Semantic Challenges for Sensor Plug & Play

Arne Bröring, Krzysztof Janowicz, Christoph Stasch, Werner Kuhn

Institute for Geoinformatics, University of Muenster, Germany
arneb|janowicz|staschc|kuhn@uni-muenster.de

Abstract. The goal of the Sensor Web Enablement (SWE) initiative of the Open Geospatial Consortium (OGC) is the definition of web service interfaces and data encodings to make sensors discoverable, taskable and accessible on the World Wide Web. The SWE specifications enable a standardized communication and interaction with arbitrary types of sensors and sensor systems. The central concepts within OGC's Sensor Web architecture are sensors, observations and features of interest. Sensors and their observations can be registered and stored through the Sensor Observation Service (SOS) to make them accessible for clients. So far, mechanisms are missing which support a semantic matching between features of interest stored in a database and referred to by an observation. The same applies for the matching between observations as sensor outputs and the properties of the features of interest. By taking a use case from disaster management, we outline the challenges and demonstrate how semantically annotated SWE data models and service interfaces support semantic matching. The result is a roadmap towards a semantically enabled sensor plug & play within the Sensor Web.

1 Introduction

Recent improvements in sensor technology and lower prices change the way we collect and process massive amounts of data in realtime. Thus, the usage of sensors increases in applications ranging from environmental monitoring over early warning systems and precision agriculture up to personal health and performance monitoring [1,2,3]. The Sensor Web Enablement initiative of the Open Geospatial Consortium (OGC) aims at standardizing the discovery, exchange, and processing of sensor data as well as their tasking. Therefore, the Sensor Web Enablement initiative defines a framework of data models and encodings for describing sensors and their observations as well as a suite of web service interfaces leveraging these models and encodings [4]. While the OGC has already done substantial work in defining protocols and service interfaces to enable syntactical interoperability, semantic enablement is still in an early stage [5]. Recently, Sheth et al. [6] coined the term Semantic Sensor Web to combine Sensor Web technology with the Semantic Web. A first step towards the realization of the Semantic Sensor Web has been presented by Henson et al. [7] by introducing a semantic enabled Sensor Observation Service called SemSOS which semantically annotates the service responses.

In contrast, our approach focuses on the semantic annotation of service requests for adding new sensors and observations to an Sensor Observation Service. The correct semantic matching from sensor inputs and outputs to the observed property of the features of interest as well as the matching between a real world entity observed by a sensor and the feature of interest have to be assured. So far, these matchings have to be established and maintained manually by the service provider. In particular, this problem appears when multiple observation suppliers publish their content via the same service instance. A sensor services can be set up for certain geographic regions and various sensors of different types can register at these services and upload their observations. Taking into account mobile sensors moving in and out of this region the problem becomes even more pressing. An automatic plug & play of sensors which realizes a correct mapping of the different Sensor Web concepts is needed. In this paper, we present a detailed analysis of the challenges of adding new sensors to the Sensor Web and publishing their gathered observations. The work will serve as a roadmap towards semantically enabled plug & play for the Sensor Web.

The remaining paper is structured as follows. Section 2 describes the basis of this work by introducing the Sensor Web and the role of ontologies. The following section 3 describes an emergency scenario to illustrate our work. We then provide an in-depth analysis of existing challenges for registering sensors and publishing observations. The paper closes with conclusions and an outlook to future work.

2 Background

The idea of the Sensor Web is to standardize the web based discovery, exchange, and processing of sensor data as well as their tasking. The OGC has established a SWE working group which defines a framework of data models and encodings for describing sensors and sensor observations, as well as a suite of web service interfaces leveraging these models and encodings [4]. The Sensor Observation Service [8] is part of the SWE framework and provides a standardized interface for the pull-based access to archived and near-realtime, sensor observations and metadata. The service interface and its operations are divided into three profiles: Core, Transactional and Enhanced. The core profile includes the three mandatory operations, GetCapabilities for requesting a description of the service and the offered sensor data, DescribeSensor for retrieving sensor metadata, and GetObservation for querying observations of particular sensors or phenomena using any combination of temporal, spatial and value filters. The RegisterSensor operation of the optional transactional profile enables the registration of new sensors. Afterwards, the InsertObservation operation allows the integration of new observations produced by registered sensors. The enhanced profile offers optional operations such as the GetResult operation to retrieve only results of observations without their metadata. A service implements the entire profile, if it supports all operations.

The SOS uses the Sensor Model Language (SensorML) specification [9] for the encoding of sensor metadata descriptions. SensorML provides models and encodings to describe any kind of process in sensor or post processing systems. Thus, the basic type of all SensorML descriptions is the process type containing input and output elements, as well as several additional parameters. Different subtypes of the process type are provided for various kinds of detectors, actuators, or aggregated systems.

