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

Armin Haller, Krzysztof Janowicz, Simon J.D. Cox, Maxime Lefrançois, Kerry Taylor, Danh Le Phuoc, Joshua Lieberman, Raúl García-Castro, Rob Atkinson, Claus Stadler

To cite this version:

Armin Haller, Krzysztof Janowicz, Simon J.D. Cox, Maxime Lefrançois, Kerry Taylor, et al.. The Modular SSN Ontology: A Joint W3C and OGC Standard Specifying the Semantics of Sensors, Observations, Sampling, and Actuation. Semantic Web – Interoperability, Usability, Applicability, 2018, 10 (1), pp.9-32. 10.3233/SW-180320 . hal-01885335

HAL Id: hal-01885335
https://hal.science/hal-01885335v1

Submitted on 1 Oct 2018

Semantic Web - (2018) 0–24
IOS Press

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

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

Editor: Pascal Hitzler, Wright State University, USA
Solicited reviews: Eva Blomqvist, Linköping University, Sweden; Alessandro Oltramari, Bosch Research and Technology Center, USA; Freddy Lecue, Accenture Tech Labs Ireland, and INRIA, France

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 [9] 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 [31] 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 [6, 26, 49] 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. That group produced the original Semantic Sensor Network ontology (SSNX[1]) [9, 36]. SSNX was designed to be 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 [42]. The SSNX ontology has been the basis for many subsequent research initiatives and some operational deployments. However, users of the original ontology have identified a number of potential points of improvement, inconsistencies with related vocabularies such as the Ontology of units of Measure (O&M) [12, 49], 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).

Four years after the publication of the 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 [22]. The main innovation of this formal standard version of SSN is the separation of SSN into ontology modules, the central of which is the Sensor, Observation, Sample, and Actuator (SOSA) ontology, providing a lightweight core for SSN. SOSA aims to broaden the target audience and application areas that can make use of Semantic Web ontologies. Other SSN modules add additional terms, introduce 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 [22], but to motivate the development decisions 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. As will be detailed below, SOSA [30] is a self-contained pattern designed for reuse and to meet the demands of a larger, schema.org-style target audience. Hence, SOSA is only introduced here to a degree necessary to understand its relation to SSN and because SSN imports and extends SOSA classes and relations.

The remainder of the paper is structured as follows. Sec. 2 briefly reviews 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 SSN imports or that are imported by SSN, respectively. We provide an overview of what

---

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

[1] In this paper SSNX will denote the 2011 version of the ontology produced by the W3C Incubator, with SSN used for the new version standardized through W3C/OGC and described here.

SSN is *not* intended 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

The development of SSN and SOSA has been strongly influenced by the SSNX ontology [36] and models from the OGC's Sensor Web Enablement initiative (SWE) [5].

Starting in 2002 SWE developed a generic framework for delivering sensor data. The Sensor Observation Service defines a HTTP query interface for sensor and observation data [8], following the pattern successfully established by OGC starting with the Web Map Service [16]. Returned data conforms with the Sensor Model Language (SensorML) [6] and with the XML implementation of O&M [12, 49]. 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 defined classes for 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. Geometries or other spatial location properties can be attached separately to features of interest, observations, platforms, and sensors, so the model can accommodate remote sensing and *ex-situ* observation scenarios alongside *in-situ* monitoring [32, §5.16][13, 26]. 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 [43], Sensei [2] and Seronto [55], were reviewed by the W3C Semantic Sensor Network Incubator Group, and guided the development of the SSNX ontology [36].

The SSNX ontology was ultimately built around a fundamental conceptual model implemented as an ontology design pattern called the Stimulus Sensor Observation (SSO) pattern [29] that was aligned to the Dolce-Ultralite upper ontology (DUL) [45]. SSO was intended to provide a minimal common ground for ontologies for the use on the Semantic Sensor Web.

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

One such significant change in the extent of the audience of SSN has been the emergence of lightweight vocabularies such as schema.org and their increasing use on the Web [20] since SSNX had been published. It led to a requirement to provide a lightweight ontology with minimal ontological commitment [32, §5.22], the SOSA core, that can be used by Web developers to annotate their IoT APIs using simple serialization formats such as JSON-LD, Microdata or RDFa 1.1 without the need to import any other ontology, including upper level ontologies (as required by SSNX).

New technical developments included particularly the emergence of complex actuation devices on the Web of Things, such as home automation devices like Google Nest, personal assistants like Amazon Echo, Apple Siri or Microsoft Cortana and personal camera drones. To be able to model the smart instrumentation of these devices and the environments they operate in more generally, the actuator and actuation perspective has been added to SSN.

The sections below highlights these significant changes and some of the other changes in the new SOSA/SSN ontology.

3. Observation, Sampling and Actuation events

Following the working group's Use Cases and Requirements analysis [32], the scope of the revised ontology has first been reduced by removing the concepts for stimulus, systems, measurement and system capabilities from the core (either for reasons of lack of implementation evidence or in response to a requirement [32, §5.47]), then expanded beyond sensors and their observations by including classes and properties for the closely linked concepts of actuation [32, §5.16] and sampling [32, §5.16].

