Environmental Observations in Knowledge Graphs

Rui Zhu[1] ✉, Shirly Stephen Ambrose[1], Lu Zhou[2], Cogan Shimizu[2], Ling Cai[1], Gengchen Mai[1], Krzysztof Janowicz[1], Pascal Hitzler[2], and Mark Schildhauer[3]

[1] Center for Spatial Studies, University of California, Santa Barbara, USA
{ruizhu, shirlystephen, lingcai, gengchen_mai, jano}@ucsb.edu
[2] Data Semantics Laboratory, Kansas State University, USA
{luzhou, coganmshimizu, hitzler}@ksu.edu
[3] National Center for Ecological Analysis & Synthesis, University of California, Santa Barbara, USA
schild@nceas.ucsb.edu

Abstract. The notion of Linked Open Science rests on the assumption that Linked Data principles contribute to science and scientific data management in several distinct ways (e.g., by adding rich semantics to improve retrieval and reuse of data). This begs the question of the right level of granularity for such semantic enrichment. On the one extreme of the spectrum, one may provide semantic annotations on the level of entire datasets to improve retrieval while leaving the actual data untouched. On the other end, one may semantically describe every single datum, such as a particular observation leading to data that supports reasoning, automated conflation, and so on, while, at the same time, dramatically increasing the size of data, including redundancy. This paper reports on our experience in modeling heterogeneous environmental data using a semantically-enabled observation framework, namely the SOSA ontology and its extensions to handle observation collections. We discuss different means of using these observation collections and compare their pros and cons in terms of data size and ease of querying.

Keywords: Environmental data · Observation · Observation collection.

1 Introduction

The exponential growth of multi-modal and highly heterogeneous data collected from diverse sensors has brought about a new means to inspect, understand, and interact with the environment. However, most of these data are collected and stored separately as different data *silos* that are difficult to share and integrate. How to timely synthesize and consequently allow researchers to reuse environmental data, therefore, becomes a challenge. Fortunately, rapid development of Linked Data techniques that utilize robust and well-known W3C standards (e.g., RDF[2], OWL [5], and SHACL [8]) enables us to represent, interlink, reason, as well as validate information, including environmental information, on the Web.

Nevertheless, environmental data are unique on the Web. First, environmental observations come from a diverse range of data formats, including raster-based remote sensing imagery, point-based monitoring station observations, polygon-based disaster impact estimates, place-based demographic census, and so on. Secondly, most environmental data are collected using (physical) sensors or based on forecast models and is scoped by both space and time. Even though this is also true for other types of data (e.g., business and laws) spatial and temporal data about sensors and their observations require a lot of local contextual information for identifying and understanding these observations, without which all downstream scientific analyses turns to be meaningless. Finally, individual environmental observations often share common characteristics. For example, two observations can both be about the same property; they can also both be located at the same location, can happen at the same time, or be part of a series taken by the same sensor.

With its uniqueness in mind, we explore the use of the SSN and SOSA ontologies [3,6] and their extension[4] to model environmental data for scientific research and beyond. These W3C and OGC standards allow us to address the points above, respectively. First, the SOSA ontology focuses on semantically describing sensors and their observations, along with a supporting framework that contextualizes these observations. Secondly, this contextualization contains explicit definitions that allow a modeler to directly capture the semantic aspects of the spatial and temporal scope of environmental data. Finally, SOSA ontology's extension of the observation collection concept can reduce the redundancy that is naturally embedded in environmental observations by aggregating homogeneous characteristics into collections.

In this paper, we discuss our experiences in modeling environmental information using the SOSA/SSN ontologies, as well as their extension. We specifically focus on proposing different types of *aggregators*, components that are jointly used in a data to enable the grouping of observations into a collection. Compared with the traditional way of modeling environmental data as pieces of *individual* observations, the *collective* approach aggregates observations that share one or more characteristics so as to reduce the redundancy of the resulted knowledge graph. For instance, several air quality sensors each providing their reading at exactly the same time may be aggregated along the temporal dimension, allowing us to reduce the number of materialized triples in a knowledge graph. However, if we apply too complicated (e.g., nested) aggregation on top of the data, we might create an overly long property path for querying. Hence, we further investigate the trade-off between graph size and query complexity while applying observation collections. We believe this work contributes to the overarching effort of making environmental data and underlying ontologies FAIR [10] and 5-Star Quality [4,7] for scientific research.