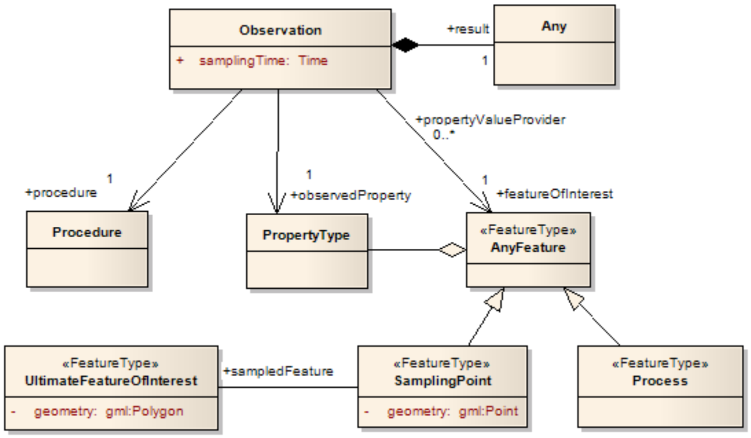


Fig. 1. Basic observation model of O&M specification

The Observations and Measurements (O&M) specification [10] is utilized by the service to encode the data gathered by sensors. It defines a model describing sensor observations as an act of observing a certain phenomenon. The basic observation model contains five components as shown in figure 1. The observation comprises a link to the procedure (usually a sensor, e.g., a water gauge), which generates the value for the observation, as well as a reference to the observed property (e.g., water level) representing the phenomenon which was observed. The feature Of interest refers to the real world entity (e.g., a river) which was target of the observation and has to carry the observed property as its feature property. The real world entity can also be a process such as the dispersion of a chemical cloud. The sampling time attribute indicates the time, when the observation was applied to the feature of interest. The observation value is contained in the result element. It depicts a symbol for the observed phenomenon during the sampling time located at a certain feature of interest. Thus, the type of the observation result must be consistent with the observed phenomenon and the observed property has to be a property of the feature of interest.

If the measurement procedure represents a sampling of a spatially distributed entity, the features of interest refer to artifacts of the sampling strategy. Therefore, part 2 of the O&M specification [11] defines certain representation types for these artifacts, the so called sampling features. The specification also provides a link from the sampling features to ultimate features of interest which represent the spatially distributed real world entities. For example, when measuring the surface temperature of a lake, the concrete locations of the measurements are represented through the sampling points. The real world entity, which carries the surface temperature property, is represented through a reference from the sampling points to the feature representation of the lake. In case of using sampling points as features of interest, the term feature of interest becomes ambiguous as it represents the artifacts of sampling as well as the real world entities which are observed. Recently, other approaches have tried to model the locations of the sampling points as part of the observation results and use the features of interest for representing the ultimate entities of interest. Consequently, in such an approach the features of interest are representations of real world entities.

Besides modeling sensors, their observations, and features of interest using SensorML and O&M, ontologies are used to specify types of sensors, observations, and features in more detail [12,7,13]. In general, ontologies are applied at three stages: modeling, integration, and discovery. First, they allow to restrict the meaning of technical terms such as *wind direction* or *pollution* towards an intended interpretation [14,15]. As executable specification, ontologies can be checked for consistency, i.e., whether they are contradiction-free, and used to make implicit knowledge explicit [16]. Second, using various reasoning services, alignment, matching, and translation [17], ontologies play a crucial role in on-the-fly integration of heterogeneous information and hence assist in establishing semantic interoperability [18]. For instance, complex service chains of Sensor Observation Services, Web Processing Services, and Web Mapping Services require more knowledge about the exchanged data than just code lists. Finally, formal definitions of sensors, observations, and feature types support information retrieval beyond simple keyword search by using reasoning services such as subsumption and similarity [19,20]. To realize these goals various research groups started to specify sensor, stimuli, and observation ontologies [21], examples include the Semantic Web for Earth and Environmental Terminology (SWEET)[1] and the sensor ontology developed as part of the W3C Semantic Sensor Network Incubator Group[2].