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 pro-

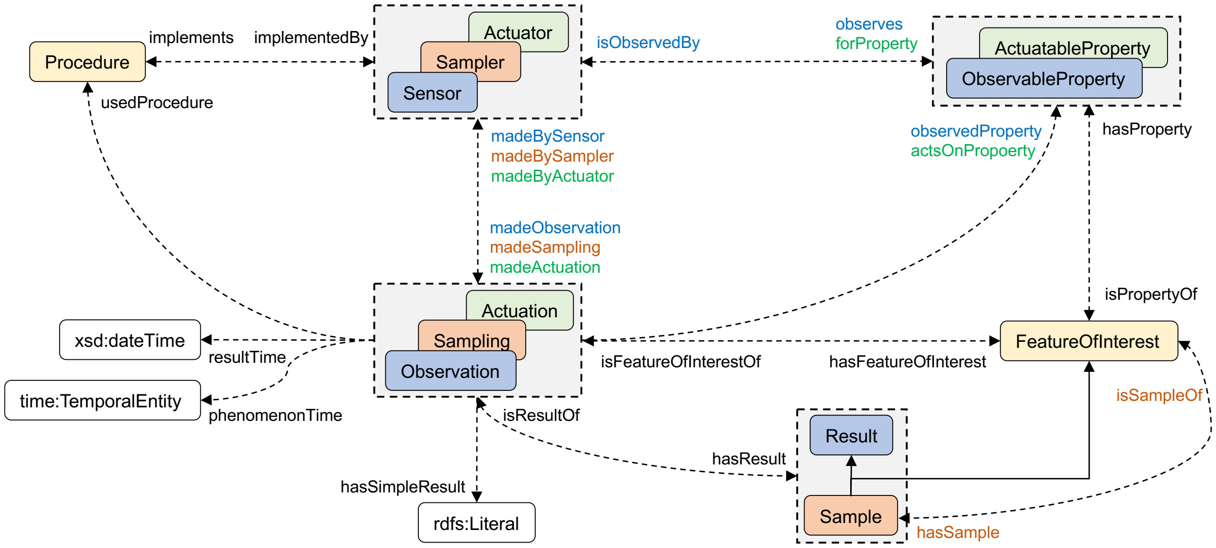

Fig. 1. Overview of the core structure of the SSN ontology, emphasizing the common patterns used by the three activities with classes stacked where they play a similar role. The elements shown are from both SOSA and SSN modules, with classes and types from external vocabularies indicated with a namespace prefix and uncolored. Classes and properties in blue relate to observations, red to sampling, and green to actuation, and classes in yellow are used across all three applications. A full set of inverse properties are defined in the ontology, but only a subset are shown in this figure.

cess. Observations have therefore 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. While this is sound conceptual modeling, 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 subsequent work focusing on the alignment of SSNX with the W3C Provenance ontology [34] the absence of a class representing an observation as an 'act of sensing' was noted [10, 14]. The revised model focuses on the complete observation as an event (sosa:Observation $\sqsubseteq$ dul:Event), completed when the result is available, 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 [13, 26] (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 new model is also in line with the view of most practitioners, e.g., the event-oriented O&M pattern used in INSPIRE implementations[2]. Furthermore, it provides a pattern and terminology that can be used analogically for sampling and actuation activities.

In the process of separating SSN from DUL, also the semantics of the sosa:Sensor class has been reconsidered. SSNX has been criticized for its partially inconsistent handling of virtual sensors (including software and simulations) and related classes and properties. In particular, the comment in the oldssn:Sensor class suggested that a sensor can also be "computational methods, a laboratory setup with a person following a method, or any other thing that can follow a Sensing Method to observe a Property". However, the oldssn:Sensor class was defined as a subclcass of dul:PhysicalObject, making the comment inconsistent with the class definition. SOSA/SSN addresses this issue (and requirement §5.19 [32]) by allowing all major classes to be virtual, allowing humans and other animals as agents that perform observations, actuation, or sampling activities. In the optional alignment to DUL, sosa:Sensors, but also sosa:Platforms, and ssn:Systems are now defined as a subclass of the higher-level dul:Object class rather than the more specific dul:PhysicalObject class.

---

[2] https://inspire.ec.europa.eu/id/document/tg/d2.9-o%26m-swe

The SOSA pattern models sosa:Sensors as entities that make sosa:Observations about some sosa:ObservableProperty of a sosa:FeatureOfInterest. From the viewpoint of foundational ontologies such as DOLCE [18], endurants participate in perdurants. Consequently, the sensors, features of interest, samples, and so forth participate in the observation event in different roles. For instance, a sensor (typically) transforms a stimulus into a result thereby also setting the temporal bounds (start and end) of an observation. The feature of interest is the bearer of the property under consideration, and so on. To improve readability and to stay in line with the literature about sensors and observations we will say that a sensor triggered, initiated, performed, or took, an observation about a property of some feature instead of speaking about endurants and their participation in perdurants. Finally, it is also important to differentiate between the conceptual model of an observation as an event and the data models and data structures used to represent such an observation as a record in a database, e.g., as served by a semantically-enabled Sensor Observation Service (SOS) [24, 28].

Throughout the paper we will use an exemplary smart house that has been modelled using the SSN ontology. Fig. 2 gives an overview of the SOSA/SSN instances[3] that we will refer to in the listings throughout the paper.

Listing 3.1 uses SOSA/SSN to describe an observation of the electric consumption in the kitchen of our exemplary smart house #134 made by sensor #927.

Listing 3.1: Observation Modelling

```
@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 from SSNX by SOSA in SSN, the oldssn:Sensing[4] class has been replaced by the sosa:Procedure class, which has a more generic definition that allows it to be used for acts of observation, sampling or actuation (see Sec. 5.3).

3.2. *Samplers and Sampling*

Almost all observations make use of sampling strategies to connect an observation event with its ultimate feature of interest, even if the sample and sampling event are not explicitly described. Nevertheless, support for explicit modelling of samples and sampling is sometimes a design requirement, particularly in scientific applications. The SOSA ontology includes sosa:Sampler that makes a sosa:Sampling of some sosa:FeatureOfInterest to produce a sosa:Sample. A sosa:Sample is both a sosa:FeatureOfInterest and a sosa:Result of sosa:Sampling.

For example, Listing 3.2 describes an act of sampling using a spade to obtain a soil sample from the garden of our example house #134.

Listing 3.2: Sampling Modelling

```
@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 as exemplified in Listing 3.3.

Listing 3.3: Result Sample Modelling

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

The relationship sosa:isSampleOf (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 that allows observation of a sample to lead in turn to an observed property of the feature of interest.

For example, Listing 3.4 asserts that the kitchen and the bedroom, which are both features of interest in their own right, serve also as samples of our home. They might be used in observations to approximate some global property of the house, such as its temperature.

---

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

[4] The following prefixes are used in the text - `sosa:` for SOSA; `ssn:` for SSN; and `oldssn:` for the old SSNX ontology which defined resources in the namespace http://purl.oclc.org/NET/ssnx/ssn#