In the next section, a brief discussion about background and related work is discussed. Following that, we introduce different types of aggregators for building

[4] Namely the ObservationCollection, from [1]; additional detail is provided in Section 2.

observation collections, alongside examples. In Section 4, we briefly explore the trade-offs between using individual observations and observation collections, in terms of the graph size and query complexity. We conclude and provide next steps in Section 5.

2 Background & Related Work

The Semantic Sensor Network (SSN) ontology and its core, the Sensor, Observation, Sample, and Actuator (SOSA) ontology are W3C recommendations[5] for representing sensors and their observations in a semantically rich way. Atomic classes in SOSA include Sensor, Observation, FeatureOfInterest, Observable Property, Result, and so on, and their interrelations are depicted in Fig. 1 (note that only the relevant classes and properties are listed here; readers are encouraged to check [6] for more details). The semantics underlying this schema can be simply interpreted as: a Sensor makes Observation(s), about the ObservableProperty of some FeatureOfInterest to yield a certain Result at time (xsd:dateTime). When the result is as simple as a rdfs:literal (e.g., numeric value such as "10"), the object property of hasResult can be replaced by a data type property hasSimpleResult. To facilitate the modeling of *homogeneous* collection of observations, such as those that share the same feature of interest, observable property, result time and so on, Cox introduced a SOSA extension[6] by adding a new class of *ObservationCollection* which can have instances of *Observation* as members (Fig. 1 right). Although this work focuses on using the SOSA ontology and its extension, there

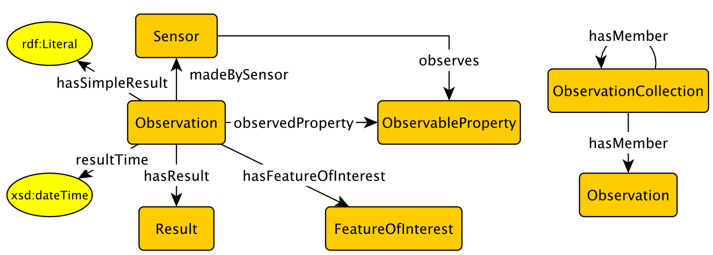


Fig. 1: A top-level, simplified schema diagram for the SOSA ontology (left) and its extension on observation collection (right).

are other ontological specifications used to model scientific observations, such as the RDF Data Cube vocabulary (DataCube)[7] and the Extensible Observation Ontology (OBOE)[8]. DataCube is designed to publish multidimensional statisti-

[5] https://www.w3.org/standards/
[6] https://www.w3.org/TR/2020/WD-vocab-ssn-ext-20200116/
[7] https://www.w3.org/TR/vocab-data-cube/#full-example
[8] https://bioportal.bioontology.org/ontologies/OBOE/?p=summary

cal data on the Web, which centers on three concepts (called cube): Dimension, Measure, and Attribute. In contrast to SOSA, observations in DataCube has a primary property defined as the Measure and all the others are Dimension(s), which are fixed for the same type of observations. Interestingly, DataCube defines a concept of Slice, which is similar to SOSA extension's ObservationCollection construct. Specifically focusing on ecology, OBOE's core concepts include Observation, Entity, Measurement, Characteristic, and MeasurementStandard to semantically record complex scientific observations and measurements [9]. By design, OBOE is capable of modeling homogeneous collections of observations because Observation in OBOE corresponds to ObservationCollection in SOSA and Measurement of OBOE is equivalent to Observation of SOSA[9]. However, in contrast to SOSA extension, OBOE is less flexible as its Observation (the collection construct) is only linked to Entity (corresponds to FeatureOfInterest in SOSA) without allowing other Characteristic(s), that might be homogeneous as well, to be the aggregator of the collection. A comprehensive comparison of the usage of these three ontologies in modeling environmental observations will be studied in the future.