3 Scenario

Based on a use case of the SoKNOS project [22], this section introduces a fire scenario to illustrate the challenges for registering sensors and publishing observations. A fast extending blaze at the waste dump of Muenster in Germany causes a dispersion of pollutants into the air. The air pollutants threaten an important European bird reserve, the so called Rieselfelder, and the surrounding settlements. In our scenario, mobile sensors are deployed to monitor air pollutants, wind speed, and wind direction. We assume that a local Sensor Web is

---

[1] `http://sweet.jpl.nasa.gov/ontology/`

[2] `http://www.w3.org/2005/Incubator/ssn/wiki`

already in place and used by a disaster relief organization. The newly deployed sensors have to be made available within the Sensor Web on-the-fly. Applications can directly utilize the gathered observations to get an overview of the situation and for dispersion simulations. The scenario definition contains three examples for the registration of sensors and access to their observations:

1. A service is already set up for certain features of interest. If new sensors are registered, it has to be checked, whether these sensors produce values for already existing properties of the monitored features of interest, or whether new properties have to be created. This example illustrates the matching between outputs of a sensor and properties of already existing features of interest.
2. A service instance is already deployed for specific meteorological phenomena. If the mobile sensors are registered, it has to be checked whether the sensor outputs comply with the wind phenomena offered by the service. Additionally, when a new feature of interest is inserted into the service, it has to be assured whether the properties of the feature correspond to the phenomena provided by the service. This example demonstrates the matching of sensor outputs as well as feature properties with the phenomena offered by the service.
3. If a new observation is inserted into the service, it has to be checked whether the observed property of the feature of interest complies with the input and output of the sensor which has been registered for this observation before. This example illustrates a consistency check between the InsertObservation request and the registered sensors.

Such a scenario is typical for Sensor Web use cases as it covers two important tasks at the same time – device discovery (e.g., which sensors are necessary to monitor the gas plume) and data discovery (e.g., which data can be used to compute the dispersion of the gas plume).

4 Semantic Challenges for Sensor Registration

In the following, we analyze the challenges for registering sensors and publishing their observations on the Sensor Web. Different kinds of sensors are necessary to compute the plume of air pollutants introduced in the scenario. For the sake of readability, the following examples focus mostly on wind direction sensors.

To avoid terminological confusion[3], in this work the term entity refers to particulars in the real world. This also includes processes such as the dispersion of pollutants. The term feature of interest (or feature for short) refers to the computational representation of real world entities, e.g., a polygon representation of the gas plume. Consequently, features can also represent processes. Sensors

---

[3] The O&M specifications are not very clear about the exact meaning of the terms phenomenon, measurand, feature, and so forth, see for example [10, p.17-18], as well as the distinction between real world entities and their representation.

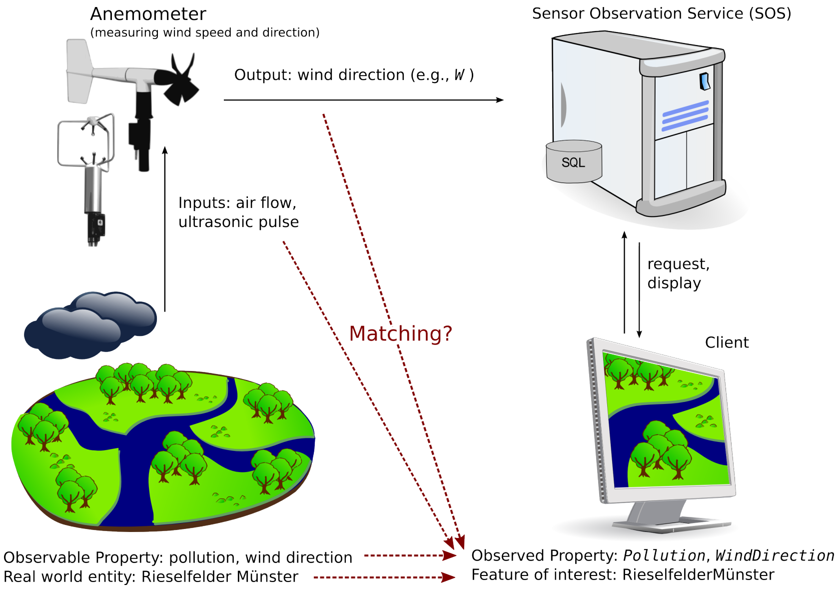


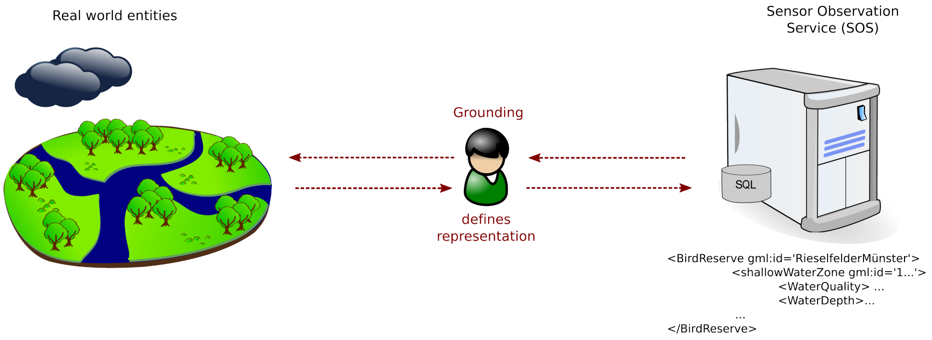
Fig. 2. Semantic Matching for Sensor Observation Services.

