

# POSO: A Generic Positioning System Ontology

Maxim Van de Wynckel ✉<sup>[0000-0003-0314-7107]</sup> and  
Beat Signer<sup>[0000-0001-9916-0837]</sup>

Web & Information Systems Engineering Lab  
Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium  
{mvdewync,bsigner}@vub.be

**Abstract.** While satellite-based positioning systems are mainly used in outdoor environments, various other positioning techniques exist for different domains and use cases, including indoor or underground settings. The representation of spatial data via semantic linked data is well addressed by existing spatial ontologies. However, there is a primary focus on location data with its specific geographical context, but a lack of solutions for describing the different types of data generated by a positioning system and the used sampling techniques to obtain the data. In this paper we introduce a new generic Positioning System Ontology (POSO) that is built on top of the Semantic Sensor Network (SSN) and Sensor, Observation, Sample, and Actuator (SOSA) ontologies. With POSO, we provide missing concepts needed for describing a positioning system and its output with known positioning algorithms and techniques in mind. Thereby, we enable the improvement of hybrid positioning systems making use of multiple platforms and sensors that are described via the presented POSO ontology.

**Keywords:** positioning system ontology · positioning techniques · positioning algorithms

## 1 Introduction

Whether we are developing a system for indoor or outdoor navigation or simply want to track the location of an object on a table, a positioning system that tracks the position based on one or multiple technologies and algorithms is needed. While outdoor positioning solutions mainly rely on satellite positioning systems such as the Global Positioning System (GPS), building-specific deployments and implementations using a wide variety of techniques [26,13] can be used indoors.

In order to facilitate the interoperability between different positioning systems or client applications, we need a vocabulary that is generic enough to cover various use cases. Expressing the position or movement in a geographical context is already well established using ontologies and vocabularies such as the Basic WGS84 vocabulary [3], the Location Ontology [10], GeoSPARQL [2] or the LinkedGeoData ontology [33]. However, positioning systems do not always need to operate within a geographical boundary and may even provide more

contextual information that is relevant for other positioning systems that would like to make use of the data.

Interoperability between multiple positioning systems also covers the fusion of the data these systems provide. Work on linked data networks for IoT sensors already exists [9,11], allowing raw sensor data to be accessible by multiple platforms. Decision-level fusion of positioning data remains a lacking capability of positioning systems due to the missing knowledge on how the location data has been processed or obtained. The additional semantic information from these systems is often not available to other systems, making the *handover* of tracking [14] between systems difficult.

In this paper we introduce *POSO*, a generic positioning system ontology for expressing the techniques, algorithms and data handled by a positioning system. We demonstrate how POSO can be used by a positioning system and that we can perform decision-level sensor fusion of positioning data between multiple independent positioning systems when data is semantically defined based on POSO.

## 2 Ontology Design

The main goal of our *Positioning System Ontology* (POSO) is to offer a solution that can model different positioning systems, their deployments, techniques, algorithms and the real-time data they are providing. However, semantics on post-processed trajectory data lies beyond the scope of POSO.

In Fig. 1 we provide a general overview of a positioning system and related components. A positioning system is deployed at a particular location or area that is meant to be covered. This can be a building, an area outdoors or even an object-specific location such as a game board that does not have to be related to any geographical boundaries. Each positioning system uses a set of algorithms and technologies to help compute a position. Finally, with positioning systems modelled based on POSO, we aim to track the position, orientation and other properties of one or more entities. These properties can be anything that is of relevance to the system and are obtained using the techniques implemented by the positioning system. Spatial properties of a tracked entity are located within the deployment using an optionally defined reference system.

We designed POSO with the Semantic Sensor Network Ontology (SSN) as a top-level ontology [22] together with the Sensor, Observation, Sample and Actuator (SOSA) ontology [23]. Combined, SOSA and SSN provide an ontology for linking sensors, actuators, observations, samplers and the systems needed to process this sensor data to an output. This provides a stable core ontology that could enable the modelling of a positioning system with its deployment, the used sensors, procedures, the entities and as well as the observable properties of those entities. However, as these ontologies are meant to be used as core ontologies, they do not offer any semantics for expressing the accuracy of individual observations, the different types of algorithms that are relevant for positioning or how the results should be represented in order to be interoperable.

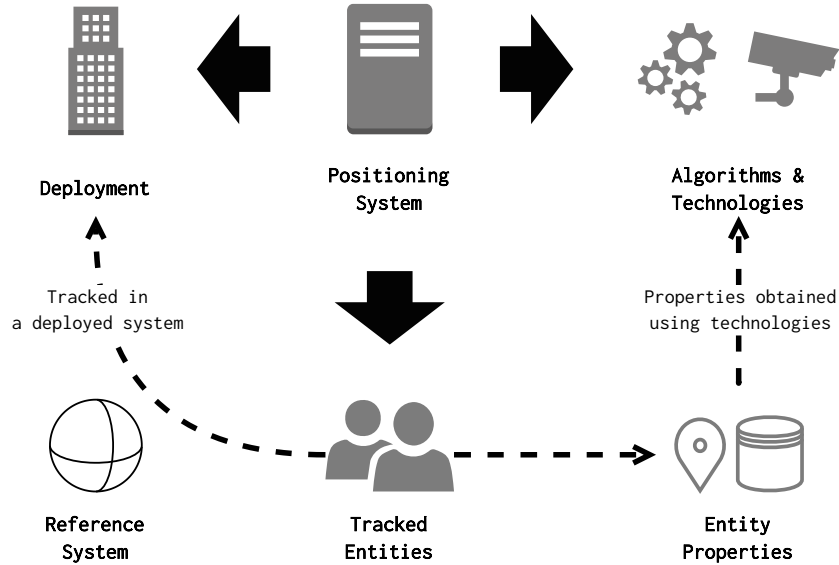


Fig. 1: Basic structure of a positioning system that tracks entities

Our ontology has been designed with the common data requirements of various positioning system technologies [26,13], datasets [37,38,41] and frameworks [42,12,30] in mind to cover all types of systems without overcomplicating the modelling of the data. An initial validation has further been conducted based on the OpenHPS framework [39].

With our proposed ontology, we aim to support concepts for defining a generic position, orientation, velocity, acceleration and the sampling of this data. We extended the `sosa:ObservableProperty` to express different types of position, orientation, velocity and acceleration. For expressing observation-based sensor data, we use the SOSA ontology together with the QUDT ontology for expressing Units of Measure, Quantity Kinds, Dimensions and Data Types [16]. Each observable property defined in POSO can also be used as a result within a SOSA observation, with a set of predicates that express the result. This enables expressing a fixed position of a feature of interest as shown later in Listing 6. The proposed vocabulary should support the following three main goals:

- **Sensor fusion:** High- and low-level sensor fusion should be possible based on the data [8]. High-level fusion, also called decision-level fusion, consists of merging processed data from multiple sources, while low-level fusion is the use of multiple sources of raw sensor data. Both fusion levels require additional knowledge on how the data has been obtained and its quality. In the context of high-level fusion in a positioning system, the additional semantics includes the accuracy as well as the techniques used to obtain the data. Using this knowledge, other systems can prioritise the observations to be used.

- **Historical data:** Positioning systems make use of previous information to predict future movement [44]. These predictions can be used to improve the calculation of a next position. In order to support this technique, historical positioning data should be available.
- **Granularity:** The position of an entity should be offered with varying ranges of granularity without causing conflicts with the decision-level sensor fusion. This enables use cases where observations of a minimum or maximum accuracy can be separated in a different triple store, further enabling access control to these individual stores.

## 2.1 Positioning System

A positioning system is a system or mechanism that can determine the position of one or multiple objects based on some sensor data. Multiple positioning systems might track the same object either individually or simultaneously. These multiple systems can work independently from each other or combine information from other systems to provide an output. We identify five types of positioning systems based on the ISO 19116:2019 standard [20]. Each positioning system extends the `ssn:System` class of the SSN ontology:

- **Satellite positioning system:** A positioning system using satellites. Examples include the Global Positioning System, Galileo or GLONASS [17].
- **Integrated positioning system:** An integrated or hybrid positioning system can be used outdoors, indoors or in any other space. Despite the fact that many positioning systems are hybrid (e.g. Assisted GPS [7]), we explicitly specify it as a type in POSO to define a system that does not fall within other more specific categories. In POSO, we define an integrated positioning system as a system that must implement *at least one* sensor fusion procedure.
- **Optical positioning system:** A positioning system that uses optical sensors to determine a position. This includes positioning systems where objects are tracked externally (e.g. Multi-Target Multi-Camera Tracking [24]) or systems where the tracked object is the optical sensor observing the environment (e.g. Visual Simultaneous Localisation and Mapping [34]).
- **Inertial positioning system:** An inertial positioning system calculates the position based on its movement and an initial reference point [15].
- **Indoor positioning system:** Indoor positioning covers all systems and techniques that are deployed indoors as opposed to outdoor positioning where often satellite positioning is used [26].

Being able to determine whether a position was obtained using satellites, an inertial- or indoor positioning system enables the reasoning about the relevance of a position sampled by one of these systems. With this additional knowledge, a fusion technique can ignore the sampled position of a satellite positioning system if an indoor positioning system is able to determine that the tracked object is inside a building. Alternatively, an inaccurate inertial positioning system may

provide useful context on the movement, rather than the position calculated using its algorithms.

Finally, we define a location-based service (LBS) as an `ssn:System` to categorise services with a black-box implementation of a positioning system. An example of such a service is the Geolocation API [29] that uses the techniques available by the underlying hardware. Note that an LBS might specify one or more positioning systems that it implements. In the `poso-common` extension discussed in Sect. 2.7, we provide a set of deployed satellite positioning systems, as these can be used as *subsystems*<sup>1</sup> in integrated positioning systems.

## 2.2 Positioning Algorithms and Techniques

The SOSA ontology describes a `sosa:Procedure` as a workflow, protocol, plan, algorithm or computational method to make an observation, sample or change the state of the world<sup>2</sup>. In a positioning system we identify a procedure as a workflow that processes sensor data to an intermediate result or observation.

A positioning system can use a broad range of techniques to calculate a position. While it might perform generic processing on raw sensor data, semantically describing the main techniques that are involved in the processing improves the reasoning that can be performed on the sampled data as well as its priority for decision-level sensor fusion. To illustrate this, we provide the example of an indoor positioning system (IPS) that uses simple QR codes for room check-ins and an IPS at the same location site that uses Bluetooth beacons. Without knowledge of the techniques used to determine a position, the accuracy of the position at a given time cannot be determined reliably. While the Bluetooth positioning provides a continuous output with varying accuracy, the QR scanning only provides a very high accuracy position when it is scanned; as the person will be near the code to scan it.

In POSO we subdivide a procedure over multiple different main categories that are based on the work of Liu et. al [26] and Gu et. al [13]:

- **Cell identification:** This covers all techniques that detect the position of an object when it is close to an object with a known position. Existing solutions range from radio frequency proximity to implicit position such as the act of scanning a QR code at a known fixed location.
- **Dead reckoning:** The velocity of an object can be used to determine its drift in space. This technique called dead reckoning can be a positioning system on its own, identified as an inertial positioning system [20], but can also form part of another technology such as Assisted GPS [7].
- **Fingerprinting:** Scene analysis techniques such as fingerprinting where sensor data is matched to a grid of positions can be used during the setup of the positioning system. Each scene analysis at a position is called a *fingerprint* and is used during the online tracking stage to determine a

<sup>1</sup> <https://www.w3.org/TR/vocab-ssn/#SSNhasSubSystem>

<sup>2</sup> <https://www.w3.org/TR/vocab-ssn/#SOSAProcedure>

position. The sensor data will be matched to the fingerprint that most closely resembles this data. POSO expresses a fingerprint as a subclass of `sosa:FeatureOfInterest` under the term `poso:Fingerprint` that requires to have a position in order to qualify as a fingerprint. This allows positioning systems that make use of this scene analysis to semantically describe the system’s setup.

- **Odometry:** Positioning techniques that use sensor data to detect the change in position are classified as odometry. This can be sensor data from motion sensors, visual observations or other environmental data such as magnetic interference [32].
- **Simultaneous localisation and mapping:** In simultaneous localisation and mapping (SLAM), a sensor determines features that are tracked during movement. By tracking these features it can determine the drift while simultaneously using the features to construct a map of the environment [36]. SLAM can be subdivided into Visual SLAM [34] when image sensors are used to track features as opposed to LiDAR sensors.
- **Triangulation:** Subdivided into angulation and lateration, triangulation covers positioning techniques that use angles or linear distance indicators to determine a position between two or more landmarks with a known position.
- **Sensor fusion:** In order to specify how multiple positioning systems or sensors are used together, a sensor fusion procedure category defines procedures where observations from multiple different (sub)systems are merged. This fusion technique can further make use of additional available context.

As an extension of POSO, the `POSO-common` module introduced later in Sect. 2.7 provides several commonly used positioning algorithms and techniques. The different positioning systems, techniques and observable properties along with their hierarchical relation to the SOSA and SSN ontologies are illustrated in Fig. 2. Properties that only contain fixed results without multiple observations are also subclasses of `sosa:Result` defining a single result as shown in Listing 6.

### 2.3 Absolute and Relative Positions

Multiple definitions exist to indicate where a spatial object is located. Our decision for using the term *position* was based on the definitions in the English language, as well as its uses within real-world applications:

- **Place/Area:** The place or area of an entity is an existing semantic definition in many vocabularies [1]. However, it implies a space rather than a particular point within this space.
- **Pose:** Often used in robotics [4] or when describing the movement of a person [27], a pose contains the position and orientation of an object. In real-world applications such as the Robotics Operation System (ROS) [30] it is meant to indicate a position and orientation within 3D space. Not every positioning system might operate within three dimensions, in which case the *pose* terminology might not be appropriate.

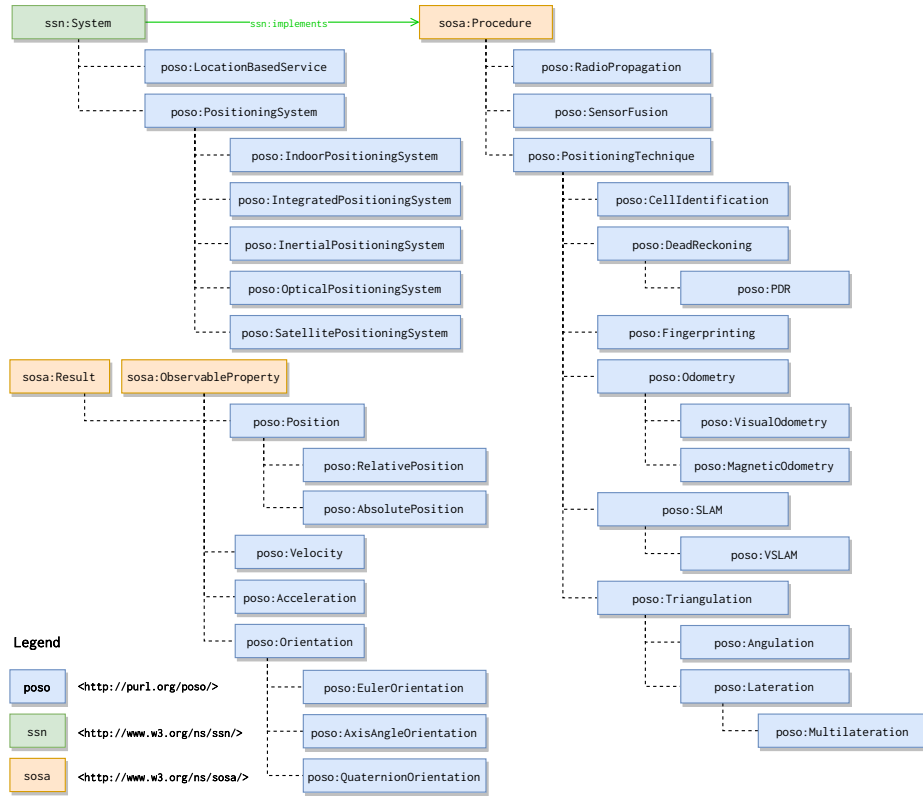


Fig. 2: Positioning systems and techniques in the POSO ontology

- **Location:** According to the Oxford English Dictionary which defines a location as “*a place where something happens or exists; the position of something*” we concluded that a location is a semantic description of either a vague place or accurate position. Because of this imprecision, we decided not to choose the term *location*.
- **Position:** A position can optionally also contain an orientation. It is the terminology used by most precise and generic positioning systems and location-based services [29].

Generic positioning systems make a distinction between absolute and relative positions [13,26]. An absolute position indicates a fixed point in space while a relative position is relative to another object or landmark. Such a relative position is a quantitative value relative in distance, angle or velocity, similar to the ‘Best Practice 9’ mentioned in [35].

When working with absolute positions in a geographical coordinate system, we make use of GeoSPARQL’s geographical position representation by the Open Geospatial Consortium (OGC). However, for absolute positions that should not be expressed as geometric coordinates, we use the QUDT ontology [16] to ex-

press Cartesian coordinates. POSO provides the concepts of `poso:xAxisValue`, `poso:yAxisValue` and `poso:zAxisValue` to express a `qudt:QuantityValue` in three dimensions.

Despite using simple Cartesian coordinates for a non-geographic position, a reference frame is still required to indicate how the Cartesian coordinates relate to each other. Similar to a reference frame in a geographical context, the reference frame allows the 2D or 3D position to be converted to other reference spaces such as a geographical context while still enabling the use of a positioning system that is only meant to operate in a specific context (i.e. an engineering reference frame as defined in ISO 19111 [21]). Defining a reference system is already well covered in GeoSPARQL [2]. In order to define the reference system of a `sosa:Result`, the `poso:hasSRS` or `poso:hasCRS` properties can be used.

For expressing a location that is covering a less specific larger 2D or 3D area, we still request the use of an absolute position, but provide the ability to indicate the accuracy as either a one-dimensional (i.e. distance) or polygonal coverage.

## 2.4 Orientation

An orientation is an important aspect of a positioning system. It does not only offer the final state of direction after a rotation of an object or person, but is also required by many positioning algorithms to determine a position. In a geographical context, the terminology *bearing*, *heading*, *course* or *azimuth* is used as a one-dimensional value [19]. However, as we aim to support use cases beyond geographical positioning and want to offer a more precise three-dimensional orientation, we resorted to mathematical concepts.

The commonly used mathematical definitions of an orientation are *Euler Angles*, *Axis Angles* and *Quaternions* [6]. Each mathematical definition has its advantages for a positioning system. Euler angles offer a well-known semantic description of a 3D rotation while still allowing the use of *yaw* only for expressing the heading in a 2D scenario. In robotics, quaternions are chosen since they avoid gimbal lock, as well as for their analytic properties.

As we aim to create a generic ontology, we have chosen to support any concept that can identify the orientation around three axes. POSO provides three extensions of the `poso:Orientation` class, including `poso:EulerOrientation`, `poso:AxisAngleOrientation` and `poso:QuaternionOrientation`.

## 2.5 Velocity and Acceleration

Active positioning systems make use of an object's velocity to determine a position and orientation based on its momentum. This procedure called dead reckoning uses an entity's last known location together with its angular and linear velocity to determine the new position and orientation at a later timestamp. POSO adds the concept of `poso:Velocity` with `poso:LinearVelocity` and `poso:AngularVelocity` as subclasses, as well as the momentary acceleration that is often returned by common Inertial Measurement Units (IMU).



## 2.6 Observations and Accuracy

Individual observations and different levels of granularity can be expressed for all properties. SSN-Systems [5], an extension of the SSN ontology, supports the description of a system’s properties, capabilities and conditions. While this enables the semantic description of the potential properties (i.e. accuracy, precision and operating environment) of a positioning system, it does not provide information on the individual observations. For a positioning system, the spatial accuracy can vary depending on the implemented procedure, the amount of sensor data as well as the accuracy of that data.

The accuracy of any observation can be expressed via `poso:hasAccuracy`, a subproperty of `ssns:qualityOfObservation`<sup>3</sup> that can be applied to an observation or individual result. Alternatively, for expressing the accuracy of spatial data (i.e. absolute or relative position) the `geosparql:hasSpatialAccuracy` from the GeoSPARQL 1.1 draft [28] can be used to express a QUDT quantity value. Further, in order to express the aimed accuracy of an observable property, the `ssns:Accuracy` class can be used to indicate that the accuracy applies to the position.

**Trajectories** Creating an observation for every calculated position provides context on historical data that can be used. Semantics of trajectories, such as segmentation, map matching and additional post-processing context [43] lies beyond the scope of our positioning system ontology. However, as each observation is a momentary timestamped result, they indirectly support the modelling of a trajectory space and time path [18].

A basic overview of how a person’s speed, orientation and position in an office deployment might be modelled is shown in Fig. 3. The green objects and properties represent the concepts from SOSA and SSN(S), the blue objects and properties represent the concepts from POSO and the purple objects represent the example individuals. Note that the full POSO specification with all the available concepts can be found in [40].

## 2.7 Alignment Module

The `poso-common` alignment module provides individual common positioning algorithms, systems and data used in positioning systems categorised under the classes defined in POSO. It describes seven satellite positioning systems [17]; known platforms such as IndoorAtlas<sup>4</sup>, Anyplace [12], OpenHPS [39,42], ROS [30] and individual algorithms for common positioning techniques. With the provided `poso-common` alignment module, we want to offer a foundation of algorithms and techniques that can easily be used to describe complete positioning systems. Future work should focus on expanding these algorithms, along with more detailed

<sup>3</sup> `ssns:` is the prefix for SSN-Systems [5]

<sup>4</sup> <https://www.indooratlas.com>

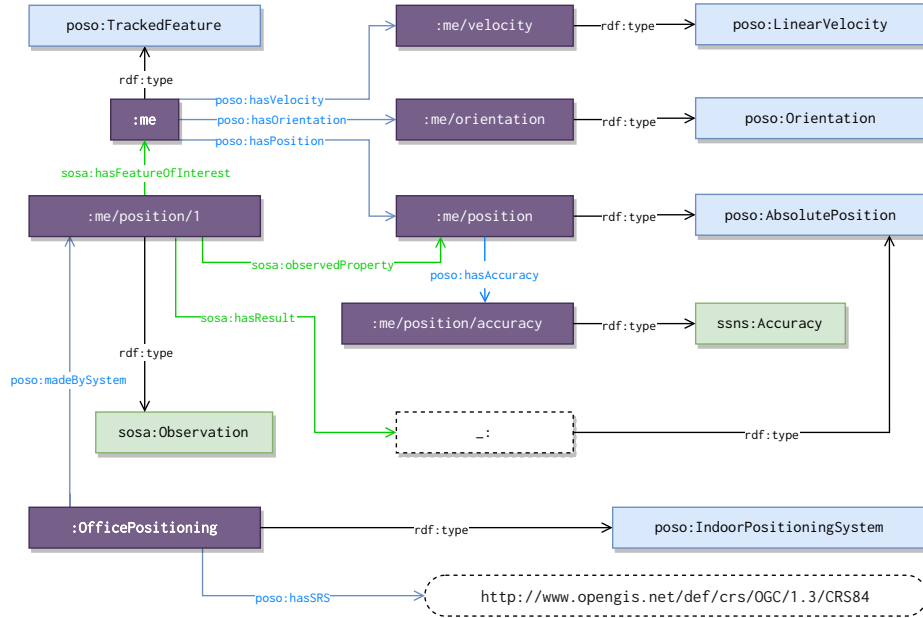


Fig. 3: Example of a positioning system with a position, orientation and velocity property

descriptions on their input and output shapes. In a hybrid or integrated positioning system as described in Sect. 2.1, the use of these common algorithms can provide insights on what observations to use in the fusion process.

### 3 Usage

In order to demonstrate the use of POSO to semantically model multiple positioning systems, we provide an example of a campus positioning system for the indoor as well as outdoor tracking of students. Our fictional setup consists of three individual systems; an outdoor positioning system using GPS, an indoor positioning system using Wi-Fi fingerprinting as introduced in Sect. 2.2 and a hybrid position system that makes use of the indoor and outdoor tracking subsystems by using a high-level sensor fusion technique.

We start by semantically describing the technical setup of the fictional deployment of the three positioning systems on our campus. Additional domain-specific ontologies such as IndoorGML [25] can be used to describe the physical context of these deployments. Throughout our examples, we make use of the prefixes defined in Listing 1.

In Listing 2 we create an outdoor campus positioning system that uses GPS. Indoors, we deploy a system that uses k-NN fingerprinting for Wi-Fi access points. For the integrated positioning system on lines 15 to 18 that uses both

```

1  @prefix poso: <http://purl.org/poso/> .
2  @prefix poso-common: <http://purl.org/poso/common/> .
3  @prefix ssn: <http://www.w3.org/ns/ssn/> .
4  @prefix sosa: <http://www.w3.org/ns/sosa/> .
5  @prefix dbr: <http://dbpedia.org/resource/> .
6  @prefix geosparql: <http://www.opengis.net/ont/geosparql#> .
7  @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
8  @prefix qudt: <http://qudt.org/schema/qudt/> .
9  @prefix unit: <http://qudt.org/vocab/unit/> .
10 @prefix ssns: <http://www.w3.org/ns/ssn/systems/> .
11 @prefix schema: <http://schema.org/> .
    
```

Listing 1: Prefixes used in the demonstration examples

the outdoor and indoor system, we add the two individual systems as subsystems with an additional procedure on how the high-level fusion of these two systems is performed.

```

1  dbr:Some_University a ssn:Deployment .
2  <deployment/building_a> a poso:IndoorDeployment, geosparql:Feature ;
3  rdfs:label "Building A"@en ;
4  geosparql:hasGeometry [
5  a geosparql:Geometry ;
6  geosparql:asWKT "...""^^geosparql:wktLiteral ] .
7  <system/OPS> a poso:LocationBasedService ;
8  rdfs:label "Outdoor campus positioning"@en ;
9  ssn:hasSubSystem poso-common:GPS ;
10 ssn:hasDeployment dbr:Some_University .
11 <system/IPS> a poso:IndoorPositioningSystem ;
12 rdfs:label "Indoor campus positioning"@en ;
13 ssn:hasDeployment <deployment/building_a> ;
14 ssn:implements poso-common:KNNFingerprinting .
15 <system/CampusPositioning> a poso:IntegratedPositioningSystem ;
16 rdfs:label "Hybrid campus positioning system"@en ;
17 ssn:hasSubSystem <system/OPS>, <system/IPS> ;
18 ssn:implements poso-common:WeightedAccuracyFusion .
    
```

Listing 2: Positioning system setup

The entity that is being tracked by the campus positioning system is configured in Listing 3. Each feature of interest, which we identify as our *tracked feature*, has multiple observable properties. A property predicate such as the `poso:hasPosition` on line 3 can be used multiple times to represent a position with different levels of granularity. In linked data front ends with data access control, such as Solid [31], these levels of granularity can control who is able to access a property with a certain accuracy. By specifying the accuracy of these

```

1 <me> a poso:TrackedFeature, foaf:Person ;
2   foaf:name "John Doe"@en ;
3   poso:hasPosition <me/position>, <me/approxposition> ;
4   poso:hasOrientation <me/orientation> .
5 <me/position> a poso:AbsolutePosition ;
6   rdfs:comment "Absolute position of John Doe"@en ;
7   poso:hasAccuracy <me/position/accuracy> .
8 <me/position/accuracy> a ssns:Accuracy ;
9   schema:maxValue "25.0"^^xsd:float ; schema:unitCode unit:CentiM .

```

Listing 3: Example setup of a tracked person and their properties

properties along with possible other semantic information, the information can be used in queries to determine which property offers the required accuracy.

Further, in Listing 4 we show an observation created by the outdoor positioning system. The GPS provides a latitude and longitude that we output using the OGC GeoSPARQL 1.1 ontology [28] as a well-known text (WKT) representation on lines 9 to 11.

```

1 <position/1654350300000> a sosa:Observation ;
2   sosa:hasFeatureOfInterest <me> ;
3   sosa:observedProperty <me/position> ;
4   sosa:resultTime "2022-06-04T15:55:00+02:00"^^xsd:dateTimeStamp ;
5   poso:usedSystem <system/OPS> ;
6   sosa:hasResult [ a geosparql:Geometry ;
7     geosparql:hasSpatialAccuracy [ a qudt:QuantityValue ;
8       qudt:unit unit:CentiM ; qudt:numericValue "28"^^xsd:float ] ;
9     geosparql:asWKT ""
10      <http://www.opengis.net/def/crs/OGC/1.3/CRS84>
11      Point(4.888028 50.31397)""^^geosparql:wktLiteral ;
12     geosparql:dimension 2 ] .

```

Listing 4: Example observation of the outdoor positioning system

Indoors, our system outputs an absolute Cartesian 3D position as illustrated in Listing 5. We identify that the 3D position is made inside a specific deployment on line 8, which contains information about its geometry and the reference system used to convert the coordinates to a common reference frame used by the campus positioning system. The technique used to obtain the result is defined using `sosa:usedProcedure` while the system where this technique is used is defined based on `poso:usedSystem`.

In previous example listings, we have shown how a positioning system might model the observations of an absolute position. With the example in Listing 6 we outline how a relative distance to a wireless access point (named `wap_1`) from our `TrackedFeature` can be expressed. Similar to an absolute position, we can have multiple observations of the relative distance. POSO requires the

`poso:isRelativeTo` predicate on a relative position to indicate the feature of interest that the position is relative to.

```

1 <position/1647513000000> a sosa:Observation ;
2   sosa:hasFeatureOfInterest <me> ;
3   sosa:observedProperty <me/position> ;
4   sosa:resultTime "2022-03-17T11:30:00+01:00"^^xsd:dateTimeStamp ;
5   sosa:usedProcedure poso-common:KNNFingerprinting ;
6   poso:usedSystem <system/IPS> ;
7   sosa:hasResult [ a poso:AbsolutePosition ;
8     poso:inDeployment <deployment/building_a> ;
9     poso:hasAccuracy [ a ssns:Accuracy ;
10      schema:maxValue "25.0"^^xsd:float ;
11      schema:unitCode unit:CentiM ] ;
12    poso:xAxisValue [ a qudt:QuantityValue ;
13      qudt:unit unit:M ; qudt:numericValue "5"^^xsd:double ] ;
14    poso:yAxisValue [ a qudt:QuantityValue ;
15      qudt:unit unit:M ; qudt:numericValue "6"^^xsd:double ] ;
16    poso:zAxisValue [ a qudt:QuantityValue ;
17      qudt:unit unit:M ; qudt:numericValue "3.5"^^xsd:double ] ] .

```

Listing 5: Example observation of the indoor positioning system

```

1 <landmark/wap_1> a poso:Landmark ;
2   rdfs:label "Wireless Access Point 1"@en ;
3   poso:hasPosition [ a poso:AbsolutePosition ;
4     poso:hasAccuracy [ ... ] ;
5     poso:xAxisValue [ ... ] ;
6     poso:yAxisValue [ ... ] ;
7     poso:zAxisValue [ ... ] ] .
8 <me/position/relative/wap_1> a poso:RelativeDistance ;
9   ssn:isPropertyOf <me> ; # Relative distance from <me> ...
10  poso:isRelativeTo <landmark/wap_1> ; # to <landmark/wap_1>
11  rdfs:comment "Relative position of John Doe to WAP_1"@en .
12 <position/relative/wap_1/1646891100000> a sosa:Observation ;
13   sosa:hasFeatureOfInterest <me>, <landmark/wap_1> ;
14   sosa:observedProperty <me/position/relative/wap_1> ;
15   sosa:resultTime "2022-03-10T06:45:00+01:00"^^xsd:dateTimeStamp ;
16   poso:madeBySystem <system/IPS> ;
17   sosa:usedProcedure poso-common:LDPL ; # Log-distance path loss
18   sosa:hasResult [ a qudt:QuantityValue ;
19     qudt:unit unit:Meter ; qudt:value "3.7"^^xsd:double ] ;
20   sosa:hasResult [ a qudt:QuantityValue ;
21     qudt:unit unit:DeciB_M ; qudt:value "-82"^^xsd:integer ] .

```

Listing 6: Example observation of a relative position

As mentioned in the beginning of Sect. 2, each observable property can also be used to express a fixed result that does not consist of multiple observations. On lines 1 to 7 of Listing 6 we utilise this ability to express a fixed result to define the fixed position of a landmark rather than creating a single observation where the position is defined as a result. On lines 12 to 21, we have one observation of this observable relative distance obtained using our indoor positioning system. The result is expressed as a distance using a path loss algorithm and the raw signal strength expressed in decibel-milliwatts (*dBm*).

In order to provide a single output for the campus positioning system, we can use the observations from the indoor and outdoor positioning systems shown in Listing 5 and Listing 4 to compute a fused output based on the weighted accuracy fusion procedure that our campus positioning system implements in Listing 2. Using the knowledge about the accuracy, the systems that produced the results and the indoor positioning system deployments, we can perform a fusion with more context than only the self-reported accuracy of each individual subsystem.

## 4 Conclusions and Future Work

In this paper we introduced our new generic positioning system ontology called POSO for describing concepts relevant to a positioning system. These concepts include the different observable properties that can be obtained by a positioning system, the different categories of systems and the different algorithms and techniques these systems can implement to handle positioning. Further note that our generic positioning system ontology does not only focus on common geospatial and geographical concepts that are already described in various existing vocabularies [10,2,33] but also offers a novel vocabulary for describing generic data outputted by a positioning system. We expanded the SSN [22] and SOSA [23] ontologies by providing common procedures and observable properties. By further presenting the `poso-common` module, we illustrated how POSO can be expanded with a set of common algorithms, existing systems and platforms.

Finally, we illustrated the usage of POSO with a scenario containing two positioning systems and a hybrid positioning system using a high-level fusion technique. In this demonstration, we have shown how each positioning system might be modelled using POSO and how observational data can be expressed.

Future work will focus on adding additional positioning technique and algorithm procedures, further describing the input and output that each procedure provides. By using known input and output RDF shapes that are used in different positioning systems, we can further classify a positioning system's technologies and the output they provide. While we already offer procedures obtaining map information (i.e. Simultaneous Localisation and Mapping), we did not showcase how the raw observations generated by such an algorithm can be created.

*Supplemental Material Statement:* All the sources of POSO and `poso-common`, along with additional documentation<sup>5</sup> is available on GitHub [40].

<sup>5</sup> <https://openhps.github.io/POSO/1.0/en/>

## References

1. Abdelmoty, A.I., Smart, P., Jones, C.B.: Building Place Ontologies for the Semantic Web: Issues and Approaches. In: Proceedings GIR 2007, 4th ACM Workshop on Geographical Information Retrieval. pp. 7–12 (2007)
2. Battle, R., Kolas, D.: GeoSPARQL: Enabling a Geospatial Semantic Web. *Semantic Web Journal* **3**(4) (2011)
3. Brickley, D.: W3C Semantic Web Interest Group: Basic Geo (WGS84 lat/long) Vocabulary. W3C Working Group Note (2004)
4. Carbonera, J.L., Fiorini, S.R., Prestes, E., Jorge, V.A., Abel, M., Madhavan, R., Locoro, A., Gonçalves, P., Haidegger, T., Barreto, M.E., et al.: Defining Positioning in a Core Ontology for Robotics. In: Proceedings of IROS 2013, IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 1867–1872 (2013). <https://doi.org/10.1109/IROS.2013.6696603>
5. Compton, M., Barnaghi, P., Bermudez, L., Garcia-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., et al.: The SSN Ontology of the W3C Semantic Sensor Network Incubator Group. *Journal of Web Semantics* **17** (December 2012). <https://doi.org/10.1016/j.websem.2012.05.003>
6. Diebel, J.: Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors. *Matrix* **58**(15-16), 1–35 (2006)
7. Djuknic, G.M., Richton, R.E.: Geolocation and Assisted GPS. *Computer* **34**(2) (2001). <https://doi.org/10.1109/2.901174>
8. Elmenreich, W.: An Introduction to Sensor Fusion. Tech. Rep. 47/2001, Vienna University of Technology (November 2002)
9. Elsaleh, T., Bermudez-Edo, M., Enshaeifar, S., Acton, S.T., Rezvani, R., Barnaghi, P.: IoT-Stream: A Lightweight Ontology for Internet of Things Data Streams. In: Proceedings of GIoTS 2019, Global IoT Summit. pp. 1–6 (2019). <https://doi.org/10.1109/GIoTTS.2019.8766367>
10. Flury, T., Privat, G., Ramparany, F.: OWL-based Location Ontology for Context-Aware Services. Proceedings of AIMS 2004, International Workshop on Artificial Intelligence in Mobile Systems pp. 52–57 (2004)
11. Ganzha, M., Paprzycki, M., Pawłowski, W., Szmaja, P., Wasielewska, K.: Semantic Interoperability in the Internet of Things: An Overview From the INTER-IoT Perspective. *Journal of Network and Computer Applications* **81**, 111–124 (2017). <https://doi.org/10.1016/j.jnca.2016.08.007>
12. Georgiou, K., Constambeys, T., Laoudias, C., Petrou, L., Chatzimilioudis, G., Zeinalipour-Yazti, D.: Anyplace: A Crowdsourced Indoor Information Service. In: Proceedings of MDM 2015, International Conference on Mobile Data Management. pp. 291–294 (2015). <https://doi.org/10.1109/MDM.2015.80>
13. Gu, Y., Lo, A., Niemegeers, I.: A Survey of Indoor Positioning Systems for Wireless Personal Networks. *IEEE Communications Surveys & Tutorials* **11**(1) (2009). <https://doi.org/10.1109/SURV.2009.090103>
14. Hansen, R., Wind, R., Jensen, C.S., Thomsen, B.: Seamless Indoor/Outdoor Positioning Handover for Location-based Services in Streamspin. In: Proceedings of MDM 2009, International Conference on Mobile Data Management. pp. 267–272 (2009). <https://doi.org/10.1109/MDM.2009.39>
15. Harle, R.: A Survey of Indoor Inertial Positioning Systems for Pedestrians. *IEEE Communications Surveys & Tutorials* **15**(3), 1281–1293 (2013). <https://doi.org/10.1109/SURV.2012.121912.00075>

16. Hodgson, R., Keller, P.J., Hodges, J., Spivak, J.: QUDT-Quantities, Units, Dimensions and Data Types Ontologies. <http://qudt.org> (2014)
17. Hofmann-Wellenhof, B., Lichtenegger, H., Wasle, E.: GNSS—Global Navigation Satellite Systems: GPS, GLONASS, Galileo, and more. Springer Science & Business Media (2007)
18. Hu, Y., Janowicz, K., Carral, D., Scheider, S., Kuhn, W., Berg-Cross, G., Hitzler, P., Dean, M., Kolas, D.: A Geo-ontology Design Pattern for Semantic Trajectories. In: Proceedings of COSIT 2013, International Conference on Spatial Information Theory. pp. 438–456 (2013). [https://doi.org/10.1007/978-3-319-01790-7\\_24](https://doi.org/10.1007/978-3-319-01790-7_24)
19. ISO Central Secretary: Information Technology – Database Languages – SQL Multimedia and Application Packages – Part 3: Spatial. Standard ISO/IEC 13249-3:2016, International Organization for Standardization, Geneva, Switzerland (2016), <https://www.iso.org/standard/60343.html>
20. ISO Central Secretary: Geographic information – Positioning Services. Standard ISO 19116:2019, International Organization for Standardization (2019), <https://www.iso.org/standard/70882.html>
21. ISO Central Secretary: Geographic Information – Referencing by Coordinates. Standard ISO 19111:2019, International Organization for Standardization (2019), <https://www.iso.org/standard/74039.html>
22. Janowicz, K., Compton, M.: The Stimulus-Sensor-Observation Ontology Design Pattern and its Integration Into the Semantic Sensor Network Ontology. In: Proceedings of SSN 2010, International Workshop on Semantic Sensor Networks (2010)
23. Janowicz, K., Haller, A., Cox, S.J., Le Phuoc, D., Lefrançois, M.: SOSA: A Lightweight Ontology for Sensors, Observations, Samples, and Actuators. *Journal of Web Semantics* **56** (May 2019). <https://doi.org/10.1016/j.websem.2018.06.003>
24. Krumm, J., Harris, S., Meyers, B., Brumitt, B., Hale, M., Shafer, S.: Multi-Camera Multi-Person Tracking for EasyLiving. In: Proceedings of VS 2000, International Workshop on Visual Surveillance. pp. 3–10 (2000)
25. Lee, K., Lee, J., Kwan, M.P.: Location-based Service Using Ontology-based Semantic Queries: A Study With a Focus on Indoor Activities in a University Context. *Computers, Environment and Urban Systems* **62** (March 2017). <https://doi.org/10.1016/j.compenvurbsys.2016.10.009>
26. Liu, H., Darabi, H., Banerjee, P., Liu, J.: Survey of Wireless Indoor Positioning Techniques and Systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **37**(6), 1067–1080 (2007). <https://doi.org/10.1109/TSMCC.2007.905750>
27. Perera, M., Haller, A., Rodríguez Méndez, S.J., Adcock, M.: HDGI: A Human Device Gesture Interaction Ontology for the Internet of Things. In: Proceedings of ISWC 2020, International Semantic Web Conference. pp. 111–126 (2020). [https://doi.org/10.1007/978-3-030-62466-8\\_8](https://doi.org/10.1007/978-3-030-62466-8_8)
28. Perry, M., Herring, J., Car, N.J., Homburg, T., J.D. Cox, S., Bonduel, M., Knibbe, F.: OGC GeoSPARQL: A Geographic Query Language for RDF Data: GeoSPARQL 1.1 Draft. Tech. rep. (2021), <https://opengeospatial.github.io/ogc-geosparql/geosparql11/spec.html>
29. Popescu, A.: Geolocation API Specification 2nd Edition. W3C Recommendation, W3C (2016), <https://www.w3.org/TR/2016/REC-geolocation-API-20161108/>
30. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: ROS: An Open-Source Robot Operating System. In: Proceedings of the International Workshop on Open Source Software (2009)



31. Sambra, A.V., Mansour, E., Hawke, S., Zereba, M., Greco, N., Ghanem, A., Zagidulin, D., Aboulnaga, A., Berners-Lee, T.: Solid: A Platform for Decentralized Social Applications Based on Linked Data. Tech. rep., MIT CSAIL & Qatar Computing Research Institute (2016)
32. Skog, I., Hendeby, G., Gustafsson, F.: Magnetic Odometry: A Model-Based Approach Using a Sensor Array. In: Proceedings of FUSION 2018, International Conference on Information Fusion. pp. 794–798 (2018). <https://doi.org/10.23919/ICIF.2018.8455430>
33. Stadler, C., Lehmann, J., Höffner, K., Auer, S.: LinkedGeoData: A Core for a Web of Spatial Open Data. *Semantic Web* **3**(4), 333–354 (2012). <https://doi.org/10.3233/SW-2011-0052>
34. Sumikura, S., Shibuya, M., Sakurada, K.: OpenVSLAM: A Versatile Visual SLAM Framework. In: Proceedings of MM 2019, International Conference on Multimedia. Nice, France (October 2019). <https://doi.org/10.1145/3343031.3350539>
35. Tandy, J., van den Brink, L., Barnaghi, P.: Spatial Data on the Web Best Practices. W3C Working Group Note (2017), <https://www.w3.org/TR/sdw-bp/>
36. Thrun, S.: Simultaneous Localization and Mapping. In: Robotics and Cognitive Approaches to Spatial Mapping, pp. 13–41. Springer (2007). [https://doi.org/10.1007/978-3-540-75388-9\\_3](https://doi.org/10.1007/978-3-540-75388-9_3)
37. Torres-Sospedra, J., Montoliu, R., Martínez-Usó, A., Avariento, J.P., Arnau, T.J., Benedito-Bordonau, M., Huerta, J.: UJIIndoorLoc: A New Multi-Building and Multi-Floor Database for WLAN Fingerprint-based Indoor Localization Problems. In: Proceedings of IPIN 2014, International Conference on Indoor Positioning and Indoor Navigation. pp. 261–270 (2014). <https://doi.org/10.1109/IPIN.2014.7275492>
38. Torres-Sospedra, J., Rambla, D., Montoliu, R., Belmonte, O., Huerta, J.: UJIIndoorLoc-Mag: A New Database for Magnetic Field-based Localization Problems. In: Proceedings of IPIN 2015, International Conference on Indoor Positioning and Indoor Navigation. pp. 1–10 (2015). <https://doi.org/10.1109/IPIN.2015.7346763>
39. Van de Wynckel, M., Signer, B.: OpenHPS: An Open Source Hybrid Positioning System. Tech. Rep. WISE-2020-01, Vrije Universiteit Brussel (2020). <https://doi.org/10.48550/ARXIV.2101.05198>
40. Van de Wynckel, M., Signer, B.: POSO: A Generic Positioning System Ontology Repository. <https://github.com/OpenHPS/POSO/> (2022)
41. Wang, S., Yue, J., Dong, Y., He, S., Wang, H., Ning, S.: A Synthetic Dataset for Visual SLAM Evaluation. *Robotics and Autonomous Systems* **124**, 103336 (2020). <https://doi.org/10.1016/j.robot.2019.103336>
42. Van de Wynckel, M., Signer, B.: Indoor Positioning Using the OpenHPS Framework. In: Proceedings of IPIN 2021, International Conference on Indoor Positioning and Indoor Navigation. pp. 1–8 (2021). <https://doi.org/10.1109/IPIN51156.2021.9662569>
43. Yan, Z., Macedo, J., Parent, C., Spaccapietra, S.: Trajectory Ontologies and Queries. *Transactions in GIS* **12**, 75–91 (2008). <https://doi.org/10.1111/j.1467-9671.2008.01137.x>
44. Zampella, F., Ruiz, A.R.J., Granja, F.S.: Indoor Positioning Using Efficient Map Matching, RSS Measurements, and an Improved Motion Model. *IEEE Transactions on Vehicular Technology* **64**(4) (2015). <https://doi.org/10.1109/TVT.2015.2391296>