Neurocomputing 250 (2017) 76–100

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

MIMU-Wear: Ontology-based sensor selection for real-world wearable activity recognition

Claudia Villalonga[a,*], Hector Pomares[a], Ignacio Rojas[a], Oresti Banos[b]

[a] Research Center for Information and Communications Technologies of the University of Granada, C/Periodista Rafael Gomez Montero 2, Granada, Spain
[b] Telemedicine Group, Center for Telematics and Information Technology, University of Twente, Enschede, The Netherlands

ARTICLE INFO

Article history:
Received 7 March 2016
Revised 9 September 2016
Accepted 22 September 2016
Available online 8 February 2017

Keywords:
Ontology
Activity recognition
Magnetic and inertial measurement unit
Wearable sensor platform
Sensor selection
Sensor description
Sensor replacement

ABSTRACT

An enormous effort has been made during the recent years towards the recognition of human activity based on wearable sensors. Despite the wide variety of proposed systems, most existing solutions have in common to solely operate on predefined settings and constrained sensor setups. Real-world activity recognition applications and users rather demand more flexible sensor configurations dealing with potential adverse situations such as defective or missing sensors. In order to provide interoperability and reconfigurability, heterogeneous sensors used in wearable activity recognition systems must be fairly abstracted from the actual underlying network infrastructure. This work presents MIMU-Wear, an extensible ontology that comprehensively describes wearable sensor platforms consisting of mainstream magnetic and inertial measurement units (MIMUs). MIMU-Wear describes the capabilities of MIMUs such as their measurement properties and the characteristics of wearable sensor platforms including their on-body location. A novel method to select an adequate replacement for a given anomalous or nonrecoverable sensor is also presented in this work. The proposed sensor selection method is based on the MIMU-Wear Ontology and builds on a set of heuristic rules to infer the candidate replacement sensors under different conditions. Then, queries are iteratively posed to select the most appropriate MIMU sensor for the replacement of the defective one. An exemplary application scenario is used to illustrate some of the potential of MIMU-Wear for supporting seamless operation of wearable activity recognition systems.

1. Introduction

Human activity recognition has lately drawn the attention of research and industry due to its application potential in areas such as manufacturing [1], sports [2] or health [3]. Different technologies have been proposed for activity recognition, albeit wearable sensors, particularly magnetic and inertial measurement units (a.k.a., MIMUs), take over most of the market nowadays. MIMUs are very cheap and tiny sensors normally embedded into ergonomic wearable platforms that can be worn by users to track the motion of the body parts where these devices are placed on. The recognition process itself consists in the automatic analysis of the signals or physical magnitudes measured by MIMUs, namely acceleration, rate of turn and magnetic field orientation. Signal processing and machine learning techniques are normally used to categorize these measurements into a specific activity kind [4].

Although several wearable activity recognition solutions have been provided to date [5–7], most of them are conceived to operate in closed environments, where sensor setups are pre-defined, well-known and steady. However, these conditions cannot be guaranteed in practical situations, where energy and computational resources are limited and sensors may be subject to diverse types of anomalies such as failures [8] or deployment changes [9]. Hence, supporting dynamic sensor selection and replacement is seen to be a key requirement for realistic activity recognition systems in order to ensure a fully functional operation. A few probabilistic and machine-learning models have been proposed for the dynamic selection of sensors [10–12]. However, these models present important limitations as they develop the selection process on the properties of the sensor data streams rather than capabilities and nature of sensors and device. More importantly, in all cases the sensor ecosystem must be known in advance, thus not supporting changeable scenarios with opportunistic additions or removals of sensor devices [13].