measure stimuli (observable phenomena) which are either directly or indirectly related to the real world entities [13,12]. The result is called an observation. For instance, one characteristic of a gas plume is the concentration of a specific pollutant. Sensors can measure this concentration and convert it into an observation value. This value then refers to the observed property of a feature of interest associated with a Sensor Observation Service. If it is not clear from the context, we will explicitly distinguish between sampling features and ultimate features.

A sensor can be added to the Sensor Web by using the RegisterSensor operation of a Sensor Observation Service. The metadata description passed along with the operation request defines input and output of the sensor. The semantic matching between inputs and outputs of sensors and the observed property of the features of interest have to be assured. Also, the real world entity observed by a sensor must match the feature of interest of an observation. The Sensor Web is missing a mechanism which ensures a meaningful matching without user interaction to support a semantically enabled sensor plug & play. In the following, three major mapping challenges are introduced and discussed in detail; an overview of the combined challenges is presented in figure 2.

4.1 Matching of Real World Entities and Features

Sensors are deployed to monitor certain entities by observing stimuli related to them [12]. Their computational representations (the features of interest) are stored in geodatabases or OGC services such as the SOS. When deploying and registering new sensors, it has to be assured that the real world entities which are observed by the sensors have their counterparts in features provided by the SOS. This challenge is depicted in figure 3.



```
<BirdReserve gml:id='RieselfelderMünster'>
            <shallowWaterZone gml:id='1...'>
                      <WaterQuality> ...
                      <WaterDepth>...
            ...
</BirdReserve>
```

Fig. 3. Matching of eeal world entities and features.

In our scenario, the sensor deployer has to define representations of real world entities (e.g., sampling points located within the Rieselfelder, or the Rieselfelder as an ultimate feature of interest) which shall be observed. After defining these features, the sensors are registered at an existing SOS instance. Since the service should only provide observations for particular features, it has to be determined whether the existing features correspond to the real world entities observed by the newly added sensors. In fact, this challenge relates to the so-called symbol grounding problem [23]. The definition of features of interest and their feature types (e.g., bird reserve) has to be grounded in a shared and commonly agreed upon reference system. Sampling points can be reduced to their spatial footprint and hence can be grounded using a spatial reference system. In contrast, the ultimate feature of interest cannot be reduced to its spatio-temporal footprint but also requires a thematic component. Therefore it has to be grounded in spatial, temporal, and semantic reference systems [24,25]. In case of the gas plume scenario, it is not clear whether the gas plume dispersion (which is a process), the waste dump, the Rieselfelder or the physical position of the wind direction sensor should be selected as feature of interest. The conceptualization of this feature also influences whether a 2D or a 3D sonic anemometer should be used as sensor (see also figure 4).

4.2 Matching of Stimuli to Sensor Inputs

The second challenge describes the matching between sensor inputs as specified in SensorML and stimuli related to real world entities. This challenge is depicted in figure 4. Sensors are used to gather information about specific characteristics of particular entities. These characteristics can only be observed by stimuli related to them [12]. Increasing temperature, for instance, can be observed by the volume expansion of mercury. Typically a single sensor is constructed to observe a single stimulus. However, a stimulus can be interpreted in different ways to learn about multiple characteristics of the observed entities. Additionally, single sensors can be combined to sensor systems. With respect to the gas plume scenario, a propeller anemometer is a combination of a wind speed and a wind direction sensor. Both sensors use the flow of air mass as stimulus. A sonic anemometer makes use of an indirect stimulus, namely the transit time of a sonic pulse between pairs of transducers to measure wind direction and speed (in 2D or 3D).

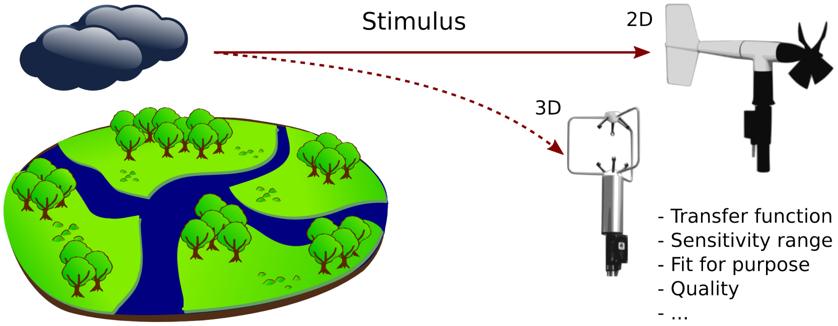


Fig. 4. Matching of stimuli to sensor inputs.