Environmental data varies greatly by their theme, publishing agency, data structure, data format, spatial and temporal scope, and so on. To ensure our work covers the spectrum of such a diverse realm as widely as possible, we collect data from 13 data sources related to topics such as natural hazard, soil health, drought, wildfire, cropland, air quality, and so on as part of our work on creating a global knowledge graph of data at the interface between humans and their environment. In the next section, they are used to exemplify the category of aggregators in building observation collections using SOSA and its ObservationCollection extension.

3 Useful Aggregators for ObservationCollection

Aggregators are those common components that are shared by a group of observations. In SOSA, aggregators can be any instances of the class FeatureOfInterst, ObservableProperty, Sensor, as well as the data type property resultTime (or the object property phenomenonTime). If a set of observations share one or more of these aggregators, they can be composed as an observation collection. Furthermore, aggregators can be used at different hierarchical (nested) levels. Namely, an observation collection made by one aggregator can be the *member* of a higher-level observation collection that is created by another aggregator. The decision about whether to use observation collection or not, and if so what aggregator(s) should be selected depend on the nature of the environmental datasets, user requirement (i.e., competency questions), as well as ontology designer's experience. In this section, we summarize our experience of using different types of aggregators to model environmental information in knowledge graphs.

[9] `https://www.w3.org/TR/vocab-ssn/#OBOE_Alignment`

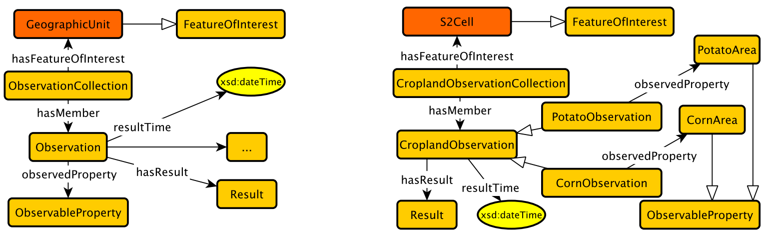
3.1 Geographic Units and Regions

Thanks to their fundamental role in environmental data, geographic units are often considered as the common factor to aggregate environmental data into collections. Geographic units and regions (e.g., geometric points, polylines, polygons, as well as places in general) can be defined by different institutes/government organizations (e.g., climate divisions, soil map units, and administrative regions) or event-induced (e.g., wildfire scars). For instance, Natural Resources Conservation Service (NRCS) used Soil Map Units as the basic geographic unit, which delineates extents of different types of soils, based on which physical, chemical, nutrient and moisture properties about soil are collected. On basis of geographic units, usually two kinds of data are collected – identification data and observable properties. The former usually contains a unique global identifier (i.e., index), name, etc., which provides the basic identifying information about each geographic unit, while the latter describes properties of geographic units observed or measured by sensors. The identifying information can be naturally modeled by individual RDF triples, while Observation and ObservationCollection in SOSA ontology are used to model observed properties of different types. For instance, when it comes to only a single observable property for each geographic unit, e.g., smoke plume density provided by NOAA[10], we model each smoke plume (often represented as a geometric polygon) as a FeatureOfInterest and the single observation of each smoke plume – the smoke density as an Observation, each of which is associated with an ObservableProperty (i.e., SmokeDensity) and its Result (i.e., measured density value). For cases when multiple observable properties are available for a single geographic unit (e.g., FeatureOfInterest), ObservationCollection can be applied (see the pattern in Fig. 2a). For instance, to represent cropland types across the US[11], whose raw data is originally represented as raster images, we can use ObservationCollection to model the areal distributions of different types of cropland (i.e., ObervableProperty) within a specific geographic unit, such as a discrete global grid cell on the surface of the Earth like S2Cell[12]. See Fig. 2b for a detailed illustration.