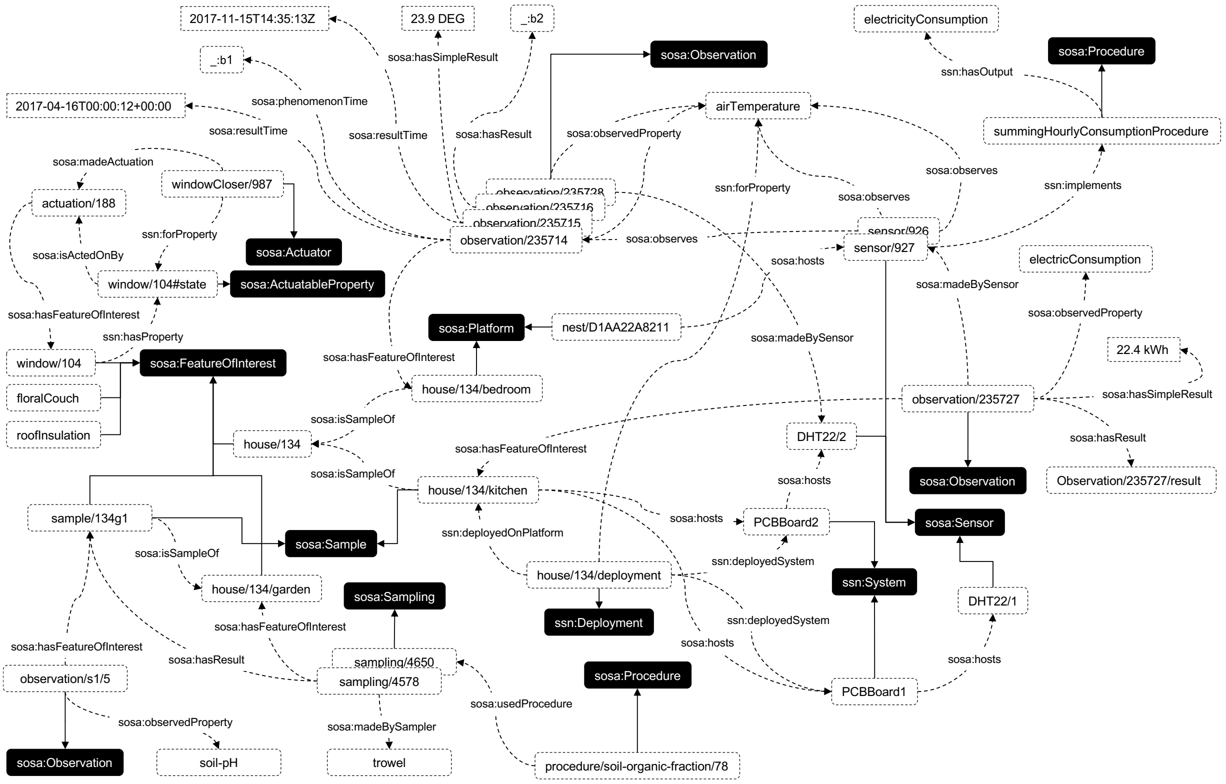

Fig. 2. Example smart house modelled using SOSA/SSN. Black boxes indicate SOSA/SSN classes, dashed boxes indicate class instances, solid lines represent `rdf:type` relations and dashed lines instances of properties.

Listing 3.4: Sample Modelling

```
<house/134> a sosa:FeatureOfInterest ;

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

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

3.3. *Actuators and Actuations*

As outlined above, there is an increasing importance of the Web of Things and smart instrumentation and environments more generally. Requirements for SSN included one to model actuations (“It should be possible to model actuation functions of sensing devices” [32, §5.27]). An actuation denotes a devices’ ability to change something in its environment upon receiving a signal. The SOSA ontology pattern is therefore extended to model sosa:Actuators, making some sosa:Actuations on some sosa:ActuatableProperty of a sosa:FeatureOfInterest.

For example, Listing 3.5 describes how actuation #188 has been made by actuator windowCloser #987 to act on the state of window #104.

Listing 3.5: Actuation Modelling

```
<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*

Spatial aspects of an observation, act of 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 also constitute a sosa:ObservableProperty of an observation, for example using a GPS sensor.

Location and other spatial properties of these entities may also be defined according to application- or community-specific models, which in turn make use of appropriate geospatial ontologies such as GeoSPARQL [47].

4. Modularization

Practitioners using the SSNX ontology have identified the complexity of both the ontology and its documentation as a significant usability issue. For example, SSNX imported the Dolce-Ultralite upper ontology (DUL) [45] 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. More generally, the monolithic ontology design of SSNX that introduces all terms within one ontology while also relying on the import of the Dolce-UltraLite ontology, makes it difficult for users 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.

Modularization has been accomplished by way of two types or directions of ontology segmentation as shown in Fig. 3.

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 object property rdfs:domain or rdfs:range axioms. In place of these, schema.org schema:domainIncludes and schema:rangeIncludes annotations provide formal semantics to SOSA properties.[5] 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.[6] 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.

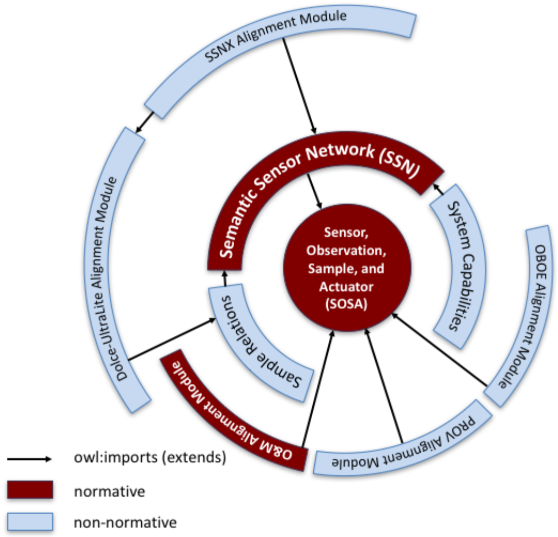

Fig. 3. SOSA/SSN ontologies "vertical" and "horizontal" modularization. Horizontal breadth of coverage is shown by arcuate modules at the same radius. Vertical height of expressivity is shown by modules at a larger radius

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.

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. Those axioms in SSN that include DUL terms have been separated into the SSN-DUL mod-

---

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

[6] 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

ule. 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. The SSN-DUL alignment module has been defined for backwards-compatibility reasons only and SSN (and SOSA) can now be used entirely independently of DUL. In fact, we are not anticipating the use of this alignment in applications other than the one's that are migrated from the old ontology.

Additional vertical modules (see Sec. 7 axiomatize alignments with other related ontologies, including O&M [13, 26], OBOE [43] and PROV-O [34].

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 the 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.

SOSA consists of 13 classes, 21 object properties and 2 datatype properties, and includes very little axiomatization. The SSN ontology is available at http://www.w3.org/ns/ssn/. It imports the SOSA ontology, adds additional axioms and 6 classes and 15 object properties. The DL expressivity of the new lightweight SOSA core module is $\mathcal{ALI}(\mathcal{D})$ which is efficiently supported by modern DL or RDFS+ reasoners [33, 46, 51], while that of the *new* SSN is $\mathcal{ALRIN}(\mathcal{D})$. In contrast, the expressivity of SSNX is $\mathcal{SRIQ}$. Table 1 provides some key facts to compare SOSA and SSN to SSNX.