Consequently, taking the vision of a sensor plug & play with minimal human intervention seriously the feature of interest has to be modeled based on the notions of observations and stimuli. This would allow to select appropriate sensors semi-automatically and register them at a Sensor Observation Service to gather their measurements.

The stimulus to which a sensor reacts is the origin of its measurement procedure. Besides this basic characteristic other more technical properties of the sensor such as the transfer function, the sensitivity range and quality parameters describe the behavior of a sensor. All these different properties have to be considered to prove the suitability of a sensor to measure certain characteristics.

4.3 Matching of Sensor Output and Feature Property

As outlined in section 2, an observation acts as a property value provider for a feature of interest. For example, an observation provides a value (e.g., 20°) generated by a sensor (e.g., an anemometer) for certain characteristics (e.g., wind direction) of a feature (e.g., the Rieselfelder) at a certain time-stamp.

The challenge in this case is whether the symbol and the semantics of the output produced by the sensor comply with the symbol and semantics of the property of the feature of interest. So far, code lists are used for a syntactic matching. Consequently, it is up to the SOS provider to ensure that the semantics of *WindDirection* in a particular SOS matches to *prevailing_direction* in

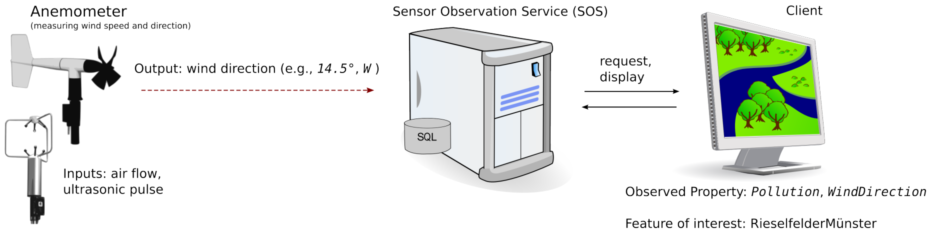


Fig. 5. Matching of sensor output and feature property.

a SensorML description. The example of wind direction shows that answering this question can be challenging. Wind direction can be defined as the direction *from* which the wind blows, or as the direction the wind is blowing *to*. The observation value for wind direction can be an angle, a textual value, or a more complex XSD type [26].

5 Towards Semantically enabled Sensor Plug & Play

While the previous section introduced several challenges for registering sensors and retrieving their observations, this section outlines the role of semantic annotation and reasoning to realize the envisioned sensor plug & play. In general, plug & play aims at reducing or avoiding any manual configuration when plugging new components into a system. With respect to Sensor Observation Services, it should be possible to select and register sensors with minimal human interaction. In large scale real world applications, it is unlikely that the provider of a Sensor Observation Service is also responsible for modeling the various features of interest, observations, and sensors using O&M and SensorML, respectively. In fact, these components are provided by external sources. Features of interest can be retrieved from Web Feature Services using semantically enabled catalogues [5,27]. The definitions of observable characteristics can be taken from ontologies such as SWEET or extracted from statistical models, while the SensorML annotations can be provided by the sensor manufacturers. In this case, the SOS provider cannot simply assume a meaningful correspondence based on the name of a sensor output and the name of a property related to a feature of interest. Semantic matching needs to assure that both names point to the same domain concept, e.g., the shared conceptualization that wind direction is denoted as pointing to the compulsion, in degrees, and in compass direction; see also [26] for details.

In the following, we assume that the sensors used in the gas plume scenario are accompanied by a SensorML self-description provided by its vendor or manufacturer. Consequently, an SOS provider does not have to create the SensorML description at runtime. Additionally, we assume that the Sensor Web infrastructure contains an SOS which offers sensor data for multiple sampling features located within the affected ultimate feature, i.e., the Rieselfelder, and it is set up for various observable properties such as wind speed, wind direction, and pollutant concentrations. A new sensor can be made available on the Sensor Web by adding it to the SOS. Therefore, the RegisterSensor operation is invoked whose request contains the SensorML description. Listing 1.1 shows a fragment of such a request which registers an anemometer. The sensor is modeled as a system which incorporates among other descriptive elements an input and an output.

```
<RegisterSensor service='SOS' version='1.0.0'>
  ...
  <sml:System>
    ...
    <sml:inputs>
      <sml:InputList>
        <sml:input name='air_movement'>
          <swe:ObservableProperty definition=
            'urn:ogc:def:phenomenon:OGC:air_movement'/>
        </sml:input>
      </sml:InputList>
    </sml:inputs>
    <sml:outputs>
      <sml:OutputList>
        <sml:output name='wind_direction'>
          <swe:Quantity definition=
            'urn:ogc:def:phenomenon:OGC:wind_direction'>
            <swe:uom code='deg' />
            <swe:quality>
              <swe:QuantityRange definition=
                'urn:ogc:def:phenomenon:OGC:tolerance'>
                <swe:value>−0.5 0.5</value>
              </swe:QuantityRange>
            </swe:quality>
          </swe:Quantity>
        </sml:output>
  ...
</RegisterSensor>
```

Listing 1.1. Request to register a new sensor.

