for OGC web services. We argued that such a layer is a prerequisite for semantics-based information retrieval tailored to the user's context, semantic translation, the orchestration of sensors and web services, and finally semantic interoperability. Three steps towards establishing a SEL have been identified. First, data encodings and service protocols have to be linked to formal specifications stored in ontologies using semantic annotations. Second, a service has to be established for managing and maintaining these ontologies. Third, a service has to encapsulate Semantic Web reasoners to integrate them into SDIs. Additional services, such as the WRS can be integrated into SDIs without changing existing clients. The proposed approach generalizes over previously suggested solutions and provides a tight (and transparent) integration into recent SDI developments. We also clearly separate data models (in any encoding) from domain ontologies. This separation acknowledges the distinction between information items from the real world.

While we focused on introducing the need for and components of the Semantic Enablement Layer, the reference implementation of the WOS and WRS is part of the 52°*North* semantics community[8]. Currently, our work on the WRS focuses on the encapsulation of the SIM-DL similarity server and Pellet reasoner

[8] http://www.52north.org/semantics

to make them accessible for OGC services such as the SOS and WFS. A semantic annotation API for the lookup and injection functionality is developed in the *sapience* project[9]. Adding annotations on-the-fly to existing OGC metadata is required as long as the data models are not represented as ontologies within the WOS. In the long term, the functionality described by the deployment conformance class will enable the creation of (parts of) the OGC Capability Documents by the WOS.

An interesting question is whether and to which degree OGC services will co-exist with upcoming semantic Web technologies and especially with linked data infrastructures. While this is difficult to predict, we assume that both approaches do not necessarily exclude each other. For instance, one could think of a micro-SDI for linked data. Examples include recent work on next generation gazetteers [34], a forthcoming linked data serialization of OpenStreetMap [35], or JavaScript reasoners such as JSExplicit[10] which can directly be embedded into web pages to generate context and user-aware information from RDF or OWL data on the fly.

5 Acknowledgments

The presented work is funded by the International Research Training Group on *Semantic Integration of Geospatial Information* (DFG GRK 1498), the DFG *SimCat II* project (DFG Ja1709/2-2), the *SWING* project of the European Commission (FP6-026514), as well as the 52°*North* semantics community which aims at establishing a semantic enablement layer for OGC services.

References

1. Whiteside, A.: OGC Implementation Specification 06-121r3: OGC Web Services Common Specification (2007)
2. Weiser, A., Zipf, A.: Web Service Orchestration of OGC Web Services for Disaster Management. LN in Geoinformation and Cartography, Springer (2007) 239–254
3. Probst, F., Lutz, M.: Giving Meaning to GI Web Service Descriptions. In: 2nd International Workshop on Web Services: Modeling, Architecture and Infrastructure (WSMAI 2004), Porto, Portugal. (2004)
4. Schut, P.: OGC Implementation Specification 05-007r7: OpenGIS Web Processing Service (2007)
5. Na, A., Priest, M.: OGC Implementation Specification 06-009r6: OpenGIS Sensor Observation Service (SOS) (2007)
6. Bröring, A., Janowicz, K., Stasch, C., Kuhn, W.: Semantic challenges for sensor plug and play. In: Web & Wireless Geographical Information Systems, 7 & 8 December 2009, Maynooth, Ireland. (2009; forthcoming)
7. Harvey, F., Kuhn, W., Pundt, H., Bisher, Y., Riedemann, C.: Semantic Interoperability: A Central Issue for Sharing Geographic Information. The Annals of Regional Science **33** (1999) 213–232