3.2 Event (Space and Time)

An event always consists of spatial (represented using geographic units) and temporal aspects. For example, an earthquake event is always linked to a seismic activity, more specifically, the coordinate, seismic type, and magnitude of the earthquake experienced over a period of time. The seismic activity is usually monitored and observed by one or more sensor networks. In order to understand how exactly the observation and observation collection in the SOSA ontology can be used to model an earthquake event, we observed the real-time earthquake

[10] `https://www.ospo.noaa.gov/Products/land/hms.html`
[11] `https://www.nass.usda.gov/Research_and_Science/Cropland/Release/index.php`
[12] `https://s2geometry.io/devguide/s2cell_hierarchy.html`



(a) Using geographic unit as the aggregator in SOSA extension.

(b) Example of using geographic unit as the aggregator.

Fig. 2: Pattern and example of using geographic unit as the aggregator. Orange boxes are the class/subclass of GeographicUnit

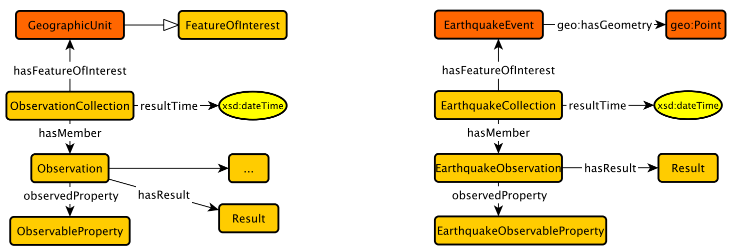
data from U.S. Geological Survey (USGS). The ANSS Comprehensive Earthquake Catalog (ComCat) contains earthquake source parameters (e.g., magnitude, depth, and coordinates) produced by contributing seismic networks[13]. Based on the nature of the dataset and SOSA ontology, we instantiated and modified the Observation and ObservationCollection pattern to describe the earthquake event. In Fig. 3b, class EarthquakeEvent is a subclass of FeatureOfInterest in SOSA ontology and a subclass of geo:Feature in GeoSPARQL ontology[14]. Each row in the USGS earthquake dataset (e.g., CSV file) can be considered as one EarthquakeObservation, which is a subclass of Observation in the SOSA ontology. Each column or each parameter in the Earthquake Catalog is modeled as one type of EarthquakeObservableProperty, and links to the EarthquakeObservation through the property observedProperty. For example, one earthquake observation might have magnitude and depth as its observable properties. And the value of each observation is stored as an instance of class Result. Most earthquakes form part of a sequence and are related to each other in terms of location and time. Therefore, earthquake observations are usually made as part of a set or collection, within which variations of the result are of interest. For efficient discovery and data transfer, ObservationCollection in Cox's SOSA extension is typically designed for this purpose and is also applied in our earthquake event schema. An earthquake observation collection consists of multiple observations that can be sorted by different networks, location, or time. This modeling strategy provides a more complex structure to enable easier enrichment of the ontology in different levels of granularity. The pattern to use events as integrators to model observation collections is depicted in Fig. 3a.

3.3 Observable Property

For some other types of environmental observation, sensors collect observations related to one kind of observable property and those observations are collected

[13] https://earthquake.usgs.gov/data/comcat/data-eventterms.php

[14] https://www.ogc.org/standards/geosparql



(a) Using event as the aggregator in SOSA extension.

(b) Example of using event as the aggregator.

Fig. 3: Pattern and example of using event as the aggregator. Orange boxes are the class/subclass of either the spatial or temporal component in the event.

periodically across a time span (e.g., time-series data). A pattern for such cases is illustrated in Fig. 4a. One example is the EPA's (US Environmental Protection Agency) Air Data[15]. This dataset provides daily air quality data collected at outdoor monitors across the US for seven kinds of air pollutants (ObervableProperty).