Ontology	SOSA	SSN	SSNX
Classes	13	6	41
Object properties	21	15	39
Datatype properties	2	0	0
Tot. logical axioms	13	127	1011[7]
DL expressivity	$\mathcal{ALI}(\mathcal{D})$	$\mathcal{ALRIN}(\mathcal{D})$	$\mathcal{SRIQ}$

Table 1

Number of classes, object properties, and datatype properties, defined in SOSA, SSN, and SSNX (for comparison); DL expressivity of these ontologies.

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 perspectives. The core component with its three conceptual perspectives (Observations, Sampling, Actuation) 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. Table 2 in Appendix A lists all of the terms defined in SOSA and SSN, and the terms in the old SSN that they supersede, if they exist.

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 the target object is called a sosa:FeatureOfInterest, which reflects a terminology heritage from OGC[8].

The property sosa:hasFeatureOfInterest is used to link between a sosa:Observation and its associated sosa:FeatureOfInterest. This replaces oldssn:featureOfInterest, since the concept oldssn:FeatureOfInterest and the property oldssn:featureOfInterest were distinguished only 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, any *Spatial Thing* could also be a sosa:FeatureOfInterest [52] of an observation, sampling or actuation, as Listing 5.1 exemplifies.

Listing 5.1: Spatial Features of Interest Modelling

```
<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 Listing 5.2 shows.

Listing 5.2: Proximate Feature of Interest Modelling

```
<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[9] `sosa:hasFeatureOfInterest/sosa:isSampleOf*` to find the ultimate feature of interest for the observation. In Listing 5.2, which continues the example introduced in Sec. 3, 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 may represent a substance; an individual organism may be representative of a taxon; a statistical sample may represent a particular community that is not characterized by location, etc.

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. The ssn:hasProperty, and its inverse ssn:isPropertyOf, relate a feature of interest to ssn:Property. We can state, as shown in Listing 5.3, that air temperature is a property of each room in the house.

Listing 5.3: Property Modelling

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

Some properties of a feature may not be observable or actuatable characteristics, such as ‘name’. SOSA/SSN focus on those feature properties that have a triple connection to observation or actuation events: as an ssn:Property of the sosa:FeatureOfInterest of a sosa:Observation or sosa:Actuation, as the property being observed or actuated, and as the property usually observed or actuated by the sensor or actuator used in the observation or actuation. Any of these property paths should be able to be inferred from the others, whether or not all are explicitly stated.

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

---

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

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

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 shown in Listing 5.4.

Listing 5.4: Generic Observable Property Modelling

```
<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 <DHT22/2> .
```

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 Listing 5.5, sensor #926 is deployed only to observe the air temperature in the bedroom of our home, and we know that it made some observations.

Listing 5.5: Observable Property Modelling

```
<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 just follow the sensor property path and infer *which room's* air temperature was being observed: we need to know the sosa:hasFeatureOfInterest as well. 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 Quantities, Units, Dimensions and Data Types vocabulary [25]. For example, various temperatures around the home may be modeled as in Listing 5.6.

Listing 5.6: Generalised Property Modelling

```
@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.[10] 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 as shown in Listing 5.7 or Listing 5.8.

Listing 5.7: Specific Property Modelling

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

<AtmosphericTemperature> rdfs:subClassOf
     <AmbientTemperature> .
```

---

[10] 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 [36] 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.

```
<airTemperature> a <AtmosphericTemperature> .
```

Listing 5.8: Specific Property Modelling using subclassing

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

<AtmosphericTemperature> rdfs:subClassOf
    <AmbientTemperature> .

<AirTemperature> rdfs:subClassOf
    <AtmosphericTemperature> .

<airTemperature32409845> a <AirTemperature> .
```

The subclass approach shown in Listing 5.8 avoids introducing an additional vocabulary to support the specialization/generalization relationships, but potentially 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 and selecting one of the modeling approaches shown above.

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 of properties that can be acted on by actuators is sosa:ActuatableProperty. Instances of sosa:ActuatableProperty are usually generic to many features of interest (e.g., `<openCloseStatus>`), but they may be defined as 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 introduced in Listing 3.5, one may model the open or close status of the window as a first-class citizen as described in Listing 5.9.

Listing 5.9: Generic Feature of Interest Modelling

```
<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 acts that use(d) (parts) of a procedure. A sosa:Procedure is a reusable workflow, protocol, plan, algorithm, or computational method that can be used to specify how to make an observation, create a sample, or make a change to the state of the world (i.e. perform an actuation). This represents a significant departure from the semantics of the oldssn:Process class, defined as a subclass of dul:Process, and which described the real-world process, rather than a description of it. In the optional DUL alignment, a sosa:Procedure is now defined as 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). The oldssn:Sensing class, which was defined as a sub-class of a oldssn:Process has seen very little implementation evidence and has been deprecated in SSN.

This change addressed several requirements (e.g. [32, §5.25] and [32, §5.40]) and is the result of extensive discussions within the group[11]. The sosa:Procedure class now allows users to describe the protocol of Web services which themselves may represent ways on how to interact with, for example, sosa:Actuators. In particular, a sosa:Procedure can be linked to some description of the ssn:Input and ssn:Output of these procedures. For example, Listing 5.10 states that the output of the `<summingHourlyConsumptionProcedure>` procedure is the electricity consumption.

Listing 5.10: Output Modelling

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

---

[11] 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.

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[12] [37] or RaUL [23] 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 [39, 48] as shown in Listing 5.11).

Listing 5.11: Validation Rule Modelling

```
@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 are performed (a log), and are linked to a sosa:Sensor or sosa:Sampler with the ssn:implements relation as described in Listing 5.12.

Listing 5.12: Procedure Modelling

```
<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 [44] and execution protocols such as WSMX [21] can be used together with lower-level specifications such as the W3C Thing Description[13] 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

Listing 5.13 shows how to attach a literal value to our previous `<observation/235715>`. It 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, which restricts the set of datatypes that can be used and may consequently lead to an error in the OWL reasoner.

Listing 5.13: Simple Result Modelling

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

The ssn:hasResult object property allows us to make statements about the ssn:Result object (as shown in Listing 5.14) 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 QUDT ontology ([25]) used in this listing, or the Ontology of Units of Measure (OM [49]). With QUDT, a sosa:Result would be a qudt:QuantityValue. With OM, a sosa:Result would be an om:Measure.

Listing 5.14: Qualified Result Modelling

```
<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

---

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

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

result of this observation that was required in SSNX[14]. Compatibility with SSNX (see Sec. 7.4) 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[15], 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 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 and the result obtained. The sosa:phenomenonTime is an object property with range time:TemporalEntity [15], since it may be either an instant or interval, or even a temporal complex.

For example, Listing 5.15 describes that the temperature was observed through April 15th 2017, and that the result was available 12 seconds after the end of this period.

Listing 5.15: Phenomenon Time Modelling

```
@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 is important where the end of the observation, actuation, or sampling, activity is significantly different from that of the phenomenon that is observed or acted on. The separation covers cases involving observations over a long period of time, late results due to long-lasting procedures, delays due to lengthy information travel (e.g., information traveling long distances), cases where the result describes the state in the distant past (e.g. observations that described the pressure and temperature of a mineral at its time of formation), and predictions and forecasts - which may be defined as observations with a sosa:resultTime before the sosa:phenomenonTime [13, 22, 26, §7.2]. In both of the last two cases, it is common to have multiple observations relating to the same feature of interest, observed property and phenomenon time, but with different result times. For example, different procedures might be used in different geology labs. And the results of forecasts obtained using computational procedures might later be subject to validation by instrumental observations. The outcome of the latter would be encoded as sosa:Observations with the same sosa:hasFeatureOfInterest, sosa:observedProperty, and sosa:phenomenonTime, as the forecast, but with a different sosa:usedProcedure, sosa:madeBySensor and sosa:resultTime.

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

---

[14] 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.

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

similar reasons, the notion of a *sampling device* has been dropped[16].

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.[17]

Temperature observations can be defined to be made, for example, by a temperature sensor that is built-in/hosted by a Nest thermostat. Listing 5.16 shows how to model this relation in our smart home use case. In this example, `<sensor/D1AA22A82>` is inferred to be an ssn:System, as only ssn:Systems can be hosted by sosa:Platforms.

Listing 5.16: Platform Modelling

```
<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> .
```

5.5.2. *Systems and Sub-systems*

The system class has remained unchanged from SSNX apart from its relation to DUL (see Sec. 3.1) 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 survival 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.

Listing 5.17 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.

Listing 5.17: Deployment Modelling

```
<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

---

[16] 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.

[17] 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.

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/> .
```

Table 3 in Appendix B lists all of the terms defined in SSN-System, and the terms in the old SSN that they supersede, if they exist.

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, Listing 6.1 specifies that the DHT22 temperature sensor sensitivity is 0.1°C in normal temperature conditions.

Listing 6.1: System Capability Modelling

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

<DHT22/2> a sosa:Sensor ;
  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 $^{\circ}$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.[18] Several sub-classes of ssn-system:SystemProperty are pre-defined as listed in Table 3. Most of these were already defined 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. Some sub-classes of ssn-system:OperatingProperty and ssn-system:SurvivalProperty are also pre-defined, and listed in Table 3.

---

[18] 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.

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/SSN. 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 [26, 49][32, §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 Listing 6.2, the nature of the relationship between a sub-sample of our soil and the sample within which it is found is described in an rdfs:comment within a blank-node.

Listing 6.2: Sub-Sample Modelling

```
<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 described in a modified version of the example above in Listing 6.3.

Listing 6.3: Sample Blank Node Modelling

```
<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. Alignment to OGC O&M

The observation and sample aspects of SSN are closely aligned to the OGC O&M model precedent [13, 26]. 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 [27].

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.

7.2. *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 [34] quite natural, as foreseen in [10, 14]. 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.

7.3. *Alignment to OBOE*

The OBOE ontology [43] 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.

7.4. *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.

8. Topics out of Scope for SSN

While the reworked SSN has achieved most of the goals set by the Charter and the Working Group chairs [54], 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 [36], the authors have been collecting feature requests from the user community. Early in the standardization process the Working Group members were asked to solicit and document use cases from which requirements were derived and published by the W3C and OGC [41]. Of the 28 requirements for SSN noted there, 10 relate to the representation of temporal and spatial aspects of sensors and their observations. Like SSNX before it, a decision was made to defer to external ontologies for such modelling, and this is consistent with the advice of the Working Group in its Best Practice publication [52], which identifies some suitable ontologies ranging from W3C-BASIC-GEO [7] to GeoSPARQL [47]. Although this omission from SSN will have the effect of making SSN harder to use, and perhaps too lightly passes off demanding requirements for streaming measurements, mobile sensors and compact time series, there is no ideal single answer to this need. These requirements may be well served by followup primer publications currently planned, in preference to ontology extension.

Two requirements are met simply by virtue of the choice of a Web Ontology Model for SSN, and three others have been met by preserving SSNX capabilities. Nine have been met by the new work reported here. Arguably, a requirement for verifiable profiles has been partly met by the modularization presented here, and a requirement for examples in concert with external vocabularies has been met via the ontology mappings and examples presented in the specification and here. Multilingual annotation is being developed at present. The requirement for aggregated observations has not been specifically addressed.

Other informally requested features that have not been implemented include 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 [3] and [4] (built on SSNX) as well as [17], and APIs such as [40] do cover some of these aspects. The recursive ssn:System class might be a starting point for network nodes, but network relations would still be required. Such relations could perhaps be modelled by extending the concepts `InteractionPattern` and `Link` of the “Thing Description” [19] standard being developed by the W3C Web of Things Working Group.

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, one of the core modules of the Smart Energy Aware Systems ontology [38] 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.

9. Evaluation of Implementation Evidence

Since its publication in 2011 the SSNX 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 [1], a case study on environmental sensing and livestock monitoring published by CSIRO [53], long-term climate observation data published by the Australian Bureau of Meteorology [35], real-time passenger data in the GetThere smartphone app [11] and fault analysis and worker support for cyber-physical production systems in a case study in the German Industrie 4.0 initiative [56].

The decision to 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 was managed through a joint W3C-OGC working group.

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/. The list of usage of SSNX and SSN was 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 working group collected implementation evidence of all old (but equivalent) and new SOSA/SSN terms in the same document [19]. Standardization of the SSN through the W3C required evidence of use of each term in at least two consumer and two provider applications, i.e. applications 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 had been used in 23 open linked datasets, and 23 openly published ontologies were known to 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

---

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

terms have been used in at least two datasets and 81% of all terms have been used in at least two ontologies.

Community expectations were an important motivation for reconsideration of the core model. In particular, with the term “observation” now used for “event of observation” this allows for closer alignment of SSN to the O&M model. As a consequence, the OGC community is now examining adoption of SSN/SOSA for internet of things and linked data applications, for example in the ELFIE [20] activity.

The newly introduced sampling terms in SOSA are implemented in a register of several million geological samples at Geoscience Australia that provides descriptions using several alternative schemas and ontologies, including SOSA/SSN[21].

Some datasets and ontologies using SSNX have already been adapted to SOSA/SSN. For example, a dataset of measurements of a meteorological station, Irstea owns in one of its experimental farms, between 2010 and 2015 and that has previously been described using SSN [50] has since been updated to SOSA[22].

10. Conclusion

The W3C/OGC Spatial Data on the Web Working Group has redesigned the SSNX 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 an official W3C recommendation and an OGC implementation standard at: http://www.w3.org/ns/ssn/.

The initial SSNX was perceived as too heavyweight in 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 axiomatization of the ontology.

Another key novelty in the modularization of SSN has been the introduction of the lightweight and self-contained core pattern called SOSA (Sensor, Observation, Sampler, and Actuator) as a replacement of the initial Stimulus Sensor Observation (SSO) pattern 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.

SSNX has already been broadly accepted as a defacto standard for representing sensor data on the Web and has been the inspiration for multiple ontologies layered on top of SSNX. 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 new ontology to be used in even more varied use cases, especially in the Internet-of-Things domain.

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), CSIRO (Oznome project) and Marie Skodowska-Curie Programme H2020-MSCA-IF-2014 (SMARTER project, under Grant No. 661180).

References

[1] 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. https://doi.org/10.3233/SW-120089.

[2] 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, Santander, Spain, 2009. IIMC.

[3] 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/.

[4] M. Bermudez-Edo, T. Elsaleh, P. Barnaghi, and K. Taylor. IoT-Lite: A Lightweight Semantic Model for the Internet of Things. In *Proceedings of the International IEEE Conferences on Ubiquitous Intelligence Computing, Advanced and Trusted Computing, Scalable Computing and*

---

[20] http://www.opengeospatial.org/projects/initiatives/elfie

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

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

Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), pages 90–97, Toulouse, France, July 2016. IEEE. https://doi.org/10.1109/UIC-ATC-ScalCom-CBDCom-IoP-SmartWorld.2016.0035.

[5] M. Botts, G. Percivall, C. Reed, and J. Davidson. OGC sensor web enablement: Overview and high level architecture. Technical Report OGC 07-165, Open Geospatial Consortium, 2007. http://www.opengeospatial.org/ogc/markets-technologies/swe.

[6] M. Botts and A. Robin. OGC SensorML: Model and XML Encoding Standard. OGC Encoding Standard, version 2.0, Open Geospatial Consortium, Feb. 04 2014. http://www.opengeospatial.org/standards/sensorml.

[7] D. Brickley. Basic Geo (WGS84 lat/long) Vocabulary. Technical report, W3C Semantic Web Interest Group, 2006. https://www.w3.org/2003/01/geo/.

[8] A. Bröring, C. Stasch, and J. Echterhoff. OGC Sensor Observation Service Interface Standard. OpenGIS Implementation Standard, version 2.0, Open Geospatial Consortium, Apr. 16 2012. http://www.opengeospatial.org/standards/sos.

[9] 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. https://doi.org/10.1016/j.websem.2012.05.003.

[10] M. Compton, D. Corsar, and K. Taylor. Sensor data provenance: SSNO and PROV-O together at last. In *Joint Proceedings of the 6th International Workshop on the Foundations, Technologies and Applications of the Geospatial Web and 7th International Workshop on Semantic Sensor Networks*, volume 1401, pages 1–16, Riva del Garda, Italy, 2014. CEUR.

[11] 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. https://doi.org/10.1016/j.websem.2017.02.002.

[12] S. J. D. Cox. Observations and Measurements - XML Implementation. OGC Encoding Standard, Version 2.0, Open Geospatial Consortium, Mar. 22 2011. http://portal.opengeospatial.org/files/?artifact_id=41510.

[13] S. J. D. Cox. Geographic information - Observations and measurements. OGC Abstract Specification, Version 2.0, Open Geospatial Consortium, Sept. 17 2013. http://portal.opengeospatial.org/files/?artifact_id=41579.

[14] S. J. D. Cox. Ontology for observations and sampling features, with alignments to existing models. *Semantic Web*, 8(3):453–470, 2017. https://doi.org/10.3233/SW-160214.

[15] S. J. D. Cox and C. Little. Time ontology in OWL. W3C Recommendation, World Wide Web Consortium, Oct. 2017. https://www.w3.org/TR/owl-time/.

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

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

[18] A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, and L. Schneider. Sweetening ontologies with dolce. In *Proceedings of the International Conference on Knowledge Engineering and Knowledge Management*, pages 166–181, Siguenza, Spain, 2002. Springer. https://doi.org/10.1007/3-540-45810-7_18.

[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] R. Guha, D. Brickley, and S. Macbeth. Schema.org: Evolution of structured data on the web. *ACM Queue*, 13(9), 2015. https://doi.org/10.1145/2857274.2857276.

[21] A. Haller, E. Cimpian, A. Mocan, E. Oren, and C. Bussler. WSMX - a semantic service-oriented architecture. In *Proceedings of the International Conference on Web Services (ICWS)*, pages 321–328, Orlando, FL, USA, 2005. IEEE. https://doi.org/10.1109/ICWS.2005.139.

[22] 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. https://www.w3.org/TR/vocab-ssn/.

[23] A. Haller, J. Umbrich, and M. Hausenblas. RaUL: RDFa User Interface Language – A Data Processing Model for Web Applications. In *Proceedings of the 11th International Conference of Web Information Systems Engineering (WISE)*, pages 400–410, Hong Kong, China, 2010. LNCS. https://doi.org/10.1007/978-3-642-17616-6_36.

[24] C. A. Henson, J. K. Pschorr, A. P. Sheth, and K. Thirunarayan. SemSOS: semantic sensor observation service. In *Proceedings of International Symposium on Collaborative Technologies and Systems (CTS'09)*, pages 44–53, Baltimore, MD, USA, 2009. IEEE. https://doi.org/10.1109/CTS.2009.5067461.

[25] 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.

[26] ISO. ISO 19156:2011 Geographic information - Observations and measurements. International Standard, ISO, Dec. 2011.

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

[28] K. Janowicz, A. Bröring, C. Stasch, S. Schade, T. Everding, and A. Llaves. A RESTful proxy and data model for linked sensor data. *International Journal of Digital Earth*, 6(3):233–254, 2013. https://doi.org/10.1080/17538947.2011.614698.

[29] 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 Workshop on Semantic Sensor Networks*, volume 668, pages 64–78, Shanghai, China, 2010. CEUR.

[30] K. Janowicz, A. Haller, S. Cox, D. Le Phuoc, and M. Lefrançois. SOSA: A lightweight ontology for sensors, observations, samples, and actuators. *Journal of Web Semantics*, In Press. https://doi.org/10.1016/j.websem.2018.06.003.

[31] R. Kitchin. The real-time city? Big data and smart urbanism. *GeoJournal*, 79(1):1–14, 2014. https://doi.org/10.1007/s10708-013-9516-8.

[32] F. Knibbe and A. Llaves. Spatial Data on the Web Use Cases & Requirements. W3C Note, World Wide Web

Consortium, Oct. 25 2016. https://www.w3.org/TR/2016/NOTE-sdw-ucr-20161025/.

[33] M. Krötzsch. *OWL 2 Profiles: An Introduction to Lightweight Ontology Languages*, pages 112–183. LNCS, Vienna, Austria, 2012. https://doi.org/10.1007/978-3-642-33158-9_4,.

[34] T. Lebo, S. Sahoo, and D. McGuiness. PROV-O: The PROV Ontology. W3C Recommendation, World Wide Web Consortium, Apr. 30 2013. https://www.w3.org/TR/prov-o/.

[35] 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. https://doi.org/10.3233/SW-160241.

[36] L. Lefort, C. Henson, and K. Taylor. Semantic Sensor Network XG Final Report. W3C Incubator Group Report, World Wide Web Consortium, June 28 2011. http://www.w3.org/2005/Incubator/ssn/XGR-ssn-20110628/.

[37] 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.

[38] 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*, Amsterdam, The Netherlands, July 2017.

[39] M. Lefrançois, A. Zimmermann, and N. Bakerally. A SPARQL extension for generating RDF from heterogeneous formats. In *Proceedings of the Extended Semantic Web Conference (ESWC'17)*, Portoroz, Slovenia, May 2017. LNCS. https://doi.org/10.1007/978-3-319-58068-5_3.

[40] S. Liang, C.-Y. Huang, and T. Khalafbeigi. OGC SensorThings API Part 1: Sensing. Technical Report 15-078r6, OGC, July 2016. http://docs.opengeospatial.org/is/15-078r6/15-078r6.html.

[41] A. Llaves and F. Knibbe. Spatial data on the web use cases & requirements. W3C note, W3C, Oct. 2016. https://www.w3.org/TR/2016/NOTE-sdw-ucr-20161025/.

[42] B. F. Lóscio, C. Burle, and N. C. (editors). Data on the Web Best Practices. W3C recommendation, W3C, January 2017. https://www.w3.org/TR/dwbp/.

[43] 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. https://doi.org/10.1016/j.ecoinf.2007.05.004.

[44] 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, W3C, Nov. 2004. https://www.w3.org/Submission/OWL-S/.

[45] 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.

[46] Y. Nenov, R. Piro, B. Motik, I. Horrocks, Z. Wu, and J. Banerjee. RDFox: A Highly-Scalable RDF Store. In *Proceedings of the 14th International Semantic Web Conference (ISWC)*, volume 9367 of *LNCS*, pages 3–20, Bethlehem, PA, USA, October 11–15 2015. Springer. https://doi.org/10.1007/978-3-319-25010-6_1.

[47] M. Perry and J. Herring. OGC GeoSPARQL - A Geographic Query Language for RDF Data. OGC Implementation Standard 11-052r4, OGC, Sept. 2012.

[48] A. Polleres, T. Krennwallner, N. Lopes, J. Kopecký, and S. Decker. XSPARQL Language Specification. W3C Member Submission, World Wide Web Consortium, Jan. 2009. http://www.w3.org/Submission/2009/SUBM-xsparql-language-specification-20090120/.

[49] H. Rijgersberg, M. van Assem, and J. Top. Ontology of units of measure and related concepts. *Semantic Web*, 4(1):3–13, 2013. https://doi.org/10.3233/SW-2012-0069.

[50] 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 *Joint Proceedings of the 6th International Workshop on the Foundations, Technologies and Applications of the Geospatial Web and 7th International Workshop on Semantic Sensor Networks*, volume 1401, Riva del Garda, Italy, Oct. 2014. CEUR.

[51] J. Subercaze, C. Gravier, J. Chevalier, and F. Laforest. Inferray: fast in-memory RDF inference. *PVLDB*, 9(6):468–479, 2016. https://doi.org/10.14778/2904121.2904123.

[52] J. Tandy, P. Barnaghi, and L. van den Brink (editors). Spatial Data on the Web Best Practices. W3C Note, W3C, September 2017. https://www.w3.org/TR/sdw-bp/.

[53] 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. https://doi.org/10.1109/MIS.2013.102.

[54] K. Taylor and E. Parsons. Where is everywhere: Bringing location to the web. *IEEE Internet Computing*, 19(2):83–87, Mar 2015. https://doi.org/10.1109/MIC.2015.50.

[55] 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, Wageningen University, 2009.

[56] 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. https://doi.org/10.1109/CBI.2017.54.

Appendix A. List of all SOSA/SSN Terms

Appendix B. SSN-System terms

Appendix C. Deprecated SSNX Terms

SOSA/SSN term	replaces	SOSA/SSN term	replaces
sosa:FeatureOfInterest	⊒ oldssn:FeatureOfInterest	ssn:hasProperty	≡ oldssn:hasProperty
ssn:Property	≡ oldssn:Property	ssn:isPropertyOf	≡ oldssn:isPropertyOf
sosa:ObservableProperty	-		
sosa:ActuatableProperty	-		
sosa:Sample	-	sosa:hasSample	-
		sosa:isSampleOf	-
sosa:Platform	≡ oldssn:Platform	sosa:hosts	≡ oldssn:attachedSystem
		sosa:isHostedBy	≡ oldssn:onPlatform
sosa:Procedure	≡ oldssn:Process		
ssn:hasInput	≡ oldssn:hasInput	ssn:hasOutput	≡ oldssn:hasOutput
ssn:Input	≡ oldssn:Input	ssn:Output	≡ oldssn:Output
ssn:implementedBy	≡ oldssn:implementedBy		
ssn:implements	≡ oldssn:implements		
sosa:Sensor	≡ oldssn:Sensor	sosa:Actuator	-
sosa:observes	≡ oldssn:observes	ssn:forProperty	≡ oldssn:forProperty
sosa:isObservedBy	-		
sosa:Sampler	-		
sosa:usedProcedure	≡ oldssn:sensingMethodUsed	sosa:hasFeatureOfInterest	≡ oldssn:featureOfInterest
		sosa:isFeatureOfInterestOf	-
sosa:Observation	oldssn:Observation	sosa:Actuation	-
sosa:madeObservation	≡ oldssn:madeObservation	sosa:madeActuation	-
sosa:madeBySensor	≡ oldssn:observedBy	sosa:actuationMadeBy	-
sosa:observedProperty	≡ oldssn:observedProperty	sosa:actsOnProperty	-
		sosa:isActedOnBy	-
sosa:Sampling	-		
sosa:madeSampling	-		
sosa:madeBySampler	-		
ssn:Stimulus	≡ oldssn:Stimulus , ≡ oldssn:SensorInput	ssn:wasOriginatedBy	-
		ssn:detects	≡ oldssn:detects
		ssn:isProxyFor	≡ oldssn:isProxyFor
sosa:Result	⊒ oldssn:ObservationValue, ⊒ oldssn:SensorOutput	sosa:hasSimpleResult	-
sosa:hasResult	⊒ oldssn:hasValue, ⊒ oldssn:observationResult	sosa:hasResultingSample	-
sosa:isResultOf	≡ oldssn:isProducedBy	sosa:isSamplingResultOf	-
sosa:resultTime	oldssn:observationResultTime	sosa:phenomenonTime	≡ oldssn:observationSamplingTime
ssn:System	≡ oldssn:System	ssn:hasSubSystem	≡ oldssn:hasSubSystem
ssn:Deployment	≡ oldssn:Deployment		
ssn:deployedSystem	≡ oldssn:deployedSystem	ssn:deployedOnPlatform	≡ oldssn:deployedOnPlatform
ssn:hasDeployment	≡ oldssn:hasDeployment	ssn:inDeployment	≡ oldssn:inDeployment

TOTAL SOSA: 13 classes, 21 object properties, 2 datatype properties

TOTAL SSN: 6 classes, 15 object properties

Table 2

Complete list of SOSA and SSN terms, and the terms in the old SSN that they supersede, if they exist, along with the formal relation between the new term and the old term. No relation means rdfs:seeAlso. The grouping of terms is only meant to ease readability.

SOSA/SSN term	replaces	SOSA/SSN term	replaces
ssn-system:qualityOfObservation	≡ oldssn:qualityOfObservation		
ssn-system:inCondition	≡ oldssn:inCondition	ssn-system:Condition	≡ oldssn:Condition
ssn-system:hasSystemCapability	⊒ oldssn:hasMeasurementCapability		
ssn-system:SystemCapability	⊒ oldssn:MeasurementCapability		
ssn-system:hasSystemProperty	⊒ oldssn:hasMeasurementProperty		
ssn-system:SystemProperty	⊒ oldssn:MeasurementProperty		
System properties for sensors			
ssn-system:MeasurementRange	≡ oldssn:MeasurementRange	ssn-system:DetectionLimit	⊒ oldssn:DetectionLimit
System property for actuators			
ssn-system:ActuationRange	-		
System properties for any system			
ssn-system:Accuracy	⊒ oldssn:Accuracy	ssn-system:Drift	⊒ oldssn:Drift
ssn-system:Frequency	⊒ oldssn:Frequency	ssn-system:Latency	⊒ oldssn:Latency
ssn-system:Precision	≡ oldssn:Precision	ssn-system:Resolution	⊒ oldssn:Resolution
ssn-system:Repeatability	-	ssn-system:ResponseTime	⊒ oldssn:ResponseTime
ssn-system:Selectivity	⊒ oldssn:Selectivity	ssn-system:Sensitivity	≡ oldssn:Sensitivity
ssn-system:hasOperatingRange	≡ oldssn:hasOperatingRange		
ssn-system:OperatingRange	≡ oldssn:OperatingRange		
ssn-system:hasOperatingProperty	≡ oldssn:hasOperatingProperty		
ssn-system:OperatingProperty	≡ oldssn:OperatingProperty		
Operating ranges for any system			
ssn-system:MaintenanceSchedule	≡ oldssn:MaintenanceSchedule		
ssn-system:OperatingPowerRange	≡ oldssn:OperatingPowerRange		
ssn-system:hasSurvivalRange	≡ oldssn:hasSurvivalRange		
ssn-system:SurvivalRange	≡ oldssn:SurvivalRange		
ssn-system:hasSurvivalProperty	≡ oldssn:hasSurvivalProperty		
ssn-system:SurvivalProperty	≡ oldssn:SurvivalProperty		
Survival ranges for any system			
ssn-system:SystemLifetime	≡ oldssn:SystemLifetime		
ssn-system:BatteryLifetime	≡ oldssn:BatteryLifetime		

TOTAL SSN-System: 23 classes, 8 object properties

Table 3

Complete list of terms in SSN-System, and the terms in the old SSN that they supersede, if they exist, along with the formal relation between the new term and the old term. The grouping of terms is only meant to ease readability.

Term	Cause	Alignment to concepts in SOSA/SSN
oldssn:startTime	Unused	-
oldssn:endTime	Unused	-
oldssn:Device	Misused	$\sqsubseteq$ sosa:Platform $\sqcap$ ssn:System
oldssn:SensingDevice	Misused	$\sqsubseteq$ sosa:Platform $\sqcap$ sosa:Sensor
oldssn:Sensing	Unused	$\sqsubseteq$ sosa:Procedure
oldssn:DeploymentRelatedProcess	Unused	-
oldssn:deploymentProcessPart	Unused	-
oldssn:ofFeature	Unused	
oldssn:SensorDataSheet	Unused	
TOTAL: 9 deprecated terms		

Table 4

Terms in the old SSN that have no equivalent in the SOSA or SSN, and are simply deprecated. *Unused* terms were deprecated because no implementation evidence was found, and no requirement justified their inclusion in SSN. *Misused* terms were proven to be prone to misinterpretation and were deprecated in SSN.