The stimulus observed by the sensor, its input, is identified by the definition attribute whose value is a Unified Resource Name (URN)[4]. It uniquely identifies the referenced concept, in this case air movement, by pointing to a description stored in a dictionary or code list. Similarly, the sensor output, the wind direction, is referenced by a URN. Sensor observations provide values for properties of particular features of interest, e.g., a gas plume, associated with the SOS. To enable sensor plug & play, we propose to refer to ontologies containing formal specifications for stimuli, observations, and functional aspects of a sensor (marked bold in the listing). One example for such an ontology is the sensor type ontology developed at the W3C Semantic Sensor Network Incubator Group. While this ontology provides definitions for sensors and their components, future ontologies have to define stimuli and observations [13]. Existing technologies for ontological alignment, matching [17], and similarity [20] can then be used to ensure that the specified output of a sensor produces appropriate values for properties of certain features of interest. For instance, if a feature

---

[4] The structure scheme for the OGC namespace is defined by Whiteside [28].

property wind direction has been modeled as a 3-dimensional quality, trying to assign a 2D anemometer to it would produce an error or warning (see also [29] for an ontological investigation on the dimensionality of qualities). Whether the semantic annotation of SensorML documents is realized using RDFa [6,7] or other technologies such as SAWSDL [30,31], is an implementation decision not discussed here[5]. A similar approach was introduced by Hornsby and King [32] for the transportation domain.

For our scenario we assume that the sensor invokes the InsertObservation operation of the SOS as soon as data is available. Listing 1.2 shows a fragment of such a request.

```
<InsertObservation service='SOS' version='1.0.0'>
  ...
  <om:Observation>
    ...
    <om:procedure xlink:href=
      'urn:ogc:object:feature:Sensor:IFGI:s01'/>
    <om:observedProperty xlink:href=
      'urn:ogc:def:phenomenon:OGC:wind_direction'/>
    <om:featureOfInterest>
      <sa:SamplingPoint gml:id='sampling01'>
        <sa:sampledFeature xlink:href=
          'urn:ogc:def:feature:OGC:Rieselfelder'/>
        <sa:position>
          <gml:Point>
            <gml:pos srsName= 'urn:ogc:def:crs:EPSG:4326'>
              7.89 52.90
            ...
      </sa:SamplingPoint>
    </om:featureOfInterest>
    <om:result uom='deg'>52.0</om:result>
...
```

Listing 1.2. Request to insert new observations.

Similar to the RegisterSensor operation, the InsertObservation request has to be semantically annotated. This way, the SOS can verify whether the output type defined by the sensor is semantically compliant with one of the observed properties associated with the SOS and corresponding to a property of an associated feature of interest, e.g., a sampling feature within the Rieselfelder.

On the long term and based on fixed stimulus-observation alignments, one could automatically discover and select appropriate sensors using features of interest and observations as queries. For instance, in case of a query such as *will the Rieselfelder be affected by a gas plume*, appropriate sensors and processing services can be automatically selected and arranged (using a Sensor Planning Service [33]). The necessary inference can be performed based on the knowledge about types of features and observations provided in the ontologies. Gas plumes, for instance, can be modeled as processes which have a direction of dispersion,

---

[5] An API for the semantic annotation of OGC services is under development and can be downloaded at `https://www.assembla.com/spaces/dashboard/index/sapience`.

concentrations of different pollutants and so forth. Each of these characteristics can be aligned to stimuli in a stimulus ontology used to describe sensors.

This, however, requires the integration of ontology repositories and reasoning services into spatial data infrastructures. First, a Web Reasoning Service (WRS) is needed to encapsulate the reasoning components developed as core parts of the Semantic Web. Such a service could be developed as a profile of the Web Processing Service specification [34]. Using a WRS, users could query a Web Feature Service for waterbodies and retrieve individual rivers, lakes, reservoirs, and so forth. Based on similarity reasoning, users can also query for specific features (e.g., canals) and get similar features back in addition (e.g., rivers). Second, a Web Ontology Service (WOS) has to be designed which acts as a catalogue to registered ontologies and enable semantics-based discovery of Sensor Web related concepts such as features, observations, and sensors. A WOS can be considered as a profile for the OGC Catalogue Service [35]. Introducing profiles instead of completely new service types enables the integration with existing SDI technologies and simplifies the service orchestration; see [5] for details.

6 Conclusions and Further Work

In this paper, we discussed the challenges related to registering new sensors and inserting their observations to a Sensor Observation Service. We argued that these matchings have to be established and maintained manually by the provider of the SOS and explained the difficulties in doing so. Starting with an abstract view on the semantic matching challenges, we described how the semantic annotation of RegisterSensor and InsertObservation requests can serve as a basis for reasoning-based consistency checking and hence improve the manual matching process. The long term vision underlying this research is to enable sensor plug & play with minimum human intervention.

The main difficulty lies in the relationship between the different OGC constructs used to model sensors, observations, and features of interest on the Sensor Web. Three challenges can be distinguished. The first describes the relation of the real world entity and the corresponding feature of interest as the computational artifact. If two sensors of different type both deliver observations assigned to a particular feature of interest in an SOS, do they both refer to the same real world entity? This challenge relates to the symbol grounding problem and requires further work on reference systems [25]. The second challenge is related to the selection of an appropriate sensor which is capable of measuring characteristics of a particular feature - the sensor inputs, i.e., real world stimuli, and the entity's observed characteristics have to match. Third, the sensor output has to comply with the property of the feature of interest stored in a Sensor Observation Service. Using the wind direction as an example, we discussed why a purely syntactic matching is not sufficient.

While we focused on introducing these challenges as well as the role of semantic annotation and reasoning, the implementation of this work is part of the 52°*North* semantics community which aims at establishing a semantic-enablement layer for OGC services[6]. Besides introducing the idea of sensor plug & play, the paper also shows that two new OGC service types are being required to incorporate semantics-based information retrieval, on-the-fly integration, and the composition of sensors and services. However, the question how to represent perdurants, for example the dispersion of the gas plume, in services such as the WFS is still an open issue.

Acknowledgment

Major parts of the presented research have been developed at a joint workshop of the Institute for Geoinformatics, University of Münster, Germany, and the National Institute for Space Research (INPE), Brazil, which has been supported by the German Research Foundation (DFG), project no. 444 BRA 121/2/09 and by The State of São Paulo Research (FAPESP), project no. 2008/11604-6. Partial funding came from the International Research Training Group on *Semantic Integration of Geospatial Information* (DFG GRK 1498).

References

1. Connaghan, D., Hughe, S., May, G., O'Brien, K., Kelly, P., Connaire, C.O., O'Connor, N., O'Gorman, D., Warrington, G., Smeaton, A.F., Moyna., N.: A Sensing Platform for Physiological and Contextual Feedback to Tennis Athletes. In: BSN 2009 - Body Sensor Networks Workshop 2009. (2009)
2. Hayes, J., O'Conor, E., Cleary, J., Kolar, H., McCarthy, R., Tynan, R., O'Hare, R., Smeaton, A., O'Connor, N., Diamond, D.: Views From the Coalface: Chemo-Sensors, Sensor Networks and the Semantic Sensor Web. In: SemSensWeb 2009 - International Workshop on the Semantic Sensor Web. (2009)
3. Shepherd, D., Kumar, S.: Microsensor Applications. In: Distributed Sensor Networks. Chapman & Hall (2005)
4. Botts, M., Percivall, G., Reed, C., Davidson, J.: OGC Sensor Web Enablement: Overview And High Level Architecture. Technical report, Open Geospatial Consortium (2007)
5. Janowicz, K., Schade, S., Bröring, A., Keßler, C., Stasch, C., Maué, P., Diekhof, T.: A transparent semantic enablement layer for the geospatial web. In: Terra Cognita 2009 Workshop In conjunction with the 8th International Semantic Web Conference (ISWC 2009). (2009; forthcomming)
6. Sheth, A., Henson, C., Sahoo, S.: Semantic Sensor Web. IEEE Internet Computing (2008) 78–83
7. 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)
8. Na, A., Priest, M.: OGC Implementation Specification 06-009r6: OpenGIS Sensor Observation Service (SOS) (2007)
9. Botts, M.: OGC Implementation Specification 07-000: OpenGIS Sensor Model Language (SensorML) (2007)

---

[6] `http://www.52north.org/semantics`

10. Cox, S.: OGC Implementation Specification 07-022r1: Observations and Measurements - Part 1 - Observation schema (2007)
11. Cox, S.: OGC Implementation Specification 07-022r3: Observations and Measurements - Part 2 - Sampling Features. Technical report, Open Geospatial Consortium (2007)
12. Stasch, C., Janowicz, K., Broering, A., Reis, I., Kuhn, W.: A Stimulus-Centric Algebraic Approach to Sensors and Observations. In: 3rd International Conference on Geosensor Networks. Volume 5659 of Lecture Notes in Computer Science., Springer (2009) 169–179
13. Kuhn, W.: A functional ontology of observation and measurement. (2009; under review) [The ontology is also available online at http://musil.uni-muenster.de/wp-content/uploads/Observation2.hs].
14. Guarino, N.: Formal Ontology and Information Systems. In Guarino, N., ed.: International Conference on Formal Ontology in Information Systems (FOIS1998). IOS Press, Trento, Italy (1998) 3–15
15. Kuhn, W.: Semantic Engineering. In Navratil, G., ed.: Research Trends in Geographic Information Science, Springer (2009; forthcoming)
16. Allemang, D., Hendler, J.: Semantic Web for the Working Ontologist: Modeling in RDF, RDFS and OWL. Morgan Kaufmann Elsevier, Amsterdam, NL (2008)
17. Shvaiko, P., Euzenat, J.: Ten Challenges for Ontology Matching. In Meersman, R., Tari, Z., eds.: On the Move to Meaningful Internet Systems: OTM 2008. Volume 5332 of Lecture Notes in Computer Science., Springer (2008) 1164–1182
18. 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
19. 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, Mexico City, Mexico, Springer (November 2007) 128–145
20. Janowicz, K., Wilkes, M.: SIM-DLA: A Novel Semantic Similarity Measure for Description Logics Reducing Inter-concept to Inter-instance Similarity. In Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvoenen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E.P.B., eds.: 6th Annual European Semantic Web Conference (ESWC2009). Volume 5554 of Lecture Notes in Computer Science., Springer (2009) 353–367
21. Compton, M., Henson, C., Lefort, L., Neuhaus, H.: A survey of the semantic specification of sensors. Technical report (2009) [Available online at http://lists.w3.org/Archives/Public/public-xg-ssn/2009Aug/att-0037/SSN-XG_StateOfArt.pdf].
22. Stasch, C., Walkowski, A.C., Jirka, S.: A Geosensor Network Architecture for Disaster Management based on Open Standards. In Ehlers, M., Behncke, K., Gerstengabe, F.W., Hillen, F., Koppers, L., Stroink, L., Wächter, J., eds.: Digital Earth Summit on Geoinformatics 2008: Tools for Climate Change Research. (2008) 54–59
23. Harnad, S.: The Symbol Grounding Problem. Physica D 42 (1990) 335–346
24. Kuhn, W.: Semantic Reference Systems. International Journal of Geographic Information Science **17**(5) (2003) 405–409
25. Scheider, S., Janowicz, K., Kuhn, W.: Grounding Geographic Categories in the Meaningful Environment. In: Conference on Spatial Information Theory (COSIT 2009). Number Lecture Notes in Computer Science, Springer (2009, forthcoming)

26. 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)
27. Stock, K., Small, M., Ou, Y., Reitsma, F.: OGC Discussion Paper 09-010 - OWL Application Profile of CSW. Technical report, Open Geospatial Consortium (2009)
28. Whiteside, A.: OGC Recommendation Paper 05-010: URNs of Definitions in OGC Namespace (2007)
29. Probst, F., Espeter, M.: Spatial dimensionality as a classification criterion for qualities. In Bennett, B., Fellbaum, C., eds.: International Conference on Formal Ontology in Information Systems (FOIS 2006). Volume 150 of Frontiers in Artificial Intelligence and Applications., IOS Press (2006) 77–88
30. Farrell, J., Lausen, H.: Semantic annotations for WSDL and XML schema. W3C recommendation, available from `http://www.w3.org/TR/sawsdl/` (2007)
31. Maué, P., Schade, S., Duchesne, P.: OGC Discussion Paper 08-167r1: Semantic annotations in OGC standards. Technical report, OGC (2009)
32. Hornsby, K., King, K.: Linking geosensor network data and ontologies to support transportation modeling. In Nittel, S., Labrinidis, A., Stefanidis, A., eds.: GeoSensor Networks: Second International Conference, GSN 2006, Springer (2008) 191–209
33. Simonis, I.: OGC Implementation Specification 07-014r3: OpenGIS Sensor Planning Service. Technical report, Open Geospatial Consortium (2007)
34. Schut, P.: OGC Implementation Specification 05-007r7: OpenGIS Web Processing Service (2007)
35. Nebert, D., Whiteside, A., Vretanos, P.: OGC Implementation Specification 07-006r1: OpenGIS Catalogue Services Specification (2007)