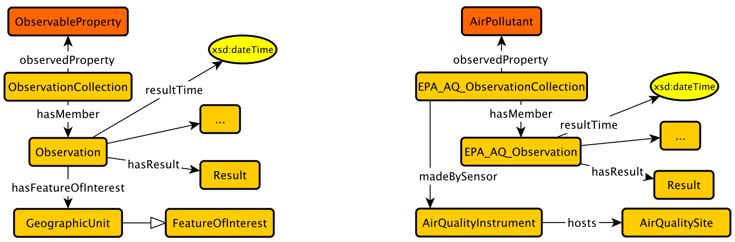
Each air quality site/instrument is modeled as a Platform, while the individual sensors on an instrument is a type of Sensor. Each instrument generates two air quality measurements: pollutant concentration and air quality index specific to an air pollutant (modeled as ObservableProperty) on a daily basis. Observations are aggregated based on the characteristics of common observable properties, while individual observations are temporally scoped (see Fig. 4b). The temporal resolution of each observation is the date in which the measurement was recorded. Finally, it is also worth noting that the feature of interest becomes implicit in this dataset as the observation (e.g., PM2.5 observed at a monitoring station) is directly associated with the sensor, which happens to be the feature of interest as well. Researchers often utilize the monitored observations to interpolate/predict the value of the same observable property at other unobserved locations (or features of interest)[16].

3.4 Other Types of Aggregators

In addition to the aforementioned three types of aggregators, there are other options to group observations into collections. One is to use temporal information (i.e., resultTime or phenomenonTime). Similar to using geographic units in Section 3.1, many environmental observations can also be aggregated through time, as shown in Fig. 5. However, such an approach is usually not recommended as it often leads to a graph of large size. In contrast to geographic units, whose number is typically finite (e.g., there are only 50 states in the US), the number

[15] `https://www.epa.gov/outdoor-air-quality-data`

[16] More strictly speaking the feature of interest is the body of air surrounding the sensor (which has no unique IRI) nor is of any interest aside of sampling.



(a) Using observable property as the aggregator in SOSA extension.

(b) Example of using observable property as the aggregator.

Fig. 4: Pattern and example of using observable property as the aggregator. Orange boxes are the class/subclass of ObservableProperty.

of temporal units is often much greater for most environmental data. For example, for the EPA data described in Section 3.3, the pollutant observations are recorded daily. So there will be about 365 time stamps for a single year, and EPA data goes back to the past 42 years. Using time as the aggregator would lead to about $365 \times 42 = 15330$ observation collections while using geographic unit instead, it will only have 50 observation collections. Other data may be collected even more frequently.

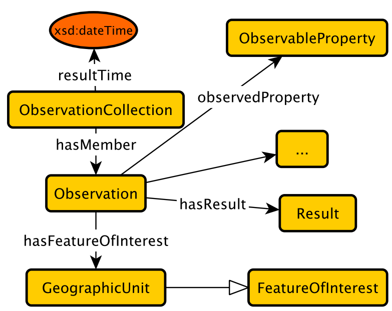
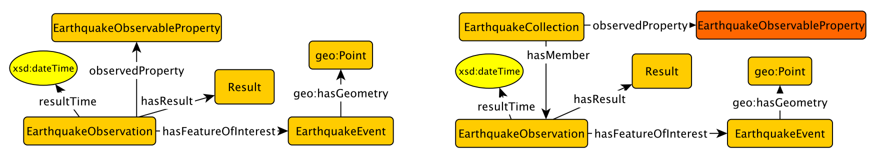


Fig. 5: Using time as the aggregator in SOSA extension

4 Individual vs Collection

While different aggregating patterns can be adopted for modeling the same dataset, they affect the number of generated triples, and the complexity of querying. To illustrate this, we use the earthquake event dataset from USGS as an example. This dataset contains in total 74,317 earthquake events that are over magnitude of 4.5 from 2011 to 2021. Each event has an associated time stamp and a geographic unit (i.e., a pair of geographic coordinates). Plus, 16 observable properties, such as depth, location source, and magnitude are recorded. Section 3.2 introduces the way of using event (time and geographic unit) as the aggregator to build the observation collection. Fig. 6 shows two alternatives: one without using the collection construct, and the other using observable property as the aggregator.



(a) Only using observation to model earthquakes.

(b) Using observable property as the aggregator to model earthquakes.

Fig. 6: Two alternatives to model earthquake observations.

Table 1 lists example triples to represent a sample observation using the three different ways, based on which the total numbers of triples for the whole dataset are estimated. When no aggregator is used (i.e., simply using observation), 5 triples are needed to represent an observation about one observable property (e.g., magnitude). Since there are 16 observable properties, we need $5 \times 16 = 80$ triples to represent one earthquake event. By multiplying it with the total number of earthquake events (i.e., $74,317$), we need $5,945,360$ triples in total. In contrast, when the aggregator of event is applied to construct the observation collection, there are in total $4,979,239$ triples, where each event uses 67 triples: 3 for triples of predicate: rdf:type, hasFeatureOfInterest and resultTime; 16 for the predicate hasMember, with each corresponds to an observation of one observable property; $16 \times 3 = 48$ for the predicate of rdf:type, observedProperty, and hasSimpleResult of individual observations of the 16 observable properties. This comparison clearly illustrates the benefit of using ObservationCollection in terms of reducing the number of redundant triples (about 1 million triples can be saved for such an earthquake dataset). With respect to query complexity (here we approximate it using the size of the graph pattern in the WHERE clause of a SPARQL query), when using observation collection, there will be two more triples added to the query of getting the magnitude of an earthquake happened at a time and a location. Moreover, different aggregators might have different performances in terms of reducing the number of triples. For instance, when using observable property as the aggregator for such an earthquake dataset, the total number of required triples will be about $5,945,392$, which is close to the approach of not using any aggregators. Specifically, there are 16 observation collections now, each of which corresponds to one observable property. Then it needs $2+74,317$ triples (predicates of rdf:type, observedProperty, and $74,317$ has-

Member) to represent one single observation collection. Furthermore, there are in total $16 \times 74,317$ observations, and each needs further 4 triples to represent it (predicate of rdf:type, hasFeatureOfInterest, resultTime, and hasSimpleResult). Full equations of computing the total number of triples can be seen in Table 1.

Aggregator	Example triples	Total number of triples	Query to get the magnitude of an earthquake happened at time X and location Y
NA	:eqMagObs1 a sosa:Observation ; sosa:hasFeatureOfInterest :eqEvent1 ; sosa:observedProperty :eqMagnitude ; sosa:hasSimpleResult ”4.8” ; sosa:resultTime ”2017-02-01” .	(5×16) $\times 74,317$ $=$ 5,945,360	SELECT ?result WHERE { ?eqMagObs a sosa:Observation ; sosa:hasFeatureOfInterest ?eqEvent ; sosa:observedProperty :eqMagnitude ; sosa:hasSimpleResult ?result ; sosa:resultTime X . ?eqEvent geo:hasGeometry Y . }
Event	:eqObsCol a sosa:ObservationCollection ; sosa:hasFeatureOfInterest :eqEvent1 ; sosa:resultTime “2017-02-01” ; sosa:hasMember :eqMagObs1, [...] . :eqMagObs1 a sosa:Observation ; sosa:observedProperty :eqMagnitude ; sosa:hasSimpleResult “4.8” .	$(3 + 16 +$ $16 \times 3)$ $\times 74,317$ $=$ 4,979,239	SELECT ?result WHERE { ?eqMagObsCol a sosa:ObservationCollection ; sosa:hasFeatureOfInterest ?eqEvent ; sosa:resultTime X ; sosa:hasMember ?eqMagObs . ?eqMagObs a sosa:Observation ; sosa:observedProperty :eqMagnitude ; sosa:hasSimpleResult ?result . ?eqEvent geo:hasGeometry Y .}
Observable Property	:eqObsCol a sosa:ObservationCollection ; sosa:observedProperty :eqMagnitude ; sosa:hasMember :eqMagObs1. [...]. :eqMagObs1 a sosa:Observation ; sosa:hasFeatureOfInterest :eqEvent1 ; sosa:resultTime “2017-02-01” ; sosa:hasSimpleResult “4.8” .	$16 \times (2 +$ 74,317) $+$ 4×16 $\times 74,317$ $=$ 5,945,392	SELECT ?result WHERE { ?eqMagObsCol a sosa:ObservationCollection ; sosa:observedProperty :eqMagnitude ; sosa:hasMember ?eqMagObs . ?eqMagObs a sosa:Observation ; sosa:hasFeatureOfInterest ?eqEvent ; sosa:resultTime X ; sosa:hasSimpleResult ?result . ?eqEvent geo:hasGeometry Y .}

Table 1: Comparison of three approaches to model earthquake observations. Notes: [...] refers to omission of subjects.

