HSSN: an ontology for hybrid semantic sensor networks

Elio Mansour, Richard Chbeir, Philippe Arnould

HSSN: An Ontology for Hybrid Semantic Sensor Networks

Elio Mansour
Univ Pau & Pays Adour
E2S UPPA, LIUPPA
Mont-de-Marsan, 40000, France
elio.mansour@univ-pau.fr

Richard Chbeir
Univ Pau & Pays Adour
E2S UPPA, LIUPPA
Anglet, 64600, France
richard.chbeir@univ-pau.fr

Philippe Arnould
Univ Pau & Pays Adour
E2S UPPA, LIUPPA
Mont-de-Marsan, 40000, France
philippe.arnould@univ-pau.fr

ABSTRACT

Semantic web techniques (e.g., ontologies) have been recently adopted for sensor network modeling. However, existing works do not fully address these challenges: (i) representing different sensor types (e.g., mobile/static sensors) to enrich the network with different data and ensure better coverage; (ii) representing a variety of platforms (e.g., environments, devices) for sensor deployment, thus, integrating new components (e.g., mobile phones); (iii) representing the diverse data (scalar/multimedia) needed for various applications (e.g., event detection); and (iv) proposing a generic model to allow re-usability in various application domains. In this paper, we propose HSSN, an ontology that extends the Semantic Sensor Network (SSN) ontology which is already re-usable and considers various platforms. We extend the representation of sensors, sensed data, and deployment environments to cope with these challenges. We evaluate the consistency, accuracy, clarity, and performance of HSSN.

KEYWORDS

Semantic Sensor Networks, Ontology, Sensor Mobility

1 INTRODUCTION

Recently, Sensor Networks (SNs) have impacted more and more application domains [14] such as environmental sensing, military, and medical fields. Various sensors (e.g., camera, microphone) are nowadays embedded in smart phones, and capable of sensing useful data for various purposes (e.g., pollution monitoring in a city). Therefore, considering such devices, and other equipment capable of sensing, is very beneficial for knowledge extraction in sensor networks. Nonetheless, SNs may produce heterogeneous data, that have to be collected, processed, and analyzed in order to provide various services for network managers. Representing, sharing, and integrating the aforementioned data is a challenging task. In order to address this challenge, semantic web techniques, such as ontologies, have been adopted for their information representation. However, existing approaches on sensor network representation [1–4, 6, 11, 13] are restrictive due to the following issues:

- *Lack of platform diversity:* existing approaches do not consider equipment with embedded sensors (e.g., smart phones, drones, machines) as platforms, in addition to traditional platforms (e.g., buildings, cities, offices) where sensors are deployed. Extending the platform representation, by both considering and detailing the representation of various types of platforms, allows the addition of new components to the network, nested platforms, and dynamic, collaborative sensing activities (e.g., crowd-sensing).
- *Lack of sensor diversity:* these works do not represent different sensor types (e.g., mobile/static sensors, simple sensor nodes/multi-sensor devices, sensors capable of sensing scalar/multimedia properties). Providing a more detailed sensor representation that considers various attributes (e.g., mobility) improves network coverage, and allows sensor tracking and dynamic sensing.
- *Lack of data diversity:* most works cover scalar environment properties (i.e., mainly focus on scalar data such as temperature, motion, and neglecting multimedia data such as sounds, images, and videos). Since several devices are capable of sensing both types, and data diversity is required for different application purposes (e.g., event detection), it is important to cover scalar and multimedia data in the representation.
- *Lack of re-usability:* these approaches are heavily linked to a specific application domain. The sensor network modeling should remain generic and re-usable in different contexts.

To answer these challenges, we present here an extension of the widely used Semantic Sensor Network ontology (SOSA/SSN) [7] called HSSN. It allows the representation of hybrid sensor networks, i.e., networks containing mobile/static sensors, scalar/multimedia properties, and infrastructures/devices as platforms where sensors are deployed. We chose to extend SSN since it is already re-usable in various contexts and allows the representation of different platforms. Nonetheless, sensor and data diversity are not fully developed. Our proposal adds diverse data, sensors, and details the description of various platform types. In addition, HSSN does not contain domain specific knowledge and can be easily aligned with other ontologies (e.g., mobile phone, smart building ontologies [16]).

The rest of the paper is organized as follows. Section 2 illustrates a scenario that motivates our proposal. Section 3 reviews related work regarding mobility, platforms, and sensed data. Section 4 details the HSSN ontology. Section 5 describes the experimental setup and results. Finally, section 6 concludes the paper and discusses future research directions.

2 MOTIVATING SCENARIO

To highlight the utility of our proposal, we choose the following scenario (we only use this example to concretely illustrate the needs, challenges, and motivations behind our work. We do not consider it to be a generic, all summarizing, sensor network application scenario). Consider a smart mall/shopping center (cf. Fig.1). In order to optimize client comfort, health, and security, the smart mall relies on a set of sensors (s_1-s_9) to monitor the environment. Video surveillance cameras (s_1-s_6) monitor security related events. Humidity, CO_2, and temperature sensors (s_7, s_8, and s_9 respectively) make observations that help regulate the indoor air quality, and temperature. The sensed data is stored and used for these applications. However, many improvements still need to be integrated:

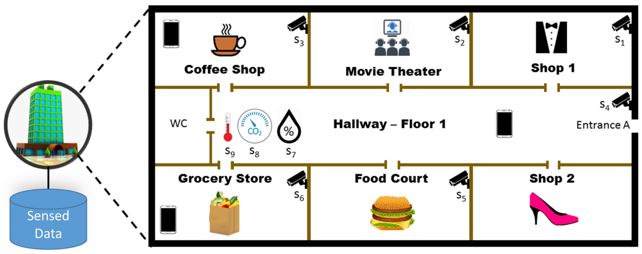

Figure 1: Smart Mall Example

- Need 1- Provide better temperature/air quality readings: relying on measures from a multitude of sensors (instead of only one) allows a more precise monitoring of the environment. Currently, this is not possible since there is only one temperature/air quality sensor in the mall.
- Need 2- Keep track of client positions in the mall since it is useful to know: the number of occupants in each zone, client positions for tracking suspicious/interesting behaviours. Cameras (s_1-s_6) are used by mall agents to monitor limited events and cannot track client locations everywhere.
- Need 3- Cover all areas of the mall: this is critical for client security and safety. In the current setup, many uncovered areas exist (e.g., no temperature monitoring in the movie theater, no video surveillance in Shop 2).
- Need 4- Provide a rich documentation of critical events: in order to increase the understanding of events (e.g., when reporting incidents, providing evidences), rich descriptions should be provided to police with a variety of sensed multimedia and scalar data (e.g., video, audio, image, temperature, humidity). Currently, reports on attack incidents (e.g., gunshot) rely only on video surveillance footage (e.g., no noise levels to confirm the gunshot, no motion data to describe how people ran away). A bigger data variety is needed.
- Need 5- Adapt to changing event detection needs: sometimes new/spontaneous events need to be detected, the mall should be able to sense the required data and detect these events. However, the current sensor configuration/deployment and sensed data cannot be easily modified. This doesn't allow the detection of new events.

In order to address these issues, the mall managers would need to add more sensors to cover all zones. This ensures full coverage of the mall (Need 3), and allows multiple observations from each zone for aggregation (Need 1). In addition, they could replace the cameras with more advanced ones that enable image processing for tracking purposes (Need 2). However, this increases the equipment, maintenance, and implementation costs without addressing Needs 4 and 5. A more appropriate solution would be to integrate visitors' mobile phones (since they embed sensors) as mobile sensors in the mall's network, while avoiding excessive resource consumption from the devices (e.g., draining a phone's battery). This provides the following benefits: (i) sensor mobility provides observations from different areas of the mall, multiple sensors can therefore collaborate to calculate more reliable air quality/temperature measures (Need 1); (ii) mall visitors can easily be tracked using their connected mobile phones (Need 2), location information can also be used to discover uncovered areas (Need 3); (iii) using various sensors from different devices helps cover a wider array of scalar/multimedia properties (Need 4); and (iv) these devices provide a diversity of hardware (e.g., sensors), software, and services that can be adapted to changing event detection needs (Need 5). However, when adding mobility, diverse data, and devices to the network, the following challenges emerge:

- Challenge 1: How to expressively describe locations in the mall?
- Challenge 2: How to consider ad-hoc devices in the network? How to query them based on their capabilities (e.g., without draining their batteries)? How to represent the services that they provide?
- Challenge 3: How to track locations and coverage areas of mobile sensors?
- Challenge 4: How to collect scalar/multimedia observations from sensors?

Other challenges also exist when modeling sensor networks. However, we address here the aforementioned four challenges from a data modeling perspective by proposing an extension of the semantic sensor network ontology that includes mobility, platform, and data related concepts.

3 RELATED WORK

In this section, we study existing sensor network ontologies. We focus our review on sensor mobility, deployment platforms, and semantic representation of multimedia data. We compare these works based on the following criteria:

1. *Sensor diversity:* Indicating if different types of sensors exist in the sensor network (e.g., mobile/static sensors, simple nodes/multi-sensor equipment, sensors capable of sensing scalar/multimedia properties).
2. *Platform diversity:* Stating if the approach allows and details the description of different platforms where sensors are deployed (e.g., in infrastructures, on devices).
3. *Data diversity:* Denoting the approach's ability to handle various data/properties (e.g., scalar, multimedia).
4. *Re-usability:* Indicating if the approach is re-usable in various contexts.

3.1 Sensor Diversity

In [2], the authors focus mainly on features that describe the sensor nodes, their functionality, and their current CPU, memory, and power supply states (in order to determine the future state of the WSN). However, they do not represent different types of sensors. In [6], the authors provide a set of ontologies describing missions, tasks, sensors, and deployment platforms for sensor to task assignment. Unfortunately, different types of sensors were not considered. In [7], the authors propose the SOSA/SSN[1] ontologies. Together, they describe systems of sensors and actuators, observations, the

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

used procedures, properties, and so forth. SOSA/SSN propose simple sensor node representation, as well as (sensing) systems/devices. However, SOSA/SSN do not propose any mobility-related concepts, nor multimedia data/properties. The authors only consider one aspect of sensor diversity (i.e., simple sensor nodes/sensor systems). In [1], the authors propose an extension of SSN, denoted MSSN (Multimedia SSN), where they detail the technical aspects of multimedia data (e.g., video, audio segments, frequencies). In this work, the authors improve the sensor diversity of SOSA/SSN by adding a media sensor (i.e., a sensor type that observes multimedia properties). However, they do not achieve full sensor diversity as they do not consider sensor mobility (i.e., mobile/static sensors).

3.2 Platform Diversity

The authors in [9] only consider embedded sensors on mobile phones to monitor noise pollution. In [4], the authors rely on traditional deployment of sensor nodes in the wilderness to detect fire events. The problem is, these works do not provide any platform diversity. In the SSN ontology [7], sensors are deployed on platforms. SSN also introduces systems, that can integrate various sensors, actuators, and samplers. Therefore, SSN provides a foundation for sensor deployment on various platforms (e.g., traditional deployment on platforms, embedding sensors in systems and devices). However, the differences between theses platforms is not detailed in SSN. The description of physical infrastructures/environments such as smart buildings and cities (where it would be interesting to model maps and locations) is different than of machines, drones, and devices that host sensors (where it would be interesting to model hardware and software). It is better to distinguish and detail the description of different platform types to better understand the environments where sensors are deployed (e.g., for location-based services in infrastructures, task assignment based on hardware/-software capabilities for devices). MSSN [1] suffers from the same limitation since it is based on the SSN ontology and does not add any new concepts related to platforms.

3.3 Data Diversity

In [5], the authors represent images for object recognition purposes. The scope of their work does not extend to other types of multimedia data (e.g., video, audio). In [10], the authors are also limited to image representation, since they propose an approach for object-based image retrieval. In [11], the authors monitor noise pollution in urban zones by sensing (audio) noise levels using occupants' mobile phones. The authors only consider noise data, and geo-locations in order to generate a noise level map. Therefore, their proposal does not fully consider data diversity (e.g., video, images, other scalar data). The SSN ontology [7] does not consider multimedia observations. It details scalar sensed data. This motivated the proposal of MSSN [1] where the authors represent multimedia data in sensor networks. For each multimedia observation value, the authors associate data descriptors (denoted media descriptors), and data segments (denoted media segments). Their proposed ontology, MSSN, complements the SSN ontology [7] since the latter does not cover multimedia contents nor multimedia sensors.

3.4 Re-usability

In [9], the authors propose a noise pollution monitoring solution in a city using mobile phones to sense noise. The authors enrich the sensed information by allowing users to add contextual information to their sensor observations. However, it lacks the genericity needed for it to be reusable in other contexts. In [1], the authors propose a multimedia wireless sensor network ontology for event detection purposes (the authors include concepts related to atomic, complex events, and event detection/composition). These added concepts are domain specific and not necessary in other application scenarios. This restricts MSSN's re-usability. Each of these works are task-centric and heavily linked to an application purpose. The SSN ontology [7] remains generic and re-usable in various contexts since it is extensible and does not contain any concepts that link it to any specific application.

3.5 Discussion

The aforementioned works do not fully integrate sensor diversity in their representation of sensor networks (i.e., static/mobile sensors, simple node/multi-sensor devices, and scalar/multimedia sensors). The SSN ontology [7] is a culmination of much of the related work on semantic sensor networks and is the most widely used (re-usable). In addition, SSN is extensible, facilitates alignments with other standards, and allows the integration of new concepts. The MSSN ontology [1], integrates multimedia data in SSN. Therefore, we propose to extend SSN since: (i) it partially allows sensor diversity; (ii) it is re-usable and does not contain any domain specific knowledge; and (iii) it allows having various platform types. Moreover, we do not neglect MSSN for its ability to cover multimedia data (data diversity). Therefore, our proposal will extend SSN and use key MSSN concepts in order to achieve full sensor diversity, platform diversity enriched with detailed descriptions of each type (e.g., infrastructures, devices), and finally data diversity through the coverage of scalar/multimedia sensed data.

4 HSSN ONTOLOGY

In this section, we detail our proposed extension of the SSN ontology, and mainly our additions related to: (i) sensor diversity; (ii) platform diversity; and (iii) data diversity. The following prefixes *sosa:*, *ssn:*, *mssn:*, *time:*, and *hssn:* refer to the SOSA[7], SSN[7], MSSN[1], Time[8], and HSSN ontologies respectively. We begin first by describing sensor-related concepts.

4.1 Sensor Diversity

4.1.1 Sensor Mobility. Fig.2 illustrates the sensor types added in HSSN. The concept *Sensor* already exists in the SSN ontology, where mobility is not extensively developed. Therefore, we add two child concepts of *Sensor*: (i) *MobileSensor*, describing any sensor that has the ability to move or change location; and (ii) *StaticSensor*, a sensor that does not change location in time. This allows the sensor network to have diverse sensor types (cf. Criterion 1 - Section 3).

4.1.2 Sensor Tracking. Every sensor has a *Location*. To consider mobility, one should be able to locate any sensor at all times. The object property *isCurrentlyLocatedAt* maps each sensor to its current

Location (cf. Challenge 3 in Section 2). This is specifically important for tracking mobile sensors, since static sensors do not change locations (cf. Fig.3). A *hasPastLocation* property is added to retrieve the previous positions of a (mobile) *Sensor*, and also a *hasLocationTime* (cf. Fig.4) property is added to map these positions to time instants or intervals in order to track sensors (temporal entities are extracted from Time ontology [8]).

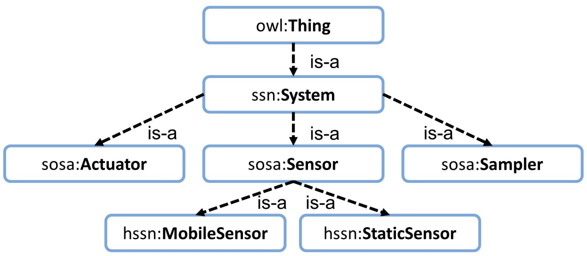

Figure 2: HSSN Sensor View

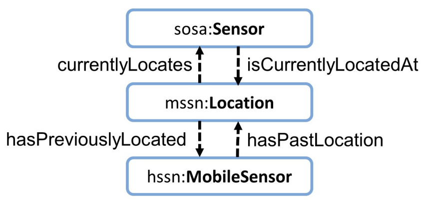

Figure 3: Sensor/Location Mapping

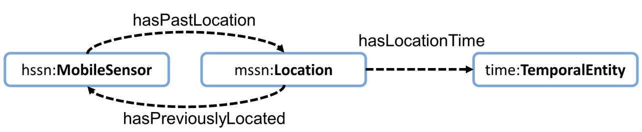

Figure 4: Previous Location/Time Mapping

4.1.3 *Coverage Area.*

Each *Sensor*, mobile or static, has a *CoverageArea* (cf. Fig.5), a geographical zone that contains any sensing activity (i.e., any happening outside of this zone is not detected by the *Sensor*). In order to represent coverage areas, we consider the following: (i) a *CoverageArea* is bound to the sensor's current *Location*; and (ii) the geographical spread of a *CoverageArea* is affected by the sensing range and sensing angles (horizontal and vertical orientation) of the concerned *Sensor*. We represent the coverage area as a sector of space (Fig.6 shows a horizontal slice of the space) where S is the focal point (the sensor's current *Location*), $\alpha, \beta \in [0; 2\pi]$ are the angles that define the horizontal/vertical rotational spread of the coverage area respectively, and the distance $SA = SB$ is the sensing range that defines the extent of the coverage area. The angles and range depend of the sensor's capability properties. For instance, a temperature sensor has $\alpha = \beta = 2\pi$, but a surveillance camera has $\alpha = \frac{\pi}{4}$, $\beta = \frac{\pi}{6}$ if the camera lens is limited to a 45° horizontal angle, and a 30° vertical angle. Similarly, the sensing range varies from one sensor to another (e.g., 10, 20, 50 meters).

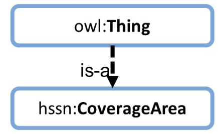

Figure 5: Coverage Area

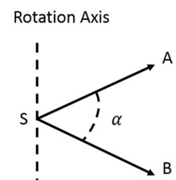

Figure 6: Coverage Area - Horizontal Spread

The composition of a *CoverageArea* is explained in Fig.7. The *SensingLocation* is equivalent to the sensor's *Location*, and the angles and range of the *CoverageArea* are equivalent to the sensor's *HorizontalAngle*, *VerticalAngle*, and *Range* properties that we added in HSSN as part of a system's properties. Since static sensors are immobile, it is easy to know their coverage areas using the sensor's location, and its sensing range and angles. In contrast, knowing the coverage areas of mobile sensors is more challenging, since these areas move when the sensors move. In order to keep track of these changes, the object property *currentlyCovers* maps each *Sensor* to its current *CoverageArea* (cf. Fig.8). Also, the property *hasPastCoverageArea* maps mobile sensors to their respective sets of previous coverage areas (cf. Challenge 3 in Section 2). Finally, *hasCoverageTime* is the property that maps previous coverage areas to temporal entities (i.e., time instant or interval from Time ontology [8]) for tracking purposes (cf. Fig.9).

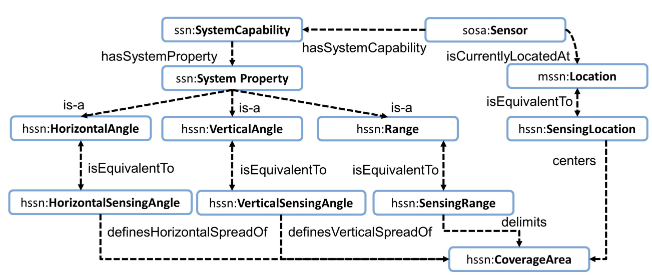

Figure 7: Coverage Area Composition

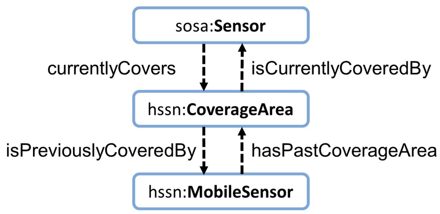

Figure 8: Sensor/Coverage Area Mapping

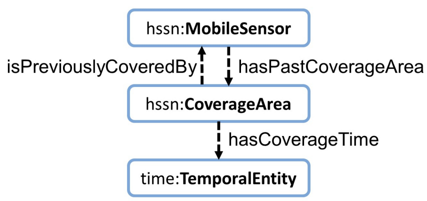

Figure 9: Coverage Area/Time Mapping

4.2 Platform Diversity

4.2.1 Infrastructure Representation. In SSN[7], sensors are deployed on platforms. In Fig.10, we define the following child concepts of *Platform*: (i) *Infrastructure*, a physical environment having locations where sensors could be deployed (cf. Challenge 1 in Section 2); and (ii) *Device*, an electronic equipment where sensors could be embedded (cf. Challenge 2 in Section 2). This allows different types of deployments such as the traditional deployment in environments (e.g., buildings, malls) or nested deployment of multi-purpose devices that in turn embed sensors (e.g., mobile phones). This provides platform diversity (criterion 2 cf. Section 3). Every *Infrastructure* describes a specific physical environment where sensors are deployed. Therefore, infrastructures can host platforms such as other infrastructures (e.g., cities host buildings) and devices (e.g., buildings host mobile phones). However, devices can embed systems of sensors, actuators, and samplers but cannot host infrastructures (e.g., buildings). Each *Infrastructure* is described by a *Location Map* which contains (*isComposedOf* property) a set of *Locations* (cf. Fig.11). For example, a building is an *Infrastructure* that has a *Location-Map*. The latter describes the spatial relations between individual *Locations* in the building such as floors, offices, etc. HSSN uses topological, distance, and directional relations to describe the spatial ties that exist between individual *Locations*. We integrate the aforementioned location-related concepts in order to locate sensors, and better understand the spatial constraints/setup of the *Infrastructure*.

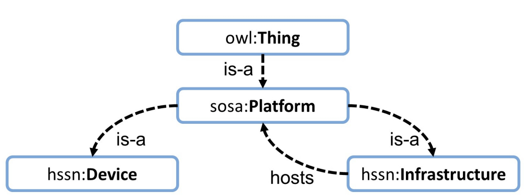

Figure 10: Platform Representation

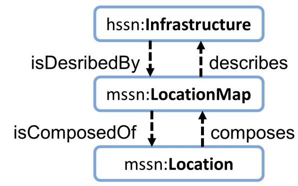

Figure 11: Infrastructures

4.2.2 Device Representation. A *Device* is another type of *Platform* where sensors are deployed. It is introduced in HSSN to represent mobile phones and other sensing equipment. A *Device* has sub-concepts for storage, communication, processing, and power supply, in addition to the ability of embedding sensors (using the *deployEntity* concept cf. Fig.12). These concepts describe the *Hardware* of a *Device*. The *Software* part is also represented. A *Device* could be used for various purposes (e.g., representing mobile phones for mobile phone sensing, machines with mounted sensors for fault detection in an Industry 4.0 scenario). The hardware and software representation allows complex queries such as assigning sensing tasks to devices based on their processing capabilities, or battery status (cf. Challenge 2 in Section 2). Finally, each *Device* can provide a set of services. Fig.13 illustrates our service modeling, inspired by the Web Service Modeling Ontology (WSMO) [12]. We created generic concepts that can be aligned with WSMO. We do not aim to detail the service description to allow alignments with any other service ontology. We limit the service modeling to the following concepts: Service *Metadata* describes the properties of a *Service*. The *Input* represents the set of variables and constraints required for correct service execution, while the *Output* is the set of generated results. The functionality of a service is described by the *Capability* concept which is mapped to a specific *UserGoal* or objective (i.e., a user desire satisfied by the service). Users communicate with a service through *UserInteractionInterfaces* (choreography in WSMO). Finally, services communicate with each other via the *ServiceInteractionInterface* (service orchestration in WSMO). Finally, the infrastructure and device detailing also improves sensor diversity by allowing the representation of simple sensor nodes in infrastructures, multi-sensor systems, and multi-sensor devices.

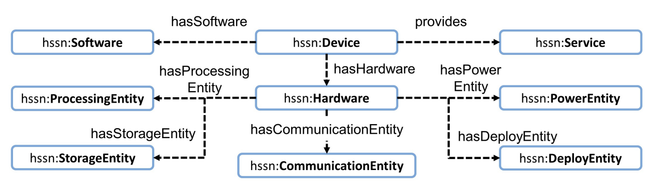

Figure 12: Device Components

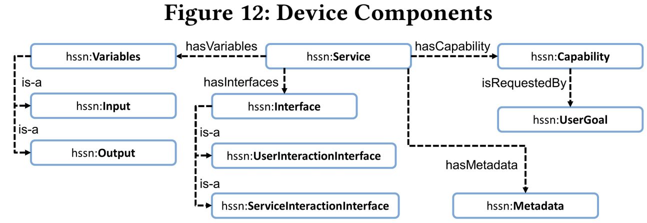

Figure 13: Service Components

4.3 Data Diversity

Audio, image, and video data can be sensed by mobile or static sensors (e.g., surveillance cameras, mobile phones). Also, in order to detect complex events (e.g., gunshot) a combination of multimedia and scalar observations is needed. Therefore, we aim to integrate concepts related to multimedia properties (cf. Criterion 3 in Section 3). In MSSN [1], multimedia data/properties are integrated in SSN. We re-organize MSSN multimedia concepts into scalar (e.g., temperature, motion) and multimedia (e.g., noise, video) properties as illustrated in Fig.14. Also, we introduce in Fig.15 the *mediaSenses* and *scalarSenses* relationships to map sensors to their corresponding scalar and/or multimedia observable properties (cf. Challenge 4 in Section 2). This highlights the sensor diversity in HSSN since static/mobile sensors can detect scalar and/or multimedia properties. The authors in [1] also describe technical aspects/metadata of multimedia objects such as annotations, audio (e.g., frequencies), motion (e.g., trajectories), visual (e.g., color histograms). We use these concepts in HSSN to describe sensor observation values. A *MediaValue* in HSSN is composed of the *MultimediaData* concept, referring to the audio, video, or image objects/files and the *MediaDescriptor* concepts, describing the metadata of the multimedia objects (e.g., frequencies, colors). *ScalarValues* are textual (e.g., temperatures, humidity levels). Finally, we map observation values to their related properties using the *hasMediaValue* and *hasScalarValue* relationships. Sensors can now be correctly mapped to observable properties and observation values (cf. Challenge 4 in Section 2).

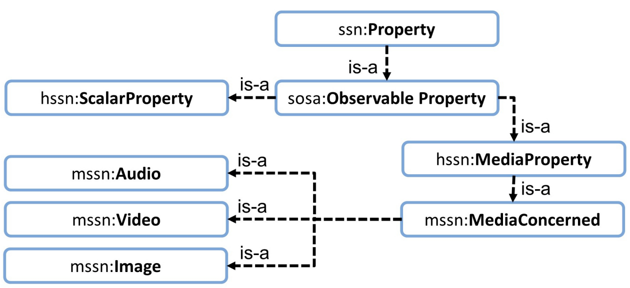

Figure 14: Observable Properties

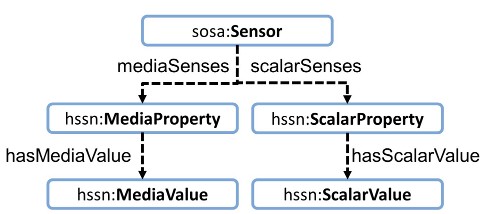

Figure 15: Sensors/Properties

In conclusion, new concepts and properties are introduced in HSSN in order to address the challenges presented in Section 2. Our proposal details the representation of infrastructures (a type of platforms) by adding location maps, individual locations, and spatial relations. This allows the expressively describe locations (cf. Challenge 1). In HSSN we describe devices as platforms that host sensors. We detail device hardware, software, and provided services. In addition, we add properties that help locate, track, and query these devices (cf. Challenge 2). HSSN also provides a description of sensor coverage areas and properties that map both locations and coverage areas to mobile/static sensors at any time (cf. Challenge 3). Finally, we address data heterogeneity by detailing multimedia data objects, their metadata, and scalar data. We also map them to their respective sensors (cf. Challenge 4).

5 IMPLEMENTATION AND EXPERIMENTAL SETUP

5.1 HSSN Implementation

We implemented the HSSN ontology using Protege 5.2.0[2]. The files are available at http://spider.sigappfr.org/research-projects/hybrid-ssn-ontology/ (External Links - Download ontology files). Also, a complete documentation can be found at http://spider.sigappfr.org/HSSNdoc/index-en.html. In the following, we detail the SPARQL queries used during the experimentation. Then, we describe the experimental setup, before discussing the obtained results from an accuracy, clarity, performance, and consistency standpoint.

5.2 Illustration Example

The challenges mentioned in Section 2 can be addressed via the following SPARQL queries: **Platform Diversity:** In order to expressively describe locations (Challenge 1) in the mall infrastructure, a detailed representation of location maps and locations is needed (Query 1). Also, covered and uncovered areas should be easily found (Query 2). In order to consider ad-hoc devices in the network (Challenge 2), one should be able to query devices, their hardware (e.g., embedded sensors), software, and services. Query 3 shows how to locate a mobile device by querying its embedded sensor. Similarly, one could query a device based on other characteristics (e.g., battery status, processing power).

Query 1: Knowing the spatial description of infrastructures

```
SELECT distinct ?infrastructure ?locationmap ?location WHERE
{?infrastructure isDescribedBy ?locationmap. ?locationmap isComposedOf ?location.}
```

Query 2: Knowing covered locations

```
SELECT distinct ?location ?coveragearea WHERE {?location isIncludedIn ?coveragearea.}
```

Query 3: Locating mobile devices, querying device hardware

```
SELECT distinct ?location ?dev WHERE {?location currentlyLocates ?sensor.
?sensor isEmbeddedOn ?du. ?du hasExpansionCard ?hd. ?hd isRelatedToDevice ?dev.}
```

Sensor Diversity: To track sensors at all times (Challenge 3), it is important to know current locations/coverage areas for all sensors (Query 4), as well as previous ones (Query 5).

[2]https://protege.stanford.edu/

Query 4: Finding current sensor locations/coverage areas

```
SELECT distinct ?location ?sensor ?coveragearea WHERE
{?location currentlyLocates ?sensor. ?sensor currentlyCovers ?coveragearea.}
```

Query 5: Finding previous sensor locations

```
SELECT distinct ?location ?sensor WHERE {?location hasPreviouslyLocated ?sensor}
```

Data Diversity: In order to consider data diversity (Challenge 4), on should be able to distinguish scalar/multimedia data and correctly map them to sensors (Queries 6 and 7).

Query 6: Mapping sensors to their scalar properties and observations

```
SELECT distinct ?sensor ?property ?observation WHERE
{?sensor scalarSenses ?property. ?property isScalarValueOf ?observation.}
```

Query 7: Mapping sensors to their multimedia properties and observations

```
SELECT distinct ?sensor ?property ?observation WHERE
{?sensor mediaSenses ?property. ?property isMediaValueOf ?observation.}
```

5.3 HSSN Experimental Setup

Here, we did not aim to experiment SSN concepts and properties. We evaluated the impact of our newly added concepts (e.g., static/mobile sensors, infrastructures/devices, multimedia/scalar data). Our objectives were the following:

(1) *Accuracy Evaluation:* Checks if the added concepts/properties answer the aforementioned challenges. This query based evaluation highlights the impact of our extensions in overcoming the challenges mentioned in Section 2.
(2) *Clarity Evaluation:* Checks if the labels used to describe the concepts/properties are clear and unambiguous to domain stakeholders. The aim is to evaluate the compatibility and clarity of our provided description with respect to the application domain.
(3) *Performance Evaluation:* Measures the impact of HSSN additions on performance (i.e., query run time). The aim is to evaluate the feasibility, performance-wise, of integrating HSSN in sensor network applications.
(4) *Consistency Evaluation:* Checks if the added concepts/properties generate inconsistencies (e.g., anti-patterns) within the structure of the ontology. The aim is to evaluate the soundness of the ontology graph.

5.3.1 Accuracy Evaluation. We created a population of individuals and ran the aforementioned queries. Then, we compared the obtained and expected results. We created two infrastructures, each described by a location map containing 500 locations. Then, 1000 sensors were deployed (500 mobile/static, 500 scalar/media). Each sensor is located in one location, covers one coverage area, observes one property, and produces one observation value.

Platform Results: We ran queries 1, 2, and 3. The returned results match perfectly the expected ones. Infrastructures were correctly assigned to their location maps and included locations. This allowed the identification of distinct spaces/areas. Query 2 correctly returned the set of distinct locations included in each coverage area. This allowed the identification of non covered locations. Query 3 allowed the identification of device hardware related to the embedded sensors. Also, the mobile devices were correctly located in the location map.

Mobility Results: We ran queries 4 and 5 on the population of individuals and for each case the returned results matched exactly the expected ones. Sensors were correctly assigned to their current/previous locations and coverage areas.

Data Results: We ran queries 6 and 7 and obtained an exact matching between the actual and expected results. Thus, scalar/multimedia properties were correctly distinguished. Also, sensors were correctly assigned to the scalar or multimedia observations that they produced.

Result Discussion: The test results showed that locating any type of sensor (i.e., simple node/multi-sensor device, static/mobile sensors, and scalar/multimedia sensors), and knowing their coverage areas is possible at any point in time. Hence, allowing tasks such as tracking mobile sensors, and detecting uncovered areas. Also, the results showed that the detailing of infrastructure and device descriptions (platform diversity) allowed a better knowledge of the environment space (also important for locating sensors). Multi-sensor devices were also detailed by describing their hardware and software which proved useful when querying devices based on their capabilities (e.g., we ran an additional query that returns sensors/devices with good battery status). From a data diversity standpoint, the results showed that sensors that sense multimedia/scalar properties were correctly distinguished and their observations were accurately retrieved. To conclude, the query results confirmed that the added extensions (i.e., regarding sensor, platform, and data diversity) accurately answer the challenges mentioned in Section 2.

5.3.2 Clarity Evaluation. We created two evaluation forms: the first[3] for evaluating the ambiguity of the labels used to describe the HSSN concepts, and the second[4] for evaluating the ambiguity of the labels used to describe inter-concept relations. We sent the two forms to 50 sensor network and ontology experts (25 networking experts, and 25 computer scientists). Results in Fig.16 and 17 show that terms considered clear by computer scientists are sometimes found ambiguous by network experts and vice-versa. Fig.16 shows that a few terms do not meet the acceptable ambiguity level (e.g., ComUnit, DeployUnit), while others (e.g., MediaProperty, MediaValue) need some clarification. Therefore, we considered the experts' suggestions in the final version of the ontology by modifying the following: (i) *ExpansionCard* instead of *DeployUnit*; (ii) *PowerSupply* instead of *PowerUnit*; (iii) *NetworkInterface* instead of *ComUnit*; (iv) *Memory* instead of *StorageUnit*; (v) *Processor* instead of *ProcessingUnit* ; and (vi) *Multimedia* instead of *Media*. Finally, Fig.17 shows that in most cases, both categories of experts assigned correctly the inter-concept relationships. Networking experts have low success on the first two questions since the latter are outside of their domain of expertise (regarding inheritance between concepts).

[3]Link: https://goo.gl/forms/blc8pKLLqtNtjXHI2
[4]Link: https://goo.gl/forms/KNNY3XsmGp0ptM2N2

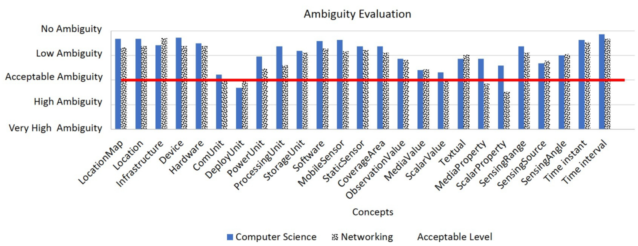

Figure 16: Concept Evaluation

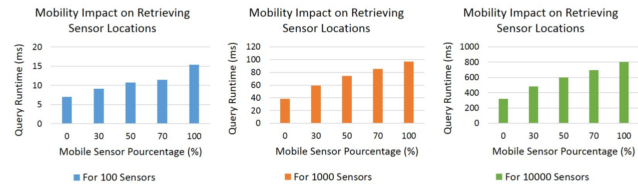

Figure 18: Mobility impact on current location retrieval

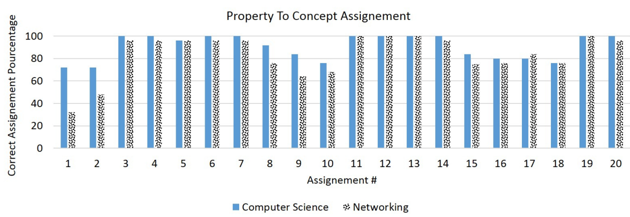

Figure 17: Property Evaluation

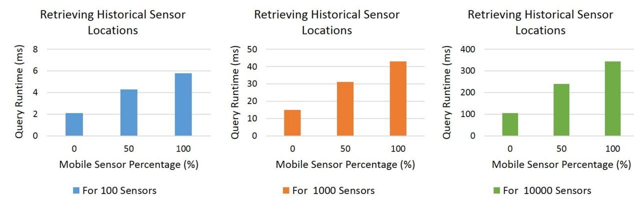

Figure 19: Mobility impact on previous location retrieval

Result Discussion: The clarity evaluation allowed the identification and correction of ambiguous/unclear labels that we used to describe our added concepts/properties. In the version currently available online, all labels achieve an acceptable level of clarity (based on the stakeholders' feedback). This reinforces the re-usability of HSSN since it is unambiguous and easily understood.

5.3.3 Performance Evaluation. In order to evaluate the performance of HSSN, we measured the query run-time by running each of the previously mentioned queries 10 times and calculating the average. We varied the size of the population (100 sensors, 1000 sensors, and 10000 sensors) in order to test various scenarios related to mobility, platforms, and data.

Mobility impact: In this test, we varied the percentage of mobile sensors in the network (0, 30, 50, 70, and 100 %). Then, we retrieved the current/previous sensor locations (cf. Fig.18 and 19). We measured the run-time for queries 4 and 5. In Fig.18, we noticed that increasing the number of mobile devices increases the time required to retrieve current sensor locations. This is due to the fact that locating a device (Query 3) was a more complex task than locating a static sensor since we needed to locate the sensor, its deployment unit, hardware, and then the device. We noticed the same pattern for all three cases (100, 1000, 10000 sensors). Finally, the progression from 0% to 100% mobile devices had a quasi-linear impact on query run-time. Similarly, Fig.19 details the query run-time for retrieving previous different sensor locations. Since mobile sensors have a larger list of previous locations in comparison with static sensors, increasing the mobility percentage (0, 50, 100 %) increases the query run-time. This progression was also quasi-linear for all three cases (100, 1000, 10000 sensors).

Platform impact: In this test, we varied the sensor distribution on the platform locations. We tested three different scenarios (i) each sensor is located in one location; (ii) all sensors are located in one location; and (iii) half of the sensors are located in a location and the other half in another. We measured the run-time of the query that retrieves sensor locations.

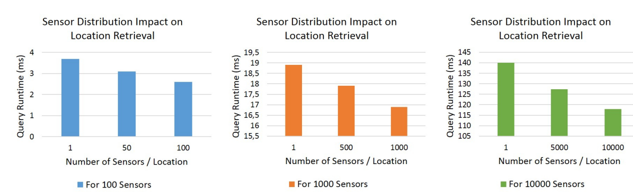

Figure 20: Platform impact on current location retrieval

Fig.20 shows how sensor distribution on locations affected the time needed to map sensors to their current locations. When all sensors were located in one location, the required time to perform this task was minimal. Then, as we began to decrease sensor densities, the query took more time. Finally, the worst case was when every location contained only one sensor.

Data impact: Here, we checked the impact of scalar/multimedia data on the run-time of queries 6 and 7 (cf. Fig.21).

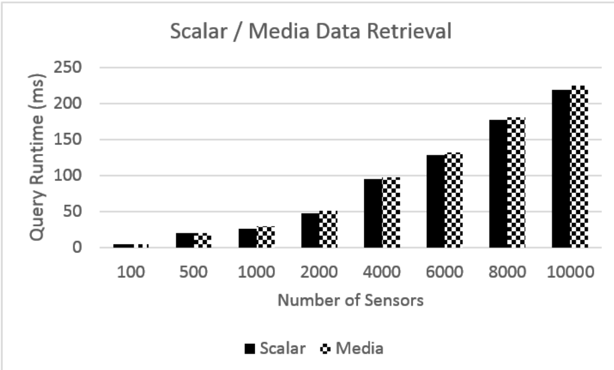

Figure 21: Data impact on observation retrieval

For data diversity impact on performance (cf. Fig.21), we noticed that in all cases (100, 1000, 10000 sensors) the query run-time was similar when considering scalar and multimedia data. This is due to the fact that we were measuring the time required to retrieve the data and not the time needed to capture/sense it.
Result Discussion: The performance evaluation showed that the added concepts/properties do not heavily impact the query run time, which remains quasi-linear in most cases. This highlights the feasibility of using of HSSN in sensor applications (from a performance point of view).

5.3.4 Consistency Evaluation. In [15], consistency is defined as a criterion that verifies if the ontology allows contradictions. The descriptions in the ontology should be consistent.
Consistency Queries: To evaluate consistency, we adopted the following SPARQL queries that search for anti-patterns, a strong indicator of inconsistencies, in the ontology. Query 8 detects concepts with no parent, and query 9 detects abnormally disjointed concepts in the ontology:

Query 8: Searching for concepts with no parent

```
SELECT ?a WHERE {?a subClassOf owl:Nothing.}
```

Query 9: Searching for abnormally dijointed concepts

```
SELECT distinct ?A ?B1 ?B2 ?C1 WHERE
{?B1 subClassOf ?A. ?B2 subClassOf ?A. ?C1 subClassOf ?B1. ?C1 disjointWith ?B2.}
```

Results & Discussion: We found no inconsistencies in the HSSN ontology structure. The only concept subsuming nothing is owl:Nothing (Query 8). Query 9 results indicate that there are no concepts that have abnormal disjoint relations with their relatives. This denotes the soundness of the integration of newly added concepts mainly with the SSN core. Finally, to conclude the inconsistency evaluation, we ran Protege's HermiT 1.3.8.413 reasoner, and found no inconsistencies between the asserted class hierarchy and inferred one. This highlights the soundness of the graph structure, which proves critical when considering future alignments between HSSN and other ontologies (e.g., that describe smart buildings, events).

6 CONCLUSION & FUTURE WORK

Many works adopted ontologies for better semantic representation of sensor networks. These approaches do not fully consider diversity in terms of sensors, data, platforms, and application purposes. In this paper, we propose an extension of the Semantic Sensor Network ontology (SSN), since it is already re-usable in various contexts. Our proposed ontology, denoted HSSN, adds to SSN sensor mobility, and multimedia data related concepts in order to have a representation of hybrid sensor networks. HSSN also extends the platform representation of SSN in order to fully consider platform diversity. We implemented HSSN, evaluated the consistency, accuracy of our additions, and their impact on performance. As future work, we would like to continue the ongoing evaluation of the completeness of the ontology through comparisons with mobility and sensor taxonomies. Finally, we want to represent a sensor network in a smart environment (e.g., smart building, city) for event detection purposes.

REFERENCES

[1] Chinnapong Angsuchotmetee, Richard Chbeir, and Yudith Cardinale. 2018. MSSN-Onto: An ontology-based approach for flexible event processing in Multimedia Sensor Networks. *Future Generation Computer Systems* (2018). https://doi.org/10.1016/j.future.2018.01.044
[2] Sasikanth Avancha, Anupam Joshi, Chintan Patel, et al. 2004. Ontology-driven adaptive sensor networks. *MobiQuitous 2004* (2004), 194–202.
[3] Payam Barnaghi, Stefan Meissner, Mirko Presser, and Klaus Moessner. 2009. Sense and sens'ability: Semantic data modelling for sensor networks. In *Conference Proceedings of ICT Mobile Summit.*
[4] David M Doolin and Nicholas Sitar. 2005. Wireless sensors for wildfire monitoring. In *Smart Structures and Materials 2005: Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*, Vol. 5765. International Society for Optics and Photonics, 477–485.
[5] Nicolas Durand, Sebastien Derivaux, Germain Forestier, Cedric Wemmert, Pierre Gançarski, Omar Boussaid, and Anne Puissant. 2007. Ontology-based object recognition for remote sensing image interpretation. In *Tools with Artificial Intelligence, 2007. ICTAI 2007.*, Vol. 1. IEEE, 472–479.
[6] Mario Gomez, Alun Preece, Matthew P Johnson, Geeth De Mel, Wamberto Vasconcelos, Christopher Gibson, Amotz Bar-Noy, Konrad Borowiecki, Thomas La Porta, Diego Pizzocaro, et al. 2008. An ontology-centric approach to sensor-mission assignment. In *International Conference on Knowledge Engineering and Knowledge Management.* Springer, 347–363.
[7] Armin Haller, Krzysztof Janowicz, Simon JD Cox, Maxime Lefrançois, Kerry Taylor, Danh Le Phuoc, Joshua Lieberman, Raúl García-Castro, Rob Atkinson, and Claus Stadler. 2018. The Modular SSN Ontology: A Joint W3C and OGC Standard Specifying the Semantics of Sensors, Observations, Sampling, and Actuation. *Semantic Web - Interoperability, Usability, Applicability an IOS Press Journal* (2018).
[8] Jerry R Hobbs and Feng Pan. 2006. Time ontology in OWL. *W3C working draft* 27 (2006), 133.
[9] Nicolas Maisonneuve, Matthias Stevens, and Bartek Ochab. 2010. Participatory noise pollution monitoring using mobile phones. *Information Polity* 15, 1, 2 (2010), 51–71.
[10] Vasileios Mezaris, Ioannis Kompatsiaris, and Michael G Strintzis. 2003. An ontology approach to object-based image retrieval. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, Vol. 2. IEEE, II–511.
[11] Rajib Kumar Rana, Chun Tung Chou, Salil S Kanhere, Nirupama Bulusu, and Wen Hu. 2010. Ear-phone: an end-to-end participatory urban noise mapping system. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks.* ACM, 105–116.
[12] Dumitru Roman, Uwe Keller, Holger Lausen, Jos De Bruijn, Rubén Lara, Michael Stollberg, Axel Polleres, Cristina Feier, Cristoph Bussler, and Dieter Fensel. 2005. Web service modeling ontology. *Applied ontology* 1, 1 (2005), 77–106.
[13] David J Russomanno, Cartik Kothari, and Omoju Thomas. 2005. Sensor ontologies: from shallow to deep models. In *System Theory, 2005. SSST'05.* IEEE, 107–112.
[14] Kazem Sohraby, Daniel Minoli, and Taieb Znati. 2007. *Wireless sensor networks: technology, protocols, and applications.* John Wiley & Sons.

[15] Steffen Staab and Rudi Studer. 2010. *Handbook on ontologies.* Springer Science & Business Media.
[16] Thanos G Stavropoulos, Dimitris Vrakas, Danai Vlachava, and Nick Bassiliades. 2012. Bonsai: a smart building ontology for ambient intelligence. In *Proceedings of the 2nd international conference on web intelligence, mining and semantics.* ACM, 30.