[9] http://purl.org/net/sapience/docs/

[10] http://jsexplicit.sourceforge.net

8. Manso, M.A., Wachowicz, M.: GIS Design: A Review of Current Issues in Interoperability. Geography Compass **3**(3) (2009) 1105 – 1124
9. INSPIRE: INSPIRE scoping paper, available from http://www.ec-gis.org/inspire/reports/inspire_scoping24mar04.pdf. Technical report (2004)
10. Botts, M., Percivall, G., Reed, C., Davidson, J.: OGC Sensor Web Enablement: Overview And High Level Architecture. Technical report, OGC (2007)
11. Botts, M.: OGC Implementation Specification 07-000: OpenGIS Sensor Model Language (SensorML) (2007)
12. Cox, S.: OGC Implementation Specification 07-022r1: Observations and Measurements - Part 1 - Observation schema (2007)
13. Kuhn, W.: Geospatial Semantics: Why, of What, and How? Journal on Data Semantics III (2005) 1–24
14. Frank, A.U.: Ontology for spatio-temporal databases. Volume 2520 of LNCS. Springer (2003) 9–77
15. Kuhn, W.: Semantic Reference Systems. International Journal of Geographic Information Science **17**(5) (2003) 405–409
16. Scheider, S., Janowicz, K., Kuhn, W.: Grounding Geographic Categories in the Meaningful Environment. In: Conference on Spatial Information Theory (COSIT 2009). LNCS, Springer (2009, forthcoming)
17. Janowicz, K., Keßler, C., Schwarz, M., Wilkes, M., Panov, I., Espeter, M., Baeumer, B.: Algorithm, Implementation and Application of the SIM-DL Similarity Server. In Fonseca, F.T., Rodriguez, A., Levashkin, S., eds.: Second International Conference on GeoSpatial Semantics (GeoS 2007). Number 4853 in Lecture Notes in Computer Science, Springer (November 2007) 128–145
18. Lutz, M., Klien, E.: Ontology-based retrieval of geographic information. International Journal of Geographical Information Science **20**(3) (March 2006) 233–260
19. Keßler, C., Raubal, M., Wosniok, C.: Semantic rules for context-aware geographical information retrieval. In Barnaghi, P., ed.: European Conference on Smart Sensing and Context, EuroSSC 2009. Volume 5741 of LNCS., Springer (2009) 77–92
20. Schade, S., Klien, E., Maue, P., Fitzner, D., Kuhn, W.: Report on Modelling Approach and Guideline. Deliverable D3.2 of the SWING Project, available from http://swing-project.org/deliverables. Technical report (2008)
21. Cruz, I., Sunna, W.: Structural alignment methods with applications to geospatial ontologies. Transactions in GIS, special issue on Semantic Similarity Measurement and Geospatial Applications **12**(6) (2008) 683–711
22. Andrei, M., Berre, A.J., Costa, L., Duchesne, P., Fitzner, D., Grcar, M., Hoffmann, J., Klien, E., Langlois, J., Limyr, A., Maué, P., Schade, S., Steinmetz, N., Tertre, F., Vasiliu, L., Zaharia, R., Zastavni, N.: Swing: An integrated environment for geospatial semantic web services. In Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M., eds.: 5th European Semantic Web Conference (ESWC 2008). LNCS 5021, Springer (2008) 767–771
23. Lemmens, R., Wytzisk, A., de By, R., Granell, C., Gould, M., van Oosterom, P.: Integrating semantic and syntactic descriptions to chain geographic services. IEEE Internet Computing **10**(5) (2006) 42–52
24. Henson, C.A., Pschorr, J.K., Sheth, A.P., Thirunarayan, K.: SemSOS: Semantic Sensor Observation Service. In: International Symposium on Collaborative Technologies and Systems (CTS 2009). (2009)
25. Grcar, M.: D4.5: Software Module for Semantic Annotation of a Web Service. Deliverable of the SWING Project available from http://swing-project.org/deliverables. Technical report (2008)

26. ISO/TC211: ISO/FDIS 19109: Geographic information – Rules for application schema
27. Maué, P., Schade, S., Duchesne, P.: OGC Discussion Paper 08-167r1: Semantic annotations in OGC standards. Technical report, OGC (2009)
28. Farrell, J., Lausen, H.: Semantic annotations for WSDL and XML schema. W3C recommendation, available from http://www.w3.org/TR/sawsdl/ (2007)
29. Schade, S., Maué, P., Wosniok, C., Finke, T., Heuer, J., Dupke, S., Scherer, F.: D3.3 Ontology Repository with Ontologies. Deliverable of the SWING Project, available from http://swing-project.org/deliverables. Technical report (2008)
30. Lieberman, J., Pehle, T., Morris, C., Kolas, D., Dean, M., Lutz, M., Probst, F., Klien, E.: Geospatial Semantic Web Interoperability Experiment Report. Technical report, Open Geospatial Consortium (2006)
31. Nebert, D., Whiteside, A., Vretanos, P.: OGC Implementation Specification 07-006r1: OpenGIS Catalogue Services Specification (2007)
32. ISO/TC211: ISO/DIS 19143: Geographic information – Filter encoding
33. Stock, K., Small, M., Ou, Y., Reitsma, F.: OGC Discussion Paper 09-010 - OWL Application Profile of CSW. Technical report, Open Geospatial Consortium (2009)
34. Keßler, C., Janowicz, K., Bishr, M.: An agenda for the next generation gazetteer: Geographic information contribution and retrieval. In: International Conference on Advances in Geographic Information Systems 2009 (ACM SIGSPATIAL GIS 2009). (2009; forthcoming)
35. Auer, S., Lehmann, J., Hellmann, S.: LinkedGeoData – Adding a spatial Dimension to the Web of Data. (2009; submitted to ISWC 2009)