To enable sensor replacement functionalities in a wearable activity recognition system, mechanisms to abstract the selection of the most adequate sensors are needed. Precisely, a

* Corresponding author.
E-mail addresses: cvillalonga@correo.ugr.es (C. Villalonga), hector@ugr.es (H. Pomares), irojas@ugr.es (I. Rojas), o.banoslegran@utwente.nl (O. Banos).

http://dx.doi.org/10.1016/j.neucom.2016.09.125
0925-2312/

comprehensive and interoperable description of these heterogeneous sensors is required, including aspects such as their on-body location or availability. This information, in combination with sophisticated search techniques could support the selection of the best replacement for an anomalous sensor. The most simplistic sensor description approach would consist in using tags that could be used by human users. However, free-text tags are insufficient for any machine-based interaction, where the sensor selection has to be executed automatically. In this case, the syntax and semantics of the sensor description rather need to be clearly defined. EXtensible Markup Language (XML) descriptions could be considered to this end. Nonetheless, XML does not provide the full potential for machines to acquire and interpret the emerging semantics from data, thus the meaning of the data has to be previously agreed in between machines. Conversely, an ontology-based formal data representation like OWL 2 [14] solves these problems while providing interoperability at the only expense of some extra overhead. In view of these advantages this work proposes the use of ontologies to thoroughly describe the wearable sensors available for the activity recognition process, further enabling the semantic selection of sensors to support a continuity of recognition.

2. Related work

This work proposes the use of ontologies for the semantic description of the MIMU sensors predominantly used in wearable activity recognizers. The final goal of the ontological modeling of activity recognition systems is enabling semantic sensor selection to ensure the continuity of recognition in case of sensor malfunctioning.

Ontologies have not been used so far in the activity recognition domain to model wearable sensor networks, abstract them from the actual underlying sensing infrastructure, and provide interoperability. However, much effort has been put in order to assure interoperability for sensor networks in the context of the Internet of Things, a much more generic scenario in where sensors of many diverse types become part of the global sensing infrastructure. Therefore, in Section 2.1 the available ontologies to model sensor networks are analyzed to assess their applicability to the activity recognition domain and to possibly benefit from their usage in that context. None of the ontologies appear to be directly applicable since they do not provide the required domain knowledge; however, they could be extended. The SSN ontology is possibly the most popular and widely-adopted sensor ontology and seems to be the perfect candidate for this purpose.

The application domain of the ontologies in this work is the semantic sensor selection in activity recognition systems. Therefore, Section 2.2 describes the existing approaches applying semantic sensor selection. The same issue arises here, none of the proposed solutions tackles the sensor replacement in activity recognition systems. Despite their application to other domains, these approaches are seen to be interesting for this work.

2.1. Ontologies for sensor networks

In the last decade many ontologies have been devised for the modeling of sensors and sensor networks. These ontologies provide a description of the sensor networks, the sensing devices, the measured information or data, the processes executed in the sensor network, and enable sensor data fusion.

The Sensor Web Enablement (SWE) initiative [15] of the Open Geospatial Consortium (OGC) has approved a set of standards and best practices for the sensors to interoperate with the Web, in what is called the Sensor Web. The OGC SWE has developed a set of standard models and XML schemas for sensors and processes in SensorML [16], and for sensor data in Observations and Measurements (O&M) [17,18]. These standards provide syntactic interoperability but lack semantic compatibility. Therefore, semantic web technologies are used to augment the OGC SWE standards in what is known as the Semantic Sensor Web [19].

OntoSensor [20] is an ontology which builds on the ideas of the OGC SensorML standard and extends the Suggested Upper Merged Ontology (SUMO) [21]. The objective of OntoSensor was to create a sensor knowledge repository enabling the fusion of heterogeneous data. Therefore, OntoSensor provides a description of the data observed by the sensors, including the geo-location of the observations, the accuracy of the observed data or the process to obtain the data.

The GLOSENT (GLObal SENsor neTwork) architecture [22] facilitates the integration of wireless sensor networks by utilizing semantics to resolve hardware heterogeneities. The proposed ontology models large systems of wireless sensor networks where sensor nodes are interpreted as sets of components, including sensor components and processing components, like a memory component or a radio component. Therefore, the GLOSENT architecture relies on the ontological representation of the wireless sensor networks and their data.

The W3C Semantic Sensor Network Incubator group (SSN-XG) has defined the SSN Ontology [23] in order to provide the layer of abstraction required to address semantic compatibility missing in the OGC SWE standards. The SSN Ontology describes the capabilities and properties of the sensors, the act of sensing and the resulting observations. The SSN Ontology covers large parts of the SensorML and O&M standards, omitting the concepts which are sensor specific, like calibrations, process descriptions and data types. The SSN Ontology was developed with focus on four types of use cases: data discovery and linking, device discovery and selection, provenance and diagnosis, and device operation, tasking and programming. Therefore, the SSN Ontology has been used in many research projects and applied to several different domains in the last years. Some of the most recently published works which utilize the SSN Ontology are the OpenIoT Project [24,25], the Semantic Gateway as Service (SGS) [26] and GeoSMA [27].

The SSN Ontology has been extended in the Wireless Semantic Sensor Network (WSSN) ontology [28]. Specifically, the communication data policy which is not characterized by the SSN ontology has been added in the WSSN ontology. The newly described pattern for communication is required to ensure the main objective of the WSSN ontology of adapting the nodes communication to optimize the lifetime of the network.

A more recent solution for handling the heterogeneity of wireless sensor networks is MyOntoSens [29]. This ontology formalizes a semantic open data model for the generic description of sensor and sensor data. MyOntoSens builds on some ideas of OntoSensor, SSN and SensorML, and divides the concepts in three categories: wireless sensor network, node and process. In the modeling of the wireless sensor network, standardized attributes like the application domain, coverage zone, location and radio technology are considered. This enables the automatic discovery of available neighboring wireless sensor networks, wireless sensor networks sharing similar properties or devised for the same application domain. The MyOntoSens ontology has been recently utilized in [30]. Moreover, a Body Area Network (BAN) dedicated instance of the MyOntoSens ontology is being standardized as a Technical Specification within the SmartBAN Technical Committee of the European Telecommunications Standards Institute (ETSI).

The SmartBAN open data model ontology [31] is part of the ETSI initiative which standardizes to support the development and implementation of BAN technologies in the domains of health, wellness, leisure and sport. The SmartBAN ontology aims at developing smarter control and monitoring operations as well as

standardized eHealth services. Therefore, the SmartBAN ontology has been designed to be utilized together with existing healthcare and telemedicine information models and standards. The SmartBAN ontology builds on three sub-ontologies: WBAN (SmartBAN or BAN cluster), Nodes (i.e., Hub, sensors, actuators) and Process and Measurements.

Finally, the Sensing Network Element Ontology Description Model for Internet of Things [32] has been developed quite recently. This ontology describes the sensing devices, their capabilities and the sensory data to automatically discover and interact with the elements of the Internet of Things. The structure, main classes and properties of this ontology are quite similar to the ones described in the SSN ontology; however, domain knowledge about the Internet of Things has been introduced.

2.2. Semantic sensor selection

The interoperability provided by the ontological description of the sensor network enables a set of interesting applications, such as semantic sensor selection.

One of the first attempts to perform semantic sensor selection was developed in the SENSEI project [33]. An ontology was proposed to model the description of wireless sensor and actuator networks, including the resource type, location, temporal availability, generated outputs, required inputs, pre-conditions and post-conditions, and quality and cost parameters [34]. Declarative requests, specifying the specific context or sensor information requested by an application were automatically interpreted and matched against the specific parameters of the sensor and actuator descriptions.

A similar approach is presented in a much more recent work. Hsu et al. [35] propose an infrastructure which allows the sensor selection based on the sensor characteristics, such as accuracy, sensing range, or residual energy. The SSN ontology is used in this work to represent the properties of the sensor. A web interface is offered to the user to select the parameters for the search, including the location, the sensing type, the required number of sensors and some optional requirements like the minimum accuracy.

In CASSARAM [36], another model for semantic sensor selection, ontologies are combined with filtering techniques to improve the sensor ranking in the selection process. CASSARAM builds on sensor descriptions represented using the SSN ontology and considers in the selection both user preferences and sensor characteristics such as reliability, accuracy, location, or battery life.

3. MIMU-Wear: an ontology for the description of MIMU-based wearable platforms

MIMU-Wear is an extensible ontology that describes wearable sensor platforms consisting of magnetic and inertial measurement units (MIMU). MIMU-Wear is an OWL 2 ontology designed in a modular manner with an upper ontology and several plugable domain ontologies (see Fig. 1). The MIMU-Wear Ontology builds on the standard W3C Semantic Sensor Network (SSN) Ontology [23], an ontology which describes sensor networks of any nature and available at http://purl.oclc.org/NET/ssnx/ssn. The SSN Ontology does not model the sensor specific concepts, such as sensor types, features, properties, units of measurement or locations, and these need to be defined in external ontologies. MIMU-Wear extends the SSN Ontology and describes these concepts for the case of MIMUs and wearable sensor platforms. The reuse of this existing ontology facilitates the design of MIMU-Wear since the key concepts are already modeled and can be directly inherited. Moreover the use of the SSN Ontology increases the chances of a higher adoption for the MIMU-Wear Ontology. The SSN Ontology is already used in the research community (as presented in Section 2), and therefore, the novel MIMU-Wear could be directly integrated with the available ontologies using SSN. The two main domain ontologies of MIMU-Wear are the MIMU Ontology (see Section 3.1) and the Wearable Sensor Platform Ontology (see Section 3.2). The MIMU Ontology describes the capabilities of MIMUs, for example, the physical property measured by a magnetometer. The Wearable Sensor Platform Ontology models the characteristics of wearable sensor platforms, including the location where the wearable is placed on the human body. The MIMU Ontology and the Wearable Sensor Platform Ontology model the basic common concepts and import several domain ontologies which describe in more detail concepts like the magnitude, the units, the measurement and the survival properties, and the human body.

An important benefit of the modularity of the MIMU-Wear is its easy extensibility. The different modules are self-contained and enable extending each of the ontology parts in an independent manner. Another important benefit of the MIMU-Wear modularity is its reusability in other domains. The Wearable Sensor Platform Ontology could be used to describe the location on the human body of any wearable sensors besides MIMUs. Using this ontology, the location of an ECG sensor in a belt could be easily described. Similarly, the MIMU Ontology could be used to describe any MIMUs, this means not only the wearable ones but also the ones embedded in ambient intelligence platforms. Using this ontology, the characteristics of a MIMU integrated in a cap or door in an ambient assisted living scenario could be modeled. The same way the MIMU Ontology and the Wearable Sensor Platform Ontology are easily combined here, the MIMU-Wear Ontology could be extended to cover new domains like the physiological wearable sensors or the ambient MIMUs.

3.1. MIMU ontology

The MIMU Ontology models the characteristics of the MIMUs, for example, the magnitude observed by a gyroscope or the measurement range of an accelerometer. The SSN Ontology is here extended to model the particular features of the MIMUs. Thus, the particular vocabularies for the properties measured by the MIMUs and the measurement capabilities of the MIMUs, which are not part of the SSN Ontology, are here extensively defined.

The main class of the MIMU Ontology is the class `MIMU` which represents the set of all the potential MIMU sensors (see Fig. 2). The class `MIMU` is defined to be a subclass of the class `ssn:SensingDevice` in the SSN Ontology. The prefix `ssn` in the class name indicates that the element belongs to the SSN Ontology. Specifically, the class `ssn:SensingDevice` is a subclass of the class `ssn:Sensor` and of the class `ssn:Device`, and represents any physical sensors. Anything that observes is considered a sensor in the SSN Ontology (`ssn:Sensor`). This definition of sensor is very broad and can include any hardware device, computational model, and even a human being. In order to narrow down the definition of sensors, the class `ssn:SensingDevice` represents the sensors which are also devices (`ssn:Device`), this means the physical sensors like MIMUs.

Not only is the class `MIMU` defined to be a subclass of the class `ssn:SensingDevice`, but also of the anonymous class `ssn:observes only MimuMagnitude`. The property `ssn:observes` links the class `ssn:Sensor` with the class `ssn:Property` and models n the SSN Ontology the property observed or measured by a sensor. `MimuMagnitude` is the subclass of the class `ssn:Property` representing the different magnitudes observed by the MIMUs and it is defined in the MIMU Magnitudes Ontology (see Section 3.1.2). An anonymous class is a class without a given name and modeled through some restrictions. In this case, a universal restriction on the property `ssn:observes` defines the anonymous class `ssn:observes`

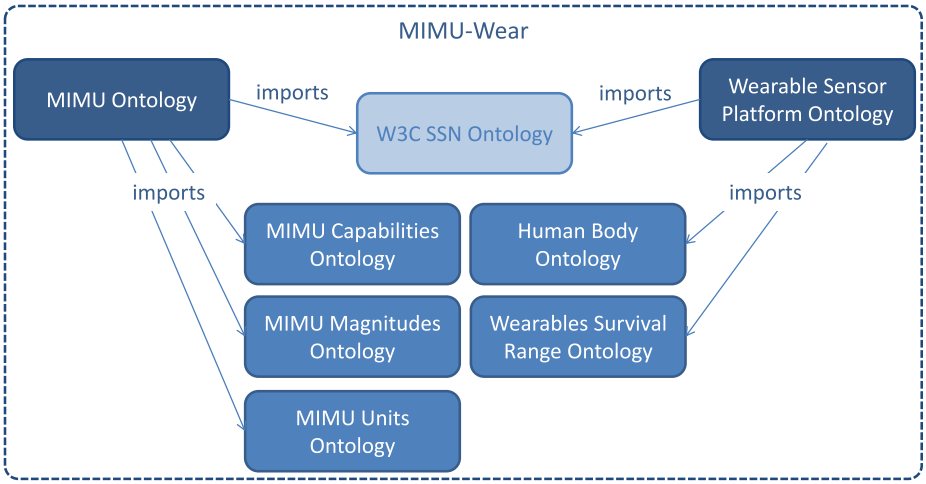

Fig. 1. Structure of the MIMU-Wear Ontology for the description of MIMU-based wearable platforms.

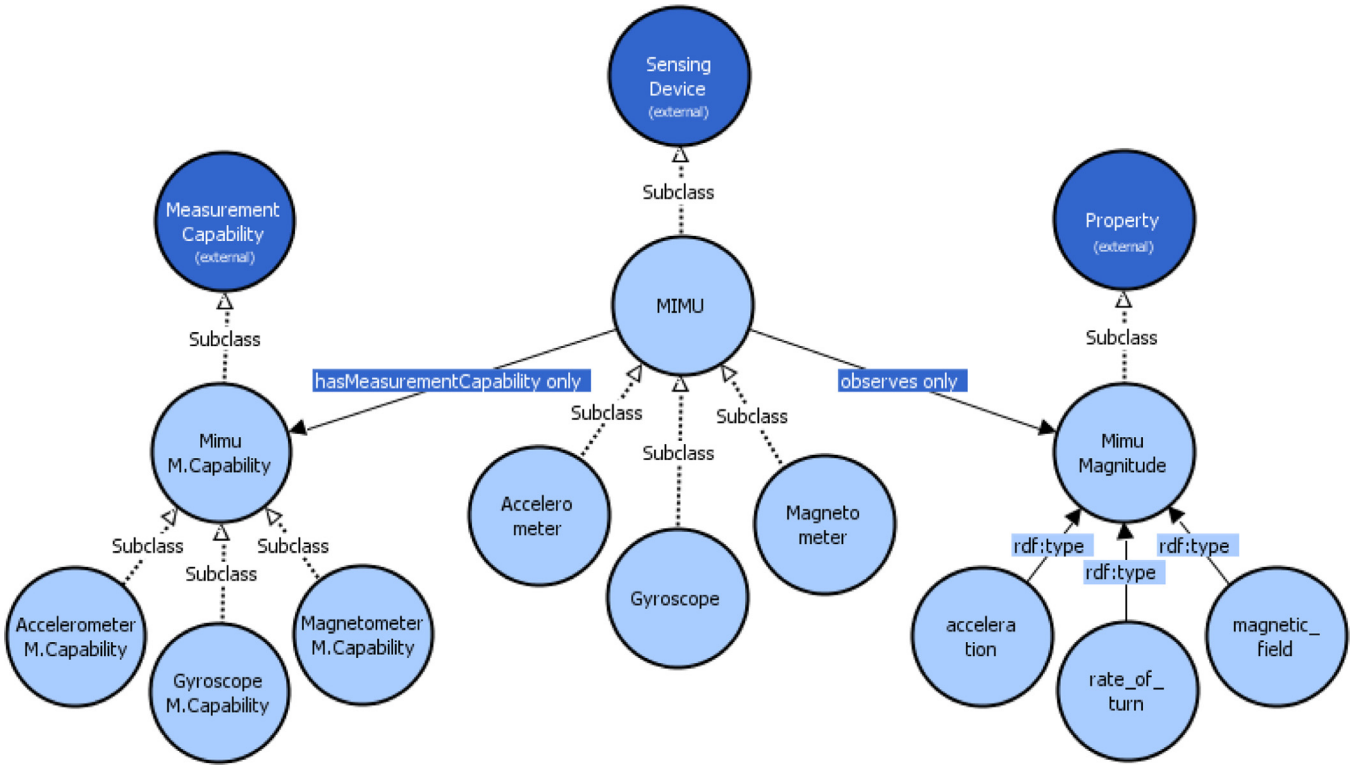

Fig. 2. MIMU Ontology: overview of the class `MIMU` and its relation to the class `MimuMeasurementCapability` and the class `MimuMagnitude`.

`only MimuMagnitude`. Universal restrictions indicate that the property can only take a set of values. For this example, the property `ssn:observes` can only take as values the members of the class `MimuMagnitude`. This restriction does not state that the property `ssn:observes` for the class `MIMU` must always be defined, but if it exists, it has to link to a member of the class `MimuMagnitude`. Conversely, existential restrictions enforce that a given property must always exist. Universal restrictions are modeled via the quantifier `owl:allValuesFrom` in OWL 2 and the quantifier `only` in protégé [37], and existential restrictions via `owl:someValuesFrom` in OWL 2 and `some` in protégé. For simplicity and since the ontology has been modeled in protégé, the simplified protégé nomenclature is used in this paper.

Completing the definition of the class `MIMU` requires modeling the relation between a MIMU and its specific sensing capabilities. In the SSN Ontology, the sensing capabilities of a sensor are represented via the class `ssn:MeasurementCapability` and linked to the sensor (`ssn:Sensor`) via the property `ssn:hasMeasurementCapability`. Thus, the class `MIMU` is defined to be a subclass of the anonymous class `ssn:has MeasurementCapability only MimuMeasurement Capability`. The class `MimuMeasurementCapability` is a subclass of the class `ssn:MeasurementCapability` defined in the MIMU Capabilities Ontology (see Section 3.1.1). From these assertions and the declared knowledge in the SSN Ontology, it can be inferred that all the members of the class `Mimu MeasurementCapability` are related via the property `ssn: forProperty` to an individual of the class `MimuMagnitude`. This means that a given set of measurement capabilities of a MIMU are applicable for the magnitude observed by the MIMU; thus, relating the measurement capabilities and the measured magnitude.

In order to model the different types of MIMUs, three disjoint subclasses of the class MIMU are defined: Accelerometer, Gyroscope and Magnetometer. These classes need to be further specified to obtain a greater level of detail by defining the anonymous classes from which they are subclasses of. The class Accelerometer is asserted to be a subclass of ssn:observes value acceleration, where acceleration is a member of the class MimuMagnitude in the MIMU Magnitudes Ontology. This means that any individual of the class Accelerometer has inferred being a subclass of the anonymous class ssn:observes value acceleration. In other words, any accelerometer is automatically defined as the MIMU which measures acceleration. Similarly, the class Gyroscope is asserted to be a subclass of ssn:observes value rate_of_turn, where rate_of_turn is a member of the class MimuMagnitude in the MIMU Magnitudes Ontology. In the same way, the class Magnetometer is asserted to be a subclass of ssn:observes value magnetic_field, where magnetic_field is a member of the class MimuMagnitude in the MIMU Magnitudes Ontology. Thus, a gyroscope is the MIMU which measures rate of turn, and a magnetometer the one which measures magnetic field. Apart from defining the restricted property values, to complete the definition of the three subclasses of the class MIMU, it is necessary to assert universal restrictions on the property ssn:hasMeasurementCapability as it is done for the class MIMU. The class Accelerometer is asserted to be a subclass of ssn:hasMeasurementCapability only AccelerometerMeasurementCapability, where AccelerometerMeasurementCapability is a subclass of the class MimuMeasurementCapability defined in the MIMU Capabilities Ontology. Similarly, the class Gyroscope is asserted to be a subclass of ssn:hasMeasurementCapability only GyroscopeMeasurementCapability and the class Magnetometer is asserted to be a subclass of ssn:hasMeasurementCapability only MagnