Semantic Web X (2018) 1–0X
IOS Press

The SOSA/SSN Ontology: A Joint W3C and OGC Standard Specifying the Semantics of Sensors, Observations, Actuation, and Sampling

Armin Haller [a,*], Krzysztof Janowicz [b], Simon J D Cox [c], Maxime Lefrançois [d], Kerry Taylor [a], Danh Le Phuoc [e], Joshua Lieberman [f], Raúl García-Castro [g], Rob Atkinson [h], and Claus Stadler [i]

[a] *Research School of Computer Science, Australian National University, Canberra, Australia*
E-mails: armin.haller@anu.edu.au, kerry.taylor@anu.edu.au
[b] *Geography Department, University of California, Santa Barbara, CA, USA*
E-mail: jano@geog.ucsb.edu
[c] *Land and Water, CSIRO, Melbourne, Australia*
E-mail: simon.cox@csiro.au
[d] *Univ Lyon, MINES Saint-Étienne, CNRS, Laboratoire Hubert Curien UMR 5516, Saint-Étienne, France*
E-mail: maxime.lefrancois@emse.fr
[e] *Open Distributed Systems, Technische Universität Berlin, Germany*
E-mail: danh.lephuoc@tu-berlin.de
[f] *Center for Geographic Analysis, Harvard University, Boston, MA, USA*
E-mail: jlieberman@fas.harvard.edu
[g] *Ontology Engineering Group, Universidad Politécnica de Madrid, Madrid, Spain*
E-mail: rgarcia@fi.upm.es
[h] *Metalinkage, Wollongong, Australia*
E-mail: rob@metalinkage.com.au
[i] *Institut für Informatik, Universität Leipzig, Leipzig, Germany*
E-mail: cstadler@informatik.uni-leipzig.de

Abstract. The joint W3C (World Wide Web Consortium) and OGC (Open Geospatial Consortium) *Spatial Data on the Web* (SDW) Working Group developed a set of ontologies to describe sensors, actuators, samplers as well as their observations, actuation, and sampling activities. The ontologies have been published both as a W3C recommendation and as an OGC implementation standard. The set includes a lightweight core module called SOSA (Sensor, Observation, Sampler, and Actuator) available at: http://www.w3.org/ns/sosa/, and a more expressive extension module called SSN (Semantic Sensor Network) available at: http://www.w3.org/ns/ssn/. Together they describe systems of sensors and actuators, observations, the used procedures, the subjects and their properties being observed or acted upon, samples and the process of sampling, and so forth. The set of ontologies adopts a modular architecture with SOSA as a self-contained core that is extended by SSN and other modules to add expressivity and breadth. The SOSA/SSN ontologies are able to support a wide range of applications and use cases, including satellite imagery, large-scale scientific monitoring, industrial and household infrastructures, social sensing, citizen science, observation-driven ontology engineering, and the Internet of Things. In this paper we give an overview of the ontologies and discuss the rationale behind key design decisions, reporting on the differences between the new SSN ontology presented here and its predecessor [10] developed by the W3C Semantic Sensor Network Incubator group (the SSN-XG). We present usage examples and describe alignment modules that foster interoperability with other ontologies.

Keywords: Ontology, Sensor, Actuator, Observation, Actuation, Sampling, Linked Data, Web of Things, Internet of Things

1. Introduction

Sensors are a major source of data available on the Web today. The trend towards making cities, offices, and homes 'smarter' by turning them into sensor-rich environments [25] drives the demand for specifications that describe how to model and publish sensor and actuator data as well as how to foster interoperability across platforms on the Web or other data infrastructures.

Sensor readings are often provided only as raw numeric values, but any searching, reusing, integrating, or interpreting of these data requires more than just the observation results. Of equal importance for the proper interpretation of these values is contextual information about the studied feature of interest, such as a river, the observed property, such as flow velocity, the utilized sampling strategy, such as the specific locations or sampling stations and times at which the velocity was measured, the procedures followed to produce a result, and a variety of other information.

The Open Geospatial Consortium's (OGC) Sensor Web Enablement (SWE) standards [1, 7, 42] provide a framework for describing sensors and observations, as well as encodings for their data and service interfaces for interacting with them. SWE implementations followed the style of other OGC standards, relying on web-hosted service calls and XML payloads, which have limited compatibility with the more recent and widely used web platforms. A W3C Incubator Group was created in 2009 with the goal, among others, of defining an OWL ontology that would, to some degree, reflect the SWE standards, and produced the original Semantic Sensor Network ontology (SSNX[1]) [29]. This was consistent with W3C endorsed best practices on publishing data on the Web, which recommend a Linked Data approach, allowing sensor data, for example, to be published as part of a global and densely interconnected graph of data [34].

Since then, the SSNX ontology has been the basis for multiple research and standardization initiatives. Users of the original ontology have also identified a number of potential points of improvement, as well as extensions to cover new aspects of sensing that have become more relevant, such as actuation, a key element of the Internet of Things (IoT). Some inconsistencies with other vocabularies such as O&M were also identified.

Four years after the publication of this first version, work started on an update and formal standardization of the SSNX ontology, this time jointly led by the W3C and the OGC, to address the feedback gathered on the usage of the ontology and lessons learned. This activity resulted in the publication of the *new* Semantic Sensor Network (SSN) modular ontology, designed to provide a flexible but coherent perspective for representing the entities, relations, and activities involved in sensing, sampling, and actuation [21]. The main innovation of this generation of SSN has been the introduction of the Sensor, Observation, Sample, and Actuator (SOSA) ontology, which provides a lightweight core for SSN. SOSA aims at broadening the target audience and application areas that can make use of Semantic Web ontologies. Other SSN modules add additional elements, additional ontological commitments, and/or clarify and support the alignment of SSN with other ontologies.

This paper does not aim at providing an exhaustive description of the SSN ontology, since that can be found in the specification [21], but to act as a primer that motivates the development decisions made and the design patterns followed while indicating the most substantial changes made since the initial release of the ontology. Throughout this paper we will introduce concepts from SSN (and SOSA) in detail, supported by examples of its use in a smart home case study.

The paper is structured as follows. Sec. 2 covers the origins of the SSN ontology. Sec. 3 explains the rationale behind the main changes to the core of SSN, with observations modeled as events alongside the related sampling and actuation activities which are introduced as well. The overall modular structure of the SSN ontology is described in Sec. 4. A general description of the *new*, simplified SSN ontology is provided in Sec. 5, along with examples and explanations for the changes from SSNX. Sec. 6 and 7 describe the horizontal and vertical ontology modules that gravitate around SSN, respectively. We provide an overview of what SSN is not indented to model in Sec. 8, and report in Sec. 9 on implementation evidence for SSN, which was crucial for it to become an OGC/W3C standard.

2. Original SSN and Antecedents

Here we review the origins of SSN and SOSA, in particular the relation to the original SSN ontology

*Corresponding author. E-mail: armin.haller@anu.edu.au.

[1] In the remainder of this paper, SSNX will be used for the original 2011 version of the ontology, whereas SSN will be used for the new version presented here.

published by the W3C Semantic Sensor Network Incubator Group [29] and work on the OGC's Sensor Web Enablement initiative (SWE) [38].

From 2002 onward, SWE developed into a generic framework for delivering sensor data, dealing with remote-sensing, moving platforms, and in-situ monitoring and sensing. The Sensor Observation Service (SOS) defines a standard HTTP query interface for sensor and observation data [9], following the pattern established by OGC starting with the Web Map Service [17]. The returned XML data conforms with the Sensor Model Language (SensorML) [7] and with OMXML [13]. The latter is an XML implementation of Observations and Measurements (O&M) [42]. SensorML and O&M provide complementary viewpoints. SensorML is 'provider-centric' and encodes details of the sensor along with a stream of (typically) raw observation data. In contrast, O&M was designed to be more 'user-centric', with the target of the observation and the observed property as first-class objects.

A key aspect of the O&M model is that it separately defines as feature types the description of the observation event, the real world feature of interest being observed, and the platform, procedure and/or system responsible for the observation. In particular, geometries or other spatial location properties can be attached separately to features of interest, observations, platforms, and sensors, so a single model can accommodate remote sensing and *ex-situ* observation scenarios alongside *in-situ* monitoring [26, §5.16][1, 14]. O&M also includes a model for sampling, since most practical observations are made on a subset of, or proxy for, the ultimate feature of interest.

O&M is only one of several similar conceptual models for observations and their results. A selection of these, also including OBOE [35], Sensei [4] and Seronto [47], were reviewed by the W3C Semantic Sensor Network Incubator Group, and guided the development of the SSNX ontology published in 2011 [29].

The SSNX ontology was ultimately built around a fundamental conceptual model implemented as an ontology design pattern called the Stimulus Sensor Observation (SSO) pattern [24] and also aligned to the Dolce-Ultralite upper ontology (DUL) [37]. SSO was intended to provide a minimal common ground for ontologies for the use on the Semantic Sensor Web.

Terms defined by SSNX are identified by URIs under the namespace http://purl.oclc.org/NET/ssnx/ssn#. In the following, we shorten this namespace with the prefix `oldssn:`.

Drawing on considerable implementation and application experience with SSNX, as well as with sensor and observation ontologies more broadly, the *new* SSN ontology presented here is set out to address changes in scope and audience, shortcomings of the initial work, as well as new technical developments. The sections below highlight the most important updates.

3. Observation, Sampling and Actuation events

Following the working group's Use Cases and Requirements analysis [26], the scope of the revised ontology has first been reduced by removing the concepts for stimulus, systems, measurement and system capabilities from the core, then expanded beyond sensors and their observations by including classes and properties for the closely linked concepts of actuation and sampling.

Figure 1 provides an overview of the classes and properties in the core of the SSN ontology, showing how the three applications use the same pattern.

3.1. Sensors and Observations

The core of the SSNX ontology placed the sensor stimulus as the critical 'event' in the observation process. While this is sound conceptual modeling from a theoretical point of view, in practice the stimulus class was rarely instantiated. This is because a stimulus triggers the sensor to perform an observation, and, thus, resides outside the scope of typical sensor and observation applications and use cases. In contrast, the revised model focuses on the complete observation as an event, completed when the result is available (see Sec. 7.1 for more details). The new model seems to be a closer match to the view of most practitioners, and also provides a pattern and terminology that can be used for sampling and actuation activities.

SOSA recognizes sosa:Sensors, that make sosa:Observations about some sosa:ObservableProperty of a sosa:FeatureOfInterest.

For example, the following snippet uses the ne SSN ontology to describe an observation of the electric consumption in the kitchen of an exemplary smart house #134 made by sensor #927[2].

[2]The complete ontology file for the example is available at: https://github.com/w3c/sdw/blob/gh-pages/ssn/integrated/examples/house134.ttl

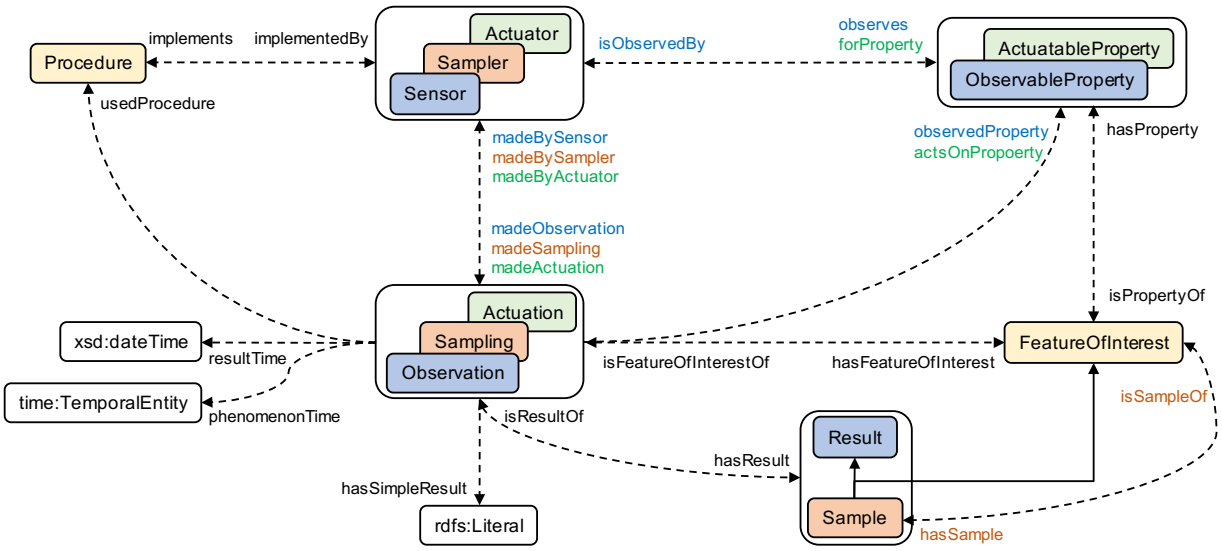

Fig. 1. Overview of the core structure of the SSN ontology, emphasizing the common patterns used by the three activities with classes stacked where they play a similar role. The elements shown are from both SOSA and SSN modules, with classes and types from external vocabularies indicated with a namespace prefix. A full set of inverse properties are defined in the ontology, but only a subset are shown in this figure.

```
@prefix cdt:
  <http://w3id.org/lindt/custom_datatypes#> .
BASE <http://example.org/>

<observation/235727> a sosa:Observation ;
  sosa:hasFeatureOfInterest <house/134/kitchen> ;
  sosa:observedProperty <electricConsumption> ;
  sosa:madeBySensor <sensor/927> ;
  sosa:hasSimpleResult "22.4 kWh"^^cdt:ucum .
```

As a result of replacing the SSO pattern in the old SSN by SOSA for the new SSN, the oldssn:Sensing class as a special kind of procedure (formerly oldssn:Process) has been removed. The ssn:Procedure class is now taking the place of oldssn:Sensing, while its definition is more generic, allowing it to be used for a description of the acts of observation, actuation and sampling (see Sec. 5.3).

3.2. Samplings and Samples

Almost all observations make use of sampling strategies, even if the sample is not explicitly described. Nevertheless, support for explicit modelling of samples is sometimes a design requirement, particularly in scientific applications. The SOSA ontology includes sosa:Samplers that make a sosa:Sampling of some sosa:FeatureOfInterest to produce a sosa:Sample.

For example, the following snippet describes an act of sampling using a spade to obtain a soil sample from the garden of our example house #134.

```
@prefix geow3c:
  <http://www.w3.org/2003/01/geo/wgs84_pos#> .

<sampling/4578> a sosa:Sampling ;
  geow3c:lat "-37.9076" ;
  geow3c:long "145.0294" ;
  sosa:madeBySampler <trowel> ;
  sosa:hasFeatureOfInterest <house/134/garden> ;
  sosa:resultTime
    "2017-12-04T08:14:00:00+10:00"^^xsd:dateTime ;
  sosa:hasResult <sample/134g1> .
```

The result of this act of sampling is a sosa:Sample.

```
<sample/134g1> a sosa:Sample ;
  sosa:isSampleOf <house/134/garden> ;
  sosa:isResultOf <sampling/4578> .
```

The relationship sosa:isSampleOf (and inverse sosa:hasSample) defines the link between a sample and the feature of interest that it represents, and is the essential property of a sample.

Note that an act of sampling is not required for a thing to serve as a sample. For example, the snippet below asserts that the kitchen and the bedroom are both samples of our home, which might be used in observations to approximate some global property of the house, such as its temperature.

```
<house/134/kitchen> a sosa:Sample ;
  sosa:isSampleOf <house/134> .

<house/134/bedroom> a sosa:Sample ;
  sosa:isSampleOf <house/134> .
```

3.3. Actuators and Actuations

There is an increasing importance of the Web of Things and smart instrumentation and environments more generally. Requirements for SSN included one to model actuation (it should be possible to model actuation functions of [systems].[3]) Actuation of a [system] is its ability to change something in its environment upon receiving a signal [26, §5.27]). The SOSA ontology pattern is therefore also extended to model sosa:Actuators, making some sosa:Actuations on some sosa:ActuatableProperty of a sosa:FeatureOfInterest.

For example, the following snippet uses SSN to describe actuation #188 that has been made by actuator windowCloser #987 to act on the state of window #104.

```
<actuation/188> a sosa:Actuation ;
  sosa:actsOnProperty <window/104#state> ;
  sosa:actuationMadeBy <windowCloser/987> ;
  sosa:resultTime
  "2017-04-18T17:24:00+02:00"^^xsd:dateTimeStamp.
```

3.4. Spatial aspects

Following the O&M model, spatial aspects of an act of observation, sampling or actuation can be associated with the sosa:FeatureOfInterest, the ssn:System (i.e. the sosa:Sensor, sosa:Sampler or sosa:Actuator) and/or the sosa:Platform on which they are mounted. Location may in fact constitute an sosa:ObservableProperty of an observation, one for example that uses a GPS sensor. Location and other spatial properties of these entities may be defined according to application- or community-specific models that in turn make use of appropriate geospatial ontologies such as GeoSPARQL [39].

4. Modularization

Practitioners using the original SSN have identified the complexity of both the ontology and its documentation as a significant usability issue. For example, SSNX is tied to Dolce-UltraLite, which introduces a strong entailment regime due to sub-class axioms. More generally, a monolithic ontology makes it difficult to focus on just those entities needed for a particular implementation. In response to this, the *new* SSN ontology is factored into several ontology subsets or modules that are similar in their domain of discourse and directly import one another as needed but differ in their ontological commitments and/or scope of coverage in order to suit different use cases and target audiences. The core module in particular, SOSA, is intended for data repositories and websites managed by web or data managers who need neither extensive axiomitization nor more specialized entities. The DL expressivity of the new lightweight SOSA core module is $\mathcal{ALI}(\mathcal{D})$ which is efficiently supported by modern triple stores, while that of the *new* SSN is $\mathcal{ALRIN}(\mathcal{D})$. In contrast, the expressivity of SSNX is $\mathcal{SRIQ}$.

Modularization has been accomplished by way of two types or directions of ontology segmentation as shown in Fig. 2.

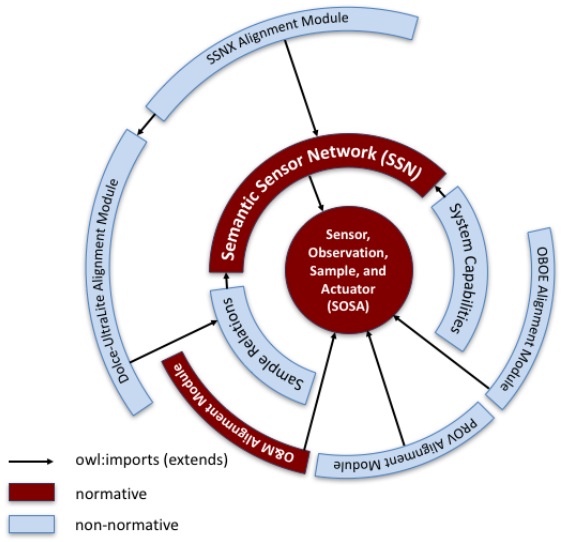

Fig. 2. SOSA/SSN ontologies "vertical" and "horizontal" modularization. Horizontal breadth of coverage is shown by arcuate modules at the same radius. Vertical height of expressivity is shown by modules at a larger radius

4.1. Vertical Segmentation

Vertically segmented SSN modules add higher levels of ontological commitment by directly importing lower modules and defining new axioms. The lower level modules are independent of the higher level modules, and logically consistent by themselves. The core SOSA module defines the key classes and properties but axiomatization is deliberately limited. In particular, SOSA uses no ob-

[3]The original requirement used term *sampling device* that has been dropped in SSN, as justified in Sec. 5.5.2.

ject property rdfs:domain or rdfs:range axioms. In place of these, schema.org schema:domainIncludes and schema:rangeIncludes annotations provide informal semantics to SOSA properties.[4] SOSA also avoids any subclass and subproperty axioms. The motivation for these choices is to make SOSA simple enough to be incorporated *as is* in the schema.org extension for IoT.[5]

The SSN module imports SOSA, and adds full axiomatization to the SOSA classes and properties, including rdfs:subClassOf, rdfs:subPropertyOf, owl:disjointWith and OWL cardinality and guarded existential restrictions on a class-level, along with some further classes and properties to complete the basic conceptual model corresponding approximately to the scope offered by SSNX.

The Dolce-UltraLite Alignment Module imports the Dolce-UltraLite Ontology and the Sample relations module, which in turn imports the SSN Ontology, and thereby transitively the SOSA Ontology. However, neither SOSA nor SSN import the Dolce-UltraLite Alignment Module in reverse. SOSA as the core, does not import any other ontologies, so is truly independent of vertical modules that add more terms, expressivity, and further ontological commitments to the lightweight semantics of SOSA.

Additional vertical modules provide alignments with other related ontologies, including O&M [1, 14], OBOE [35] and PROV-O [27].

4.2. Horizontal Segmentation

Modules that are horizontally layered may depend on each other, i.e., they may rely on the import of another horizontal module, but only extend scope by adding classes and properties, and do not otherwise enrich the semantics of existing terms. Two horizontal modules are defined in SSN (see Sec. 6.1): the Sample Relations Module and the System Capabilities Module.

5. The New SSN Ontology and its Core

SSNX was developed with ontology engineers as the primary audience in mind. Due to the widespread adoption of SSNX, the increasing role of citizen science, the strong focus on lightweight vocabularies by the Linked Data community, and the rising importance of simple vocabularies such as Schema.org, the SSNX ontology needed streamlining. The *new* SSN has been modularized so that a core ontology module - SOSA - can be used as a standalone ontology or even a simple vocabulary targeting web developers, citizen science, lightweight Linked Data publishing, resource-constrained IoT devices, and data intensive applications.

It consists of 13 classes, 21 object properties and 2 datatype properties, and includes very little axiomatization compared to SSNX. The SSN ontology is available at http://www.w3.org/ns/ssn/. It imports the SOSA ontology, adds additional axioms and 5 classes and 15 object properties to SOSA.

Terms defined by the core SOSA ontology are identified by URIs under the namespace http://www.w3.org/ns/sosa/, while those defined by the SSN ontology are identified by URIs under the namespace http://www.w3.org/ns/ssn/. In the rest of this article, we shorten these namespaces with the following prefixes (that are registered at http://prefix.cc).

```
@prefix sosa: <http://www.w3.org/ns/sosa/> .
@prefix ssn: <http://www.w3.org/ns/ssn/> .
```

SSN (with its imported SOSA core) is organized, conceptually, but not physically, into eight components. The different conceptual components of SOSA/SSN can be seen in Fig. 3. The core component with its three conceptual perspectives (Sensor, Actuator, Sampler) has already been described in Sec. 3. In the rest of this section we describe the other conceptual components, how they evolved from SSNX and the main rationale for this evolution.

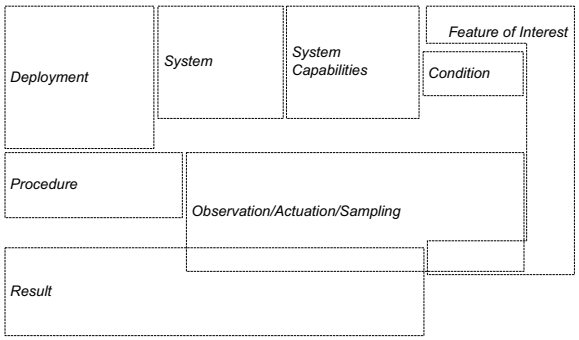

Fig. 3. Overview of the SSN ontology structure

[4] See https://www.w3.org/2015/spatial/wiki/Domain,_Range,_InverseOf for pointers to the group discussions on this topic.

[5] The Spatial Data on the Web group endorses SOSA being taken up by schema.org, see thread https://lists.w3.org/Archives/Public/public-sdw-wg/2017Jun/0067.html

5.1. Feature of Interest and Sample

In an ontology that aims to describe interactions between the physical and digital world, the target object in the physical world is of primary concern. Even though it is quite common in practice to report measurements of, and actions on, the physical world without *explicit* reference to a specific target object, the object is still necessarily there, cognitively completing the act of observation. In SSN, reflecting an OGC and ISO terminology heritage, the target object is called a sosa:FeatureOfInterest[6].

Property sosa:hasFeatureOfInterest is used to link between a sosa:Observation and its associated sosa:FeatureOfInterest. This property is a replacement for the oldssn:featureOfInterest, since the concept oldssn:FeatureOfInterest and the property oldssn:featureOfInterest were merely distinguished by case. The *new* SSN avoids this to make it easier to be used in languages without case distinction.

In the broader context of Spatial Data on the Web, a *Spatial Thing* could also be a sosa:FeatureOfInterest [44], as the next example shows.

```
<house/134> a sosa:FeatureOfInterest ;
  rdfs:comment "House #134."@en .

<floralCouch> a sosa:FeatureOfInterest ;
  rdfs:comment "The lumpy 2-seater couch with the
       floral pattern."@en .

<roofInsulation> a sosa:FeatureOfInterest ;
  rdfs:comment "The insulation material in the roof
       of house #134."@en .
```

SSN defines sosa:Sample as a subclass of sosa:FeatureOfInterest, because, when used for an observation, the sample is the (proximate) feature of interest, as the next example shows.

```
<observation/s1/5> a sosa:Observation ;
  sosa:hasFeatureOfInterest <sample/134g1> ;
  sosa:observedProperty <soil-pH> ;
  sosa:phenomenonTime [
   a time:Instant ;
   time:inXSDDateTimeStamp
    "2017-12-04T08:14:00+10:00"^^xsd:dateTimeStamp ;
   ] ;
  sosa:resultTime
   "2017-12-12T10:24:00+10:00"^^xsd:dateTime ;
  sosa:hasSimpleResult "6.1"^^cdt:pH .
```

In most circumstances the sample is only of interest in the context of observations, because it *represents* the (ultimate) feature of interest that is the object of real world or scientific interest within the investigation. In any observation we can query the property path[7] `sosa:hasFeatureOfInterest/sosa:isSampleOf*` to find the ultimate feature of interest for the observation. In the snippet shown above, which continues the example introduced in section 4.2, we can infer that the ultimate feature of interest is in fact the garden denoted `<house/134/garden>`.

A sosa:FeatureOfInterest and sosa:Sample will often have a specified location or other spatial properties, but this is not required. Observations can be made and samples taken from features for which the location is of no direct interest, for example a material sample, or a representative of a taxon, or certain communities.

5.2. Properties

An observation targets a feature of interest but interacts with it only through estimating the value of a characteristic or property of that feature. The general class of feature properties, ssn:Property, is defined in the SSN module, complementing its subclasses in SOSA, sosa:ObservableProperty and sosa:ActuatableProperty. Two *OWL/RDF* properties, ssn:hasProperty and ssn:isPropertyOf relate a feature of interest and its ssn:Property. We can state, for example, that air temperature is a property of each room in the house as follows:

```
<airTemperature> a ssn:Property ;
  sosa:isPropertyOf <house/134/bedroom>,
    <house/134/kitchen> .
```

Not all properties of a feature may be observable or actuatable characteristics of that feature. Some characteristics may also be modeled as a combination of a predicate and a literal, for example *hasColor "blue"*. SOSA/SSN focus on those feature properties that have a triple connection to observation [actuation] events: as a ssn:Property of the sosa:FeatureOfInterest of an sosa:Observation [sosa:Actuation], as the property being observed [actuated], and as the property observed [actuated] by the sensor [actuator] used in the observation [actuation]. Any of these property paths should be able to be inferred from the others, whether or not all are explicitly stated.

[6]The geospatial community uses the term 'feature' primarily to refer to discernable, identifiable objects in the landscape and their digital representations.

[7]See https://www.w3.org/TR/sparql11-query/#propertypaths for the definition of SPARQL property paths

5.2.1. Observable Properties

The multiplicity of property paths in which sosa:observedProperty is involved can lead to significant modeling and representation choices. On the one hand, as an inherent characteristic, a sosa:observedProperty is *specific* to a feature. On the other hand, we consider that multiple observations across different features of interest or by different sensors or both can measure the same *generic* property, e.g., air temperature. Thus, the air temperature in different rooms of a smart home might be observed as follows:

```
<observation/235714> a sosa:Observation ;
  sosa:hasFeatureOfInterest <house/134/bedroom> ;
  sosa:observedProperty <airTemperature> ;
  sosa:madeBySensor <sensor/926> .

<observation/235728> a sosa:Observation ;
  sosa:hasFeatureOfInterest <house/134/kitchen> ;
  sosa:observedProperty <airTemperature> ;
  sosa:madeBySensor <DHT22/4578#TemperatureSensor> .
```

It might even be that the same digital portable thermometer is being used to measure both air and water temperature (considered as an even more generic AmbientTemperature) alternately in both rooms.

The link from a sensor to the property that it observes, is made using the sosa:observes property, implying that every observation involving this sensor is about the same property. In the example above, sensor #926 is deployed only to observe the air temperature in the bedroom of our home, and we know that it made some observations.

```
<sensor/926> a sosa:Sensor ;
  sosa:observes <airTemperature> ;
  sosa:madeObservation <observation/235714>,
    <observation/235715>, <observation/235716> .
```

Since `<airTemperature>` is a generic Property and not specific to the bedroom, we cannot follow the sensor property path and infer which room's air temperature was being observed. In the case of a generic sosa:observedProperty, it may therefore not be possible to infer one property path from another and all may need to be specified. A specific Property could instead be used, such as `<airTemperature#house134bedroom>`, but it would then be more difficult either to express the capabilities of a portable sensor or to represent the generalization `<airTemperature>`.

One approach to modeling generalization relationships between individual properties might be to use the qudt:specialization and qudt:generalization properties from the QUDT vocabulary [23]. For example, various temperatures around the home may be modeled as follows.

```
@prefix qudt: <http://qudt.org/schema/qudt#> .

<temperature> a sosa:ObservableProperty,
     qudt:QuantityKind ;
  rdfs:label "Thermodynamic Temperature"@en .

<airTemperature> a sosa:ObservableProperty,
     qudt:QuantityKind ;
   qudt:generalization <temperature> .

<soilTemperature> a sosa:ObservableProperty,
     qudt:QuantityKind ;
   qudt:generalization <temperature> .

<waterTemperature> a sosa:ObservableProperty,
     qudt:QuantityKind ;
   qudt:generalization <temperature> .

<tapWaterTemperature> a sosa:ObservableProperty,
     qudt:QuantityKind ;
   qudt:generalization <waterTemperature> .

<poolWaterTemperature> a sosa:ObservableProperty,
     qudt:QuantityKind ;
   qudt:generalization <waterTemperature> .
```

Some ontology engineers (and some controlled vocabularies for quantity kinds) prefer to model such generalization-specialization relations using sub-class relations between observable property classes. In SSNX, for example, the class of observable properties was a subclass of dul:Quality and types of properties were usually modeled as subclasses of oldssn:ObservedProperty rather than as individuals.[8] Using either approach in SSN, it is still necessary to create an individual of whatever observed property class(es) is appropriate, whether that individual is still a generic property or is specific to an individual feature of interest. For example:

```
<AmbientTemperature> rdfs:subClassOf
     sosa:ObservableProperty .

<AtmosphericTemperature> rdfs:subClassOf
     <AmbientTemperature> .

<airTemperature> a <AtmosphericTemperature> .
```

[8] In SSNX, the class of observable properties was defined as a subclass of dul:Quality and as the class of "*observable Quality of an Event or Object. That is, not a quality of an abstract entity as is also allowed by DUL's Quality, but rather an aspect of an entity that is intrinsic to and cannot exist without the entity and is observable by a sensor.*" Section 5.3.1.3.4 in [29] further adds that types of properties, such as temperature or pressure should be added as subclasses of oldssn:ObservedProperty instead of individuals. Yet, usage reports [19] of SSNX revealed that most datasets were using observable properties as applicable to many features of interest. See https://www.w3.org/2015/spatial/wiki/What_is_an_instance_of_ssn:Property for more details on this point.

or

```
<AmbientTemperature> rdfs:subClassOf
     sosa:ObservableProperty .

<AtmosphericTemperature> rdfs:subClassOf
     <AmbientTemperature> .

<AirTemperature> rdfs:subClassOf
     <AtmosphericTemperature> .

<airTemperature32409845> a <AirTemperature> .
```

The subclass approach avoids introducing an additional vocabulary to support the specialization/generalization relationships, but usually at the cost of decreased flexibility in describing relations between defined properties. We expect different communities and applications to develop their own approaches to building catalogues of observable properties and choosing appropriate levels of specificity.

5.2.2. Stimulus as Proxies for Observing Properties

Sensors respond to stimuli, e.g., changes in the environment, or input data composed from the results of prior observations, to generate the result. The class ssn:Stimulus is unchanged from SSNX and is a proxy (i.e. ssn:isProxyfor) for an observable property, or a number of observable properties. For example, the amount of dynamic acceleration in an accelerometer as a proxy for the tilt angle of a smart phone. Properties themselves are observable characteristics of (i.e. ssn:isPropertyOf) real-world entities (i.e. ssn:FeatureOfInterests).

5.2.3. Actuatable Properties

The class sosa:ActuatableProperty is the parallel to observable properties for actuations, and instances of sosa:ActuatableProperty are usually generic to many features of interest (e.g., `<openCloseStatus>`), or may be specific to a single feature of interest (e.g., `<window/104#state>`).

For example, using a feature of interest -generic modeling choice for the previous example, one may model the open or close status as a first-class citizen.

```
<actuation/188> a sosa:Actuation ;
  sosa:hasFeatureOfInterest <window/104> ;
  sosa:actsOnProperty <window/104#state> ;
  sosa:hasSimpleResult true ;
  sosa:resultTime
   "2017-04-18T17:24:00+02:00"^^xsd:dateTimeStamp .
```

SOSA does not define a specific property to link an actuator to the property it acts on, but the SSN property ssn:forProperty can be used for this purpose.

5.3. Procedures and their Specification

Conceptually, an observation, sampling, or actuation, can be thought of as implementations (of parts) of a procedure. A sosa:Procedure is a reusable workflow, protocol, plan, algorithm, or computational method that can be used, among others, to specify how an observation activity has been made. In SSNX the name for sosa:Procedure was oldssn:Process. This name change is the result of extensive discussions within the group.[9]

The SSNX term oldssn:Sensing, that has proven to be prone to misinterpretation, has been deprecated. Its superclass sosa:Procedure should be used instead.

Using SSN, a sosa:Procedure can be linked to some description of the ssn:Input and ssn:Output of these procedures. For example, the following listing states that the output of the `<summingHourlyConsumptionProcedure>` procedure is the electricity consumption.

```
<summingHourlyConsumptionProcedure> a
     sosa:Procedure ;
  ssn:hasOutput <electricityConsumption> .
```

If the result of an observation is a web document having some representation (e.g., in JSON or XML) other ontologies such as the RDF Presentation ontology[10] [30] or RaUL [22] can be used to model the result using the ssn:hasOutput relation of SSN. It links the output of an observation to a description of how a result document can be validated (e.g., using some JSON Schema), or how an RDF description can be obtained from it (using some RDF lifting rule such as [32, 40]).

```
@prefix rdfp: <https://w3id.org/rdfp/> .

<summingHourlyConsumptionProcedure> a
     sosa:Procedure ;
  ssn:hasOutput [
    rdfp:presentedBy [
      rdfp:validationRule <consumption.schema.json> ;
      rdfp:liftingRule <lifting-rule.rqg> ] ] .

<observation/235727> a sosa:Observation ;
  sosa:hasResult <observation/235727/result> .
```

Procedures linked to observation and sampling activities, beyond the description of the presentation of the output, are typically a record of how these activities

[9] See https://www.w3.org/2015/spatial/track/issues/89 and https://www.w3.org/2015/spatial/wiki/Procedure_Process for a summary of the discussion and decision made on this topic.

[10] https://w3id.org/rdfp/

are performed (a log), and are linked to a sosa:Sensor or sosa:Sampler with the ssn:implements relation as in the listing below.

```
<sensor/927> a sosa:Sensor ;
  ssn:implements
       <summingHourlyConsumptionProcedure> .
```

Procedures linked to actuation activities, however, can be either a record of how the actuation has been performed or a description of how to interact with an actuator (i.e., the recipe for performing actuations). The ssn:System concept and its relations to activities through ssn:implementedBy/implements in combination with the inputs and outputs of a procedure can define the interface of how to interact with a sosa:Actuator. How much detail is provided to model inputs and outputs of the actuation procedure as well as the orchestration of multiple actuators is beyond the scope of SSN. Existing ontologies such as OWL-S [36] and execution protocols such as WSMX [20] can be used together with lower-level specifications such as the W3C Thing Description[11] to model these details.

5.4. Results

SSN offers two ways of attaching properties to activities that observe, actuate or sample them. For simple (though frequent) cases that merely require a literal typed with an appropriate datatype, one may use the sosa:hasSimpleResult datatype property. Alternatively, observation results can be modeled as individuals and linked to the observation using the object property sosa:hasResult. In cases where the observed property is a physical quantity, one may then use one of the several existing ontologies to model the result.

5.4.1. hasSimpleResult and hasResult

Revisiting our example observation from above, the datatype property ssn:hasSimpleResult can be used as in the listing below, which also uses a custom datatype (i.e. cdt:temperature) whose value space is some set of quantity values. However, such custom datatypes are not compatible with the OWL specifications, that restrict the set of datatypes that can be used and may lead to an error in an OWL reasoner.

```
<observation/235715> a sosa:Observation ;
  sosa:resultTime
       "2017-11-15T14:35:13Z"^^xsd:dateTime ;
  sosa:hasSimpleResult "23.9 DEG"^^cdt:temperature .
```

[11] https://w3c.github.io/wot-thing-description/

The ssn:hasResult object property allows us to make statements about the ssn:Result object (as shown in the listing below) by explicitly stating the unit of measurement for the value of the observation. Although it was not in the scope of the SSN specification to recommend any particular way of modeling results as quantity values, there exist several external vocabularies that are specifically designed for modeling quantity values as OWL individuals. Examples include the Quantities, Units, Dimensions and Data Types Ontologies (QUDT [23]) used in this listing or the Ontology of Units of Measure (OM [14]). With QUDT 1.1, a sosa:Result would be a qudt:QuantityValue. With OM 2, a sosa:Result would be an om:Measure or om:Point.

```
<observation/235716> a sosa:Observation ;
  sosa:hasResult [
    a qudt-1-1:QuantityValue ;
    qudt-1-1:unit qudt-unit-1-1:DegreeCelsius ;
    qudt-1-1:numericValue "22.9"^^xsd:double ] .
```

The two alternative patterns reduce the complexity of the path to link an observation to the actual literal that encodes some value for the result of this observation that was required in SSNX[12]. Compatibility with SSNX (see Sec. 7.2) is obtained through both oldssn:SensorOutput and oldssn:ObservationValue being subclasses of the new sosa:Result, and oldssn:observationResult being a subproperty of ssn:hasResult.

There have been discussions in the working group on potential actuation commands[13], but their inclusion has been deemed out of scope for the current version of SSN.

5.4.2. Result of a Sampling activity

sosa:Sample is a subclass of sosa:Result because a sample is the result of a sosa:Sampling activity. This is in addition to being a subclass of sosa:FeatureOfInterest so that it can serve as the target of a sosa:Observation.

5.4.3. Result Time and Phenomenon Time

SOSA distinguishes between the sosa:phenomenonTime and sosa:resultTime, the former being the time that the result of an observation, actuation, or sampling applies to the feature of interest, while the latter specifies the *instant* of time when

[12] Wiki page https://www.w3.org/2015/spatial/wiki/Storing_Observation_Value lists pros and cons on the options that were considered, and excerpts of the discussion within the group.

[13] See https://www.w3.org/2015/spatial/wiki/Result_and_Command for a summary of the discussion on this topic.

an act, such as an observation, was completed. The sosa:resultTime is a datatype property with rdfs:range xsd:dateTime to capture the time instant that the activity completed. The sosa:phenomenonTime is an object property with rdfs:range time:TemporalEntity [16], since it may be either an instant or interval, or even a temporal complex.

For example, the following snippet precises that the temperature was observed through April 15th 2017, and that the result was available 12 seconds after the end of this period.

```
@prefix time: <http://www.w3.org/2006/time#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<observation/235714>
  sosa:resultTime
    "2017-04-16T00:00:12+00:00"^^xsd:dateTimeStamp ;
  sosa:phenomenonTime [
    a time:Interval ;
    time:hasBeginning [
      a time:Instant ;
      time:inXSDDateTimeStamp
        "2017-04-15T00:00:00+00:00"^^xsd:dateTimeStamp
      ] ;
    time:hasEnd [
      a time:Instant ;
      time:inXSDDateTimeStamp
        "2017-04-16T00:00:00+00:00"^^xsd:dateTimeStamp
      ]
    ] .
```

The distinction between sosa:resultTime and sosa:phenomenonTime covers cases where the end of the observation, actuation, or sampling, activity is significantly different from that of the phenomenon that is observed or acted on. As a consequence, this covers use cases involving observations over a long period of time, late result obtainment due to long-lasting procedure, delays due to lengthy information travel (e.g., information traveling at the speed of light across long distances), and even forecasts - which are observations with a sosa:resultTime before the sosa:phenomenonTime [1, 14, 21, §7.2].

5.5. Systems and their deployments

The modelling of a sensor as a physical piece of technology and the way multiple sensors are attached to other such entities has been greatly simplified in SSN. Whereas SSNX distinguished between an oldssn:Sensor that could be any entity that performed oldssn:Sensing and an oldssn:SensingDevice that was a subclass of an oldssn:Device, i.e. a physical piece of technology, the *new* SSN drops the notion of an oldssn:Device as well as the notion of an oldssn:SensingDevice. The ssn:System class that already existed in SSNX takes the place of those two classes, i.e. it can be used if someone wants to explicitly model that a sosa:Sensor, sosa:Actuator or sosa:Sampler is a physical piece of technology. For similar reasons, the notion of a *sampling device* has been dropped[14].

5.5.1. Platforms and hosts

One or more sensors (as well as actuators and samplers) can be hosted or mounted on a sosa:Platform. Such platforms can also define the geometric properties, i.e., placement, of sensors in relation to one another. sosa:Platforms can also host other sosa:Platforms.

The properties oldssn:attachedSystem and oldssn:onPlatform have been renamed to sosa:hosts and sosa:isHostedBy, respectively, and they can now be used on both, the sosa:Platform and the sosa:System class to define that they host sensors, actuators or samplers.[15]

In the aforementioned smart home example, temperature observations can be defined to be made, for example, by a temperature sensor that is built-in/hosted by a Nest thermostat. The below listing shows how to model this relation.

```
<nest/D1AA22A8211> a sosa:Platform ;
  rdfs:label "3rd gen Nest Learning Thermostat
       D1AA22A8211"@en ;
  rdfs:comment "Nest Thermostat in bedroom of house
       #134"@en ;
  sosa:hosts <sensor/926> .

<sensor/926> a sosa:Sensor ;
  rdfs:label "Nest temperature sensor #1"@en ;
  sosa:observes <airTemperature> .
```

In this example, `<sensor/D1AA22A82/Nest>` is inferred to be an ssn:System, as only ssn:Systems can be hosted by sosa:Platforms.

5.5.2. Systems and Sub-systems

The system class has remained unchanged from SSNX apart from its relation to DUL (see Sec. 7.1) as a unit of abstraction for pieces of infrastructure that implement procedures and that are hosted by platforms. A system can have components (ssn:hasSubSystem) which are also systems. Classes and relationships related to system capabilities, operating ranges, and sur-

[14] See https://www.w3.org/2015/spatial/wiki/Link_between_platform_and_device for a summary of the discussion and decision made on this topic.

[15] Wiki page https://www.w3.org/2015/spatial/wiki/Terms provides an overview of how each term in SSNX has been dealt with, potentially renamed or simply deprecated in the *new* SSN.

vival ranges, under given conditions have been relegated to a separate horizontal module (see Sec. 6.1).

5.5.3. Deployment

An ssn:Deployment is an activity or a set of activities that encompass all phases in the lifecycle of a deployed system, such as, the installation, maintenance and decommissioning of the platform, sensors, actuators or samplers attached to that system. The class of ssn:Deployment describes the deployment of one or more systems (ssn:deployedSystem) or platforms (ssn:deployedOnPlatform) for a particular purpose. For example, a temperature sensor deployed on a wall, or a whole network of sensors deployed for an observation campaign.

The below listing shows an example of how to model a deployment (i.e. `<house/134/deployment>`) of two systems `<PCBBoard1>`, `<PCBBoard1>` that are deployed in the kitchen of our example house #134.

```
<house/134/kitchen> a sosa:Platform ;
   rdfs:label "House #134 Kitchen."@en ;
   rdfs:comment "House #134 Kitchen that hosts
        PCBoard1 and PCBoard2."@en ;
   sosa:hosts <PCBBoard1>, <PCBBoard2> .

<PCBBoard1> a ssn:System ;
  rdfs:label "PCB Board 1"@en ;
  rdfs:comment "PCB Board 1 hosts DHT22 temperature
       sensor #1."@en ;
  sosa:hosts <DHT22/1> .

<PCBBoard2> a ssn:System ;
  rdfs:label "PCB Board 2"@en ;
  rdfs:comment "PCB Board 2 hosts DHT22 temperature
       sensor #2."@en ;
  sosa:hosts <DHT22/2> .

<house/134/deployment> a ssn:Deployment ;
  rdfs:comment "Deployment of PCB Board 1 and 2 in
       the kitchen for the purpose of observing the
       temperature."@en ;
  ssn:deployedOnPlatform <house/134/kitchen> ;
  ssn:deployedSystem <PCBBoard1>, <PCBBoard2> ;
  ssn:forProperty <airTemperature> .
```

The oldssn:DeploymentRelatedProcess and oldssn:deploymentProcessPart have been deprecated as no usage of these terms has been reported.

6. Horizontal Extension Modules

Horizontal extension modules are ontologies that introduce new classes and relations on top of SSN.

6.1. The System Capabilities Module

The System Capabilities module, a.k.a. SSN-System, gathers classes and properties used to model system capabilities, operating range, and survival range. This part of SSNX was rarely used in implementations, hence specific effort has been made to clarify and simplify its documentation, along with providing illustrative examples. Terms defined by the SSN-System ontology are identified by URIs under the namespace http://www.w3.org/ns/ssn/systems/. We shorten this namespace with prefix `ssn-system:`.

```
@prefix ssn-system:
     <http://www.w3.org/ns/ssn/systems/> .
```

6.1.1. System Capabilities

An ssn:System may be linked to some ssn-system:SystemCapability, which in turn is linked to some ssn-system:SystemProperty: measurement, actuation, or sampling properties that describe the capabilities of the ssn:System such as its accuracy, latency, precision, etc. An ssn-system:SystemCapability can furthermore be linked to some ssn-system:Condition that define its validity context such as a temperature range. For example, the following snippet specifies that the DHT22 temperature sensor sensitivity is 0.1°C in normal temperature conditions.

```
@prefix schema: <http://schema.org/> .
@prefix qudt-unit-1-1:
     <http://qudt.org/1.1/vocab/unit#> .

<DHT22/4578#TemperatureSensor> a sosa:Sensor ,
     ssn:System ;
  rdfs:comment "The DHT22 #4578 embedded
       temperature sensor."@en ;
  ssn-system:hasSystemCapability
       <DHT22TempCapability> .

<DHT22TempCapability> a ssn:Property ,
     ssn-system:SystemCapability ;
  rdfs:comment "The capabilities of the temperature
       sensor in normal temperature conditions."@en;
  ssn-system:inCondition <normalTemp> ;
  ssn-system:hasSystemProperty
       <DHT22TempSensitivity> .

<normalTemp> a ssn-system:Condition ,
     schema:PropertyValue ;
  rdfs:comment "A temperature range of -40 to 80
       Celsius."@en ;
  schema:minValue -40.0 ;
  schema:maxValue 80.0 ;
  schema:unitCode qudt-unit-1-1:DegreeCelsius .

<DHT22TempSensitivity> a ssn:Property ,
     ssn-system:Sensitivity ,
     ssn-system:Resolution , schema:PropertyValue ;
  rdfs:comment "The sensitivity and resolution of
       the temperature sensor is 0.1 C in normal
       temperature and humidity conditions."@en ;
  schema:value 0.1 ;
  schema:unitCode qudt-unit-1-1:DegreeCelsius .
```

The Terms ssn-system:SystemCapability and ssn-system:SystemProperty are named after the SSNX

oldssn:SensorCapability and oldssn:SensorProperty terms, that were generalized so that the new ones apply to the more general concept of ssn:System instead of just sosa:Sensor.[16] Several sub-classes of ssn-system:SystemProperty are pre-defined: ssn-system:MeasurementRange, ssn-system:ActuationRange, ssn-system:Accuracy, ssn-system:DetectionLimit, ssn-system:Drift, ssn-system:Frequency, ssn-system:Latency, ssn-system:Precision, ssn-system:Resolution, ssn-system:ResponseTime, ssn-system:Selectivity, and ssn-system:Sensitivity. Most of these were already in SSNX but their definition was harmonized, simplified, and sometimes generalized when the term can be applied equally to a sosa:Sensor, sosa:Actuator or sosa:Sampler. A system property ssn-system:ActuationRange has been introduced as it was considered of high interest for actuators.

6.1.2. *Operating Range and Survival Range*

The pattern used to describe a system capability in terms of some of its property values under certain conditions is replicated to describe its operating range and survival range. An ssn-system:OperatingRange (resp. ssn-system:SurvivalRange) describes some normal ssn-system:OperatingProperty (resp. ssn-system:SurvivalProperty) of an ssn:System under some specified ssn-system:Conditions. For example, the power requirement or maintenance schedule of an ssn:System (resp. the system or its battery lifetime) under a specified temperature range. In the absence of an ssn-system:OperatingProperty (an ssn-system:SurvivalProperty), it simply describes the ssn-system:Conditions in which a System is expected to operate (under which the ssn:System can be exposed to without damage). The ssn:System continues to operate as defined by its ssn-system:SystemCapability. If, however, the ssn-system:OperatingProperty (resp. ssn-system:SurvivalProperty) is violated, the ssn:System is operating 'out of operating range' (resp. is 'damaged') and its ssn-system:SystemCapability specification may no longer hold.

Two sub-classes of ssn-system:OperatingProperty are pre-defined: ssn-system:MaintenanceSchedule and ssn-system:OperatingPowerRange. Two sub-classes of ssn-system:OperatingProperty are also pre-defined: ssn-system:SystemLifetime and ssn-system:BatteryLifetime.

6.1.3. *Extensibility of the System Capabilities Module*

As a matter of fact, the SSN System Capabilities module contains a predefined list of capabilities, operating ranges, and survival ranges, that were considered of high relevance for SOSA/SSN users. External ontologies may propose new such terms in their own namespace, or even reuse the design pattern to describe other characteristics of other kinds of systems (for example, the maximal bandwidth or payload of a communicating device in some conditions).

6.2. *The Sample Relations Module*

Support for samples and sampling is one of the major enhancements in SOSA. The defining property of a sosa:Sample is the sosa:isSampleOf relationship with the thing that it is intended to represent.

However, in many cases a sample also has a relationship with another sample or samples as part of a study or deployment [1, 42][26, §5.38]. The nature of the relationship is typically quite specific, for example:

- spatial, such as pixels within a remote-sensed scene, stations along a traverse, or specimens collected along a borehole
- specific fractions of a mixture, such as platelets from a blood sample
- specific ("biased") fractions of an assay sample, such as the fraction of a powder that passes a specific sieve grating, the fraction that is magnetically susceptible, or the fraction whose density is higher than a specified value
- non-specific ("un-biased") fractions of a sample ("splits")
- parts of a specimen, such as the leg of an insect, which in turn represents a taxon

The design of sub-sampling strategies is a key element of scientific investigations, and is a critical source of innovation. Therefore, generic relationships such as 'parent-child', or 'subset' are insufficient.

The Sample Relations module provides a scalable pattern for linking samples which also allows relationship details to be captured. This is accomplished by introducing an intermediate class in the relationship between samples, so that the nature of the relationship can be recorded as an annotation on the association.

In the following snippet, the nature of the relationship between a sub-sample of our soil and the

[16] See https://www.w3.org/2015/spatial/wiki/Measurement_and_Operating_properties_for_actuators and https://lists.w3.org/Archives/Public/public-sdw-wg/2017Mar/0233.html for more details on the discussions.

sample within which it is found is described in an rdfs:comment within a blank-node.

```
<sample/134g1/organics> a sosa:Sample ;
  rdfs:label "Soil sample 134g1 organic fraction" ;
  sampling:hasSampleRelationship [
    a sampling:SampleRelationship ;
      sampling:natureOfRelationship [
        a sampling:RelationshipNature ;
        rdfs:comment "organic fraction of material" ;
      ] ;
      sampling:relatedSample <sample/134g1> ;
    ] ;
  sosa:isResultOf <sampling/4650> .

<sampling/4650> a sosa:Sampling ;
   sosa:featureOfInterest <sample/134g1> ;
   sosa:usedProcedure
        <procedure/soil-organic-fraction/78> .

<procedure/soil-organic-fraction/78> a
     sosa:Procedure ;
   rdfs:comment "... details of procedure to
        separate the organic fraction of a soil
        sample ..." .
```

If there is a set of 'standard' relationships used within a particular discipline or community, these could be registered and assigned URIs. A reference to the standard relationship can then be used instead of the blank-node, as in this modified version of the example above.

```
<sample/134g1/organics> a sosa:Sample ;
  rdfs:label "Soil sample 134g1 organic fraction" ;
  sampling:hasSampleRelationship [
     a sampling:SampleRelationship ;
     sampling:natureOfRelationship
          <http://soil.example.org/rel/organic-fraction>
          ;
     sampling:relatedSample <sample/134g1> ;
    ] ;
   sosa:isResultOf <sampling/4650> .
```

The Sample Relations module is included in SSN/-SOSA as a non-normative horizontal extension.

7. Vertical Extension Modules

Vertical extension modules are ontologies that introduce additional expressivity or axiomatic constraints on top of SSN.

7.1. The Alignment Module to the DUL Upper Level Ontology

SSNX imported the Dolce-Ultralite upper ontology (DUL) [37] and many SSNX terms inherited semantics from their parent DUL terms. The DUL alignment contributed a level of complexity and abstraction which affected adoption in some communities. In line with the changes implemented for the *new* SSN, alignment between SSN terms and DUL terms was reconsidered and externalized to an optional vertical module called the Dolce-Ultralite Alignment module (SSN-DUL) that is available at http://www.w3.org/ns/ssn/dul. As a result, SSN (and SOSA) can now be used entirely independently of DUL. Those axioms in SSN that include DUL terms have been separated into the SSN-DUL module that imports SSN (and SOSA) and the DUL ontology.

7.1.1. Evolution of the alignment to DUL

In the process of separating SSN from DUL, some of the alignments with DUL terms have been reconsidered. In particular, SSNX has been criticized for its partially inconsistent handling of virtual sensors (including software and simulations) and related classes and properties. SSN and SOSA address this issue by allowing all major classes to be virtual, and to better support humans and other animals as agents that perform observations, actuation, or sampling activities. The sosa:Sensor, sosa:Platform, and ssn:System class have therefore been defined as a subclass of the higher-level dul:Object class rather than the more specific dul:PhysicalObject class.

The other notable changes in the alignment include the sosa:Observation and the sosa:Procedure class. In contrast to SSNX where observations have been modelled as a social construct (oldssn:Observation $\sqsubseteq$ dul:Situation), i.e. observations were contexts for interpreting incoming stimuli and fixing parameters such as time and location, observations in SSN are activities (sosa:Observation $\sqsubseteq$ dul:Event) (see Sec. 3), i.e. acts of carrying out an (observation) procedure in order to estimate or calculate a value of a sosa:ObservableProperty of a sosa:FeatureOfInterest or a sosa:Sample thereof. This modelling is aligned with other standard ontologies such as the OGC Observations and Measurements [1, 42] (i.e. sosa:Observation $\equiv$ `so19156-om:OM_Observation`) and the PROV Ontology (i.e. sosa:Observation $\sqsubseteq$ prov:Activity) where entities (i.e. sosa:Sensors in SSN) perform activities (i.e. sosa:Observations in SSN).

The sosa:Procedures (formerly oldssn:Process) is now defined as a subclass of dul:Method to describe a workflow that specifies how to make an observation, create a sample, or make a change to the state of the world via an actuator (see Sec. 5.3), rather than to model it as a dul:Process that describes the actual real-world process (not its description).

7.2. Alignment to the Old SSN Ontology

It is strongly recommended that applications that use the SSNX ontology are migrated to the *new* SSN ontology published at: http://www.w3.org/ns/ssn/. However, for legacy applications that continue to use SSNX, a new version of that ontology has been published at the old namespace: http://purl.oclc.org/NET/ssnx/ssn. This new version has been updated as follows:

- some errors with the previous version, such as the outdated import location for the DUL ontology, have been fixed;
- all axioms that pertain to the alignment of the SSNX to the Dolce-UltraLite ontology have been removed and an import of the DUL alignment module http://www.w3.org/ns/ssn/dul has been added instead;
- and mapping relations to the *new* SSN through owl:equivalentClass, owl:equivalentProperty and owl:subClassOf axioms have been added, while owl:deprecated annotation properties have been added to terms that were removed in the *new* SSN.

The June 2011 version of SSNX remains available at https://www.w3.org/2005/Incubator/ssn/ssnx/ssn, where either an HTML (ssn.html) or an RDF/XML (ssn.owl) representation will be served by the W3C server depending on the client preferences.

7.3. Alignment to OGC O&M

The observation and sample aspects of SSN are closely aligned to the OGC O&M model precedent [1, 14]. However, (a) OGC O&M is defined using UML and (b) some of the class and property names have been adjusted. Therefore, a formal (axiomatized) alignment to OGC O&M is provided as a vertical module which owl:imports SOSA, and uses the URI scheme defined by ISO 19150-2 to denote the O&M classes and properties [2].

The main conceptual difference between SSN and O&M is that the latter conflates both Procedures and Systems into a pair of classes: OM_Process and SF_Process. Alignment is achieved by way of the following axiom: iso19156-om:OM_Process $\equiv$ sosa:Sensor $\sqcup$ sosa-om:ObservationProcedure where sosa-om:ObservationProcedure $\sqsubseteq$ sosa:Procedure. Some other entities and properties of O&M, such as validTime, are not presently mapped to any terms of SOSA/SSN.

7.4. Alignment to OBOE

The OBOE ontology [35] is used by parts of the biodiversity community in particular to represent observations of traits or characteristics of organisms. A formal (axiomatized) alignment to OBOE is provided as a vertical module which owl:imports SOSA and OBOE.

In OBOE a set of oboe:Measurements of different characteristics relating to the same entity (usually an organism) are grouped into a single oboe:Observation. Thus, the alignment expresses this as a property-chain axiom: oboe:measurementFor $\circ$ oboe:ofEntity $\sqsubseteq$ sosa:hasFeatureOfInterest.

7.5. Alignment to PROV-O

The re-orientation of SSN so that observations (as well as samplings and actuations) are activities or "acts of sensing | sampling | actuation" means that SSN now clearly matches a process-flow model. This makes an alignment with PROV-O [27] quite natural, as foreseen in [11, 15]. A formal (axiomatized) alignment to PROV-O is provided as a vertical module which owl:imports SOSA and PROV-O. The key alignments are:

1. sosa:Observation $\sqsubseteq$ prov:Activity
   sosa:Sampling $\sqsubseteq$ prov:Activity
   sosa:Actuation $\sqsubseteq$ prov:Activity;
2. ssn:System $\sqsubseteq$ prov:Agent $\sqcap$ prov:Entity
   sosa:Sensor $\sqsubseteq$ prov:Agent $\sqcap$ prov:Entity
   sosa:Sampler $\sqsubseteq$ prov:Agent $\sqcap$ prov:Entity
   sosa:Actuator $\sqsubseteq$ prov:Agent $\sqcap$ prov:Entity.

The latter merits a little more explanation. When participating in an act of observation, sampling or actuation the sensors, samplers or actuators are responsible for the activity, so are thus agents. When not involved in the activity, they are merely entities which have to be maintained or stored.

8. Scope Limits of SSN

While the reworked SSN has achieved most of the goals set by the Charter and the Working Group chairs [46], and more, there are some concepts that are relevant to the intended scope of SSN but are not included in any of the present SOSA/SSN modules.

Since the publication of SSNX [29], the authors have been collecting feature requests from the user community. Due to lack of resources, only a few

of these requested features have been adopted into presently available modules of the *new* SSN. Of particular irony, and especially given the increased importance of IoT applications, is the absence of direct support for networks of sensors, such as device communications, network structures, relations between sensors, or specific data discovery and exchange interfaces. Some IoT-specific ontologies such as [5] and [6] (built on SSNX) as well as [18], and APIs such as [33] do cover some of these aspects. For network structure, the recursive ssn:System class may be a starting point, but network topology relations would still be required. Such relations can be modelled by extending by specializations of the concepts `InteractionPattern` and Link of the under-development standard of the W3C Web of Things Working Group, namely "Thing Description" [17].

Like SSNX, SSN does not include a taxonomy of sensor types. It does offer a small vocabulary of sensor descriptors which could be organized separately into a taxonomy. SSN considers humans-as-sensors to be first-class citizens (see the class definition for sosa:Sensor), but lacks terms for sensor capability applicable to human sensors such as in [21, Ex. 21]. Works from other domains, such as [41], could potentially fill this gap.

The new terms for actuation do not include actuation commands (analogous to sensor capabilities for observations). Other ontologies could add these concepts as a vertical module. For example, the Procedure Execution ontology (PEP), one of the core modules of the SEAS ontology [31] generalizes the core classes and properties of SSN to describe pep:ProcedureExecutor (sensor, actuator, web service, etc.) that implement pep:Procedure methods (sensing, actuating, forecasting, some algorithm) and make pep:ProcedureExecution activities (observation, actuation, web service execution, forecast), which may then be linked to some description of the command and/or the result.

As with SSNX, the new ontology does not itself provide terms for encoding locations (see Sec. 3.4) of sensors, observations, features, or other entities. Following the recommendations of the closely-related and concurrent work of the Spatial Data on the Web Best Practices [44], these entities can be considered as *Spatial Things* for which Best Practice 5 advises to "Provide geometries on the Web in a usable way", and summarises several well-known vocabularies for doing this in Appendix A, ranging from W3C-BASIC-GEO [8] to GeoSPARQL [39].

Finally, although SSN does allow for modelling the broader accuracy and precision capabilities of sensors (via the System capability module), it provides little support for modelling the uncertainty of particular data values, beyond providing the property ssn-system:qualityOfObservation.

9. Evaluation with Implementation Evidence

Since its publication in 2011 the original SSN ontology has been used in many IoT applications, linked datasets and ontologies. Examples of its use in linked datasets include meteorological models from the Spanish Meteorological Office [3], a case study on environmental sensing and livestock monitoring published by CSIRO [45], long-term climate observation data published by the Australian Bureau of Meteorology [28], real-time passenger data in the GetThere smartphone app [12] and fault analysis and worker support for cyber-physical production systems in a case study in the German Industrie 4.0 initiative [48].

The decision to form a working group to, amongst other things, formally standardize the SSN was a consequence of the interest expressed at a public workshop held in London in March 2015 https://www.w3.org/2014/03/lgd/ and the subsequent approval of the working group charter by both the W3C and the OGC members.

The use of SSN for data and applications published openly on the Web has been documented in detail by the Spatial Data on the Web working group in a note published at: https://w3c.github.io/sdw/ssn-usage/. This non-exhaustive list of usage of SSN has been obtained by crawling the LOD Laundromat, the LOD Cloud Cache, LODStats and the LOV Ontology repository, as well as through requests for implementation evidence on the Spatial Data on the Web working group mailing list. During the development of the *new* SSN between March 2016 and October 2017, the Spatial Data on the Web working group has also proactively encouraged and collected implementation evidence of all old (but equivalent) and new SOSA/SSN terms in the same document [19]. The minimal requirement for standardization of the SSN through the W3C was the use of each term in two consumer and two provider applications, i.e. applications

[17] https://www.w3.org/TR/wot-thing-description/

or datasets that use SSN to describe their data and ontologies that extend SSN, respectively.

At the time of its publication as a W3C candidate recommendation in July 2017, SOSA/SSN terms have been documented to be used in 23 open linked datasets, while also 23 openly published ontologies have been collected that use the SSN ontology to either describe their data or extend the SSN ontology, respectively. All terms have been used at least once in a dataset and an ontology, while 87% of terms have been used in at least two datasets and 81% of all terms have been used in at least two ontologies.

An example of an implementation of the newly introduced sampling terms in SSN is a register of several million geological samples at Geoscience Australia that provides descriptions using several alternative schemas and ontologies, including SOSA/SSN [18]. Some datasets and ontologies using the previous version of SSN have already been adapted to SOSA/SSN. For example a dataset about a meteorological station Irstea owns in one of its experimental farms between 2010 and 2015 was previously described in [43], and has recently been updated to SOSA[19].

10. Conclusion

The W3C/OGC Spatial Data on the Web Working Group has revisited the original SSN ontology that was published through a W3C Incubator Group in 2011, incorporated feedback from users of the ontology and extended it with terms to model sampling and actuation activities that have been identified as missing from the original ontology in many use cases. A new version of the ontology has been published as a W3C recommendation and an OGC implementation standard at: http://www.w3.org/ns/ssn/.

SSN was also perceived as too heavyweight (on its axiomatization) and too dependent on OWL reasoning by some users. To strike a balance, the complexity of the *new* SSN ontology has been reduced through a modularization that allows different user groups to pick and chose terms of the ontology appropriate for their domain of discourse and also chose between different levels of axiomatisation of the ontology.

The main novelty in the modularization of SSN has been the introduction of the lightweight core module called SOSA (Sensor, Observation, Sampler, and Actuator) which is available at: http://www.w3.org/ns/sosa/. SOSA and SSN together can now be used to describe sensors and their observations, the involved procedures, the studied features of interest, the samples used to do so, the feature's properties being observed or sampled, as well as actuators and the activities they trigger. SSN has already been broadly accepted as a de-facto standard for representing sensor data on the Web and has been the inspiration for multiple ontologies layered on top of SSN. With the standardization of the SSN and SOSA ontologies through the OGC and the W3C and the introduction of different modules for different audiences, we expect the ontology to be used in even more varied use cases, especially for use cases in the Internet-of-Things domain.

[18] http://pid.geoscience.gov.au/sample/

[19] http://ontology.irstea.fr/pmwiki.php/Site/Weather2017

Acknowledgements

The SSN ontology was developed through the OGC/W3C Spatial Data on the Web Working Group, see https://www.w3.org/2000/09/dbwg/details?group=75471&public=1 for the list of members. The efforts of W3C staff Phil Archer and François Daoust were invaluable in enabling the successful completion of the work through to publication as a W3C Recommendation and OGC Standard. The authors acknowledge partial support from NSF (award number 1540849).

References

[1] ISO 19156:2011 Geographic information - Observations and measurements. International Standard, International Organization for Standardization, Dec. 2011.

[2] ISO 19150-2:2015 Geographic information - Ontology – Part 2: Rules for developing ontologies in the Web Ontology Language (OWL). International Standard, International Organization for Standardization, July 2015.

[3] G. A. Atemezing, Óscar Corcho, D. Garijo, J. Mora, M. Poveda-Villalón, P. Rozas, D. Vila-Suero, and B. Villazón-Terrazas. Transforming meteorological data into Linked Data. *Semantic Web*, 4(3):285–290, 2013.

[4] P. Barnaghi, S. Meissner, M. Presser, and K. Moessner. Sense and sensability: Semantic data modelling for sensor networks. In *Proceedings of the ICT Mobile Summit*, pages 1–9, 2009.

[5] M. Bermudez-Edo, T. Elsaleh, P. Barnaghi, and K. Taylor. IoT-Lite Ontology. W3C Member Submission, World Wide Web Consortium, nov 2015. http://www.w3.org/Submission/iot-lite/.

[6] M. Bermudez-Edo, T. Elsaleh, P. Barnaghi, and K. Taylor. IoT-Lite: A Lightweight Semantic Model for the Internet of Things. In *2016 Intl IEEE Conferences on Ubiquitous Intelligence Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, pages 90–97, July 2016.

[7] M. Botts and A. Robin. OGC SensorML: Model and XML Encoding Standard. OGC Encoding Standard, Open Geospatial Consortium, Feb. 04 2014. version 2.0.

[8] D. Brickley. Basic Geo (WGS84 lat/long) Vocabulary. Technical report, W3C Semantic Web Interest Group, 2006.

[9] A. Bröring, C. Stasch, and J. Echterhoff. OGC Sensor Observation Service Interface Standard. OpenGIS Implementation Standard, Open Geospatial Consortium, Apr. 16 2012. version 2.0.

[10] M. Compton, P. Barnaghi, L. Bermudez, R. García-Castro, Óscar Corcho, S. J. D. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, V. Huang, K. Janowicz, W. D. Kelsey, D. L. Phuoc, L. Lefort, M. Leggieri, H. Neuhaus, A. Nikolov, K. Page, A. Passant, A. Sheth, and K. Taylor. The SSN ontology of the W3C semantic sensor network incubator group. *Web Semantics: Science, Services and Agents on the World Wide Web*, 17(0):25–32, 2012.

[11] M. Compton, D. Corsar, and K. Taylor. Sensor data provenance: SSNO and PROV-O together at last. In *CEUR: 7th International Conference on Semantic Sensor Networks*, pages 1–16, 2014.

[12] D. Corsar, P. Edwards, J. Nelson, C. Baillie, K. Papangelis, and N. Velaga. Linking open data and the crowd for real-time passenger information. *Web Semantics: Science, Services and Agents on the World Wide Web*, 43(Supplement C):18–24, 2017.

[13] S. J. D. Cox. Observations and Measurements - XML Implementation. OGC Encoding Standard, Open Geospatial Consortium, Mar. 22 2011. version 2.0.

[14] S. J. D. Cox. Geographic information - Observations and measurements. OGC Abstract Specification, Open Geospatial Consortium, Sept. 17 2013. version 2.0.

[15] S. J. D. Cox. Ontology for observations and sampling features, with alignments to existing models. *Semantic Web*, 8(3):453–470, 2017.

[16] S. J. D. Cox and C. Little. Time ontology in OWL. W3C Recommendation, World Wide Web Consortium, Oct. 2017.

[17] J. de la Beaujardiere. OpenGIS Web Map Server Implementation Specification. OpenGIS Implementation Specification, Open Geospatial Consortium, Mar. 15 2006. version 1.3.0.

[18] D. P. *et al*. Spitfire: Toward a semantic web of things. *IEEE Communications Magazine*, 49(11):40–48, 2011.

[19] R. García-Castro, A. Haller, and N. Mihindukulasooriya. On the usage of the SSN ontology. W3C note, World Wide Web Consortium, November 2017. https://w3c.github.io/sdw/ssn-usage/.

[20] A. Haller, E. Cimpian, A. Mocan, E. Oren, and C. Bussler. WSMX-a semantic service-oriented architecture. In *Web Services, 2005. ICWS 2005. Proceedings*, pages 321–328. IEEE, 2005.

[21] A. Haller, K. Janowicz, S. J. D. Cox, D. Le Phuoc, K. Taylor, and M. Lefrançois. Semantic Sensor Network Ontology. W3C Recommendation, World Wide Web Consortium, Oct. 19 2017.

[22] A. Haller, J. Umbrich, and M. Hausenblas. *RaUL: RDFa User Interface Language – A Data Processing Model for Web Applications*, pages 400–410. Hong Kong, China, 2010.

[23] R. Hodgson, D. Mekonnen, D. Price, J. Hodges, J. E. Masters, S. J. D. Cox, and S. Ray. Quantities, Units, Dimensions and Types (QUDT) Schema - Version 2.0. Technical report, qudt.org, Jan. 2017.

[24] K. Janowicz and M. Compton. The stimulus-sensor-observation ontology design pattern and its integration into the semantic sensor network ontology. In *Proceedings of the 3rd International Conference on Semantic Sensor Networks-Volume 668*, pages 64–78, 2010.

[25] R. Kitchin. The real-time city? big data and smart urbanism. *GeoJournal*, 79(1):1–14, 2014.

[26] F. Knibbe and A. Llaves. Spatial Data on the Web Use Cases & Requirements. W3C Working Group Note, World Wide Web Consortium, Oct. 25 2016.

[27] T. Lebo, S. Sahoo, and D. McGuiness. PROV-O: The PROV Ontology. W3C Recommendation, World Wide Web Consortium, Apr. 30 2013.

[28] L. Lefort, A. Haller, K. Taylor, G. Squire, P. Taylor, D. Percival, and A. Woolf. The ACORN-SAT linked climate dataset. *Semantic Web*, 8(6):959–967, 2017.

[29] L. Lefort, C. Henson, and K. Taylor. Semantic Sensor Network XG Final Report. W3C Incubator Group Report, World Wide Web Consortium, June 28 2011.

[30] M. Lefrançois. Interopérabilité sémantique libérale pour les services et les objets. In *Actes de la 17ème conférence Extraction et Gestion des Connaissances (EGC'17)*, Grenoble, France, Jan. 2017.

[31] M. Lefrançois. Planned ETSI SAREF Extensions based on the W3C&OGC SOSA/SSN-compatible SEAS Ontology Patterns. In *Proceedings of Workshop on Semantic Interoperability and Standardization in the IoT, SIS-IoT,*, July 2017.

[32] M. Lefrançois, A. Zimmermann, and N. Bakerally. A SPARQL extension for generating RDF from heterogeneous formats. In *Proc. Extended Semantic Web Conference (ESWC'17)*, Portoroz, Slovenia, May 2017.

[33] S. Liang, C.-Y. Huang, and T. Khalafbeigi. OGC SensorThings API Part 1: Sensing. Technical Report 15-078r6, OGC, July 2016.

[34] B. F. Lóscio, C. Burle, and N. C. et al. Data on the web best practices. W3C recommendation, W3C, January 2017. https://www.w3.org/TR/dwbp/.

[35] J. Madina, S. Bowers, M. Schildhauer, S. Krivovc, D. Pennington, and F. Villa. An ontology for describing and synthesizing ecological observation data. *Ecological Informatics*, 2(3):279–296, 2007.

[36] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, T. Payne, et al. OWL-S: Semantic markup for web services. *W3C member submission*, 22:2007–04, 2004.

[37] C. Masolo, S. Borgo, A. Gangemini, N. Guarino, A. Oltramari, and L. Schneider. The wonderweb library of fundational ontologies and the dolce ontology. Technical report, loa-istc, 2003.

[38] Open Geospatial Consortium. Sensor web enablement. last accessed: 26/11/2017.

[39] M. Perry and J. Herring. OGC GeoSPARQL-a geographic query language for RDF data. *OGC Implementation Standard. Sept*, 2012.

[40] A. Polleres, T. Krennwallner, N. Lopes, J. Kopecký, and S. Decker. XSPARQL Language Specification. W3C Member Submission, World Wide Web Consortium, Jan. 2009.

[41] A. Ramzan, H. Wang, and C. Buckingham. Representing human expertise by the owl web ontology language to support knowledge engineering in decision support systems. *Innovation in Medicine and Healthcare 201*, 207:290–299, 2015.

[42] H. Rijgersberg, M. van Assem, and J. Top. Ontology of units of measure and related concepts. *Semantic Web*, 4(1):3–13, 2013.

[43] C. Roussey, S. Bernard, G. André, O. Corcho, G. De Sousa, D. Boffety, and J.-P. Chanet. Weather station data publication at irstea: an implementation report. In *Proceedings of the 7th International Workshop on Semantic Sensor Networks (SSN 2014) in conjunction with the 13th International Semantic Web Conference (ISWC 2014)*, Oct. 2014.

[44] J. Tandy, P. Barnaghi, and L. van den Brink et al. Spatial data on the web best practices. W3C note, W3C, September 2017. https://www.w3.org/TR/sdw-bp/.

[45] K. Taylor, C. Griffith, L. Lefort, R. Gaire, M. Compton, T. Wark, D. Lamb, G. Falzon, and M. Trotter. Farming the Web of Things. *IEEE Intelligent Systems*, 28(6):12–19, 2013.

[46] K. Taylor and E. Parsons. Where is everywhere: Bringing location to the web. *IEEE Internet Computing*, 19(2):83–87, Mar 2015.

[47] D. van der Werf, M. Adamescu, M. Ayromlou, N. Bertrand, J. Borovec, H. Boussard, C. Cazacu, T. V. Daele, S. Datcu, M. Frenzel, V. Hammen, H. Karasti, M. Kertesz, P. Kuitunen, M. Lane, J. Lieskovsky, B. Magagna, J. Peterseil, S. Rennie, H. Schentz, K. Schleidt, and L. Tuominen. Seronto a socio-ecological research and observation ontology: the core ontology. Alter-Net Deliverable 4.I6.D2, 2009.

[48] I. Zinnikus, A. Antakli, P. Kapahnke, M. Klusch, C. Krauss, A. Nonnengart, and P. Slusallek. Integrated semantic fault analysis and worker support for cyber-physical production systems. In *Proceedings of the IEEE 19th Conference on Business Informatics (CBI)*, Thessaloniki, Greece, 2017. IEEE.