In summary, using aggregators to group observations into collections can reduce the redundancy embedded in environmental data. However, it might increase the complexity of specific queries as well. Last but not least, different types of aggregators have various levels of performance in terms of saving triples for a specific data. Therefore, ontology engineers have to be careful when selecting aggregators for an environmental data. Our work provides some options.

5 Conclusion

Environmental data is ubiquitous on the Web and key to many global challenges such as climate change and disaster response. Unfortunately, this data can be siloed for a number of reasons, where (arguably) primarily, there is not an agreed upon *efficient* method for making the data FAIR and 5-Star Quality, although some W3C and OGC standards exist. To address this problem, we have identified ways, using SOSA and its extension, in order to richly, efficiently, and semantically model observations and their collections. That is, we identify so-called *aggregators* that allow for a significant reduction in captured data by aggregating observations along certain dimensions.

However, if we apply too complicated aggregations on top of the data, we might create a very long property path from a feature of interest to its corresponding observation results, which will lead to the inefficiency of query. Basically, to design an appropriate aggregator for a given dataset, we need to consider the trade-off between the size of the graph (space complexity) and the length of the property paths (query efficiency). In the future, we plan to (1). compare SOSA with other standards for modeling environmental data; (2) work with domain environmental scientists to design a guideline of choosing the right aggregator to build observation collections based on real world applications; (3). investigate new metrics to compare query performance; and (4). adopt the concept of collection in the modeling of other types of scientific data, such as spatial features (e.g, those having multi-polygon geometries).

References

1. Cox, S.: Extensions to the semantic sensor network ontology. Tech. rep., W3C (Jan 2020), https://www.w3.org/TR/2020/WD-vocab-ssn-ext-20200116/
2. Cyganiak, R., Wood, D., Lanthaler, M. (eds.): RDF 1.1 Concepts and Abstract Syntax. W3C Recommendation 25 February 2014 (2014)
3. Haller, A., Janowicz, K., Cox, S.J., Lefrançois, M., Taylor, K., Le Phuoc, D., Lieberman, J., García-Castro, R., Atkinson, R., Stadler, C.: The modular ssn ontology: A joint w3c and ogc standard specifying the semantics of sensors, observations, sampling, and actuation. Semantic Web **10**(1), 9–32 (2019)
4. Hasnain, A., Rebholz-Schuhmann, D.: Assessing fair data principles against the 5-star open data principles. In: European Semantic Web Conference. pp. 469–477. Springer (2018)
5. Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P.F., Rudolph, S. (eds.): OWL 2 Web Ontology Language: Primer (Second Edition). W3C Recommendation 11 December 2012 (2012)
6. Janowicz, K., Haller, A., Cox, S.J., Le Phuoc, D., Lefrançois, M.: Sosa: A lightweight ontology for sensors, observations, samples, and actuators. Journal of Web Semantics **56**, 1–10 (2019)
7. Janowicz, K., Hitzler, P., Adams, B., Kolas, D., Vardeman II, C.: Five stars of linked data vocabulary use. Semantic Web **5**(3), 173–176 (2014)
8. Kublauch, H., Kontokostas, D. (eds.): Shapes Constraint Language (SHACL). W3C Recommendation 20 July 2017 (2017)
9. Madin, J., Bowers, S., Schildhauer, M., Krivov, S., Pennington, D., Villa, F.: An ontology for describing and synthesizing ecological observation data. Ecological informatics **2**(3), 279–296 (2007)
10. Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.W., da Silva Santos, L.B., Bourne, P.E., et al.: The fair guiding principles for scientific data management and stewardship. Scientific data **3**(1), 1–9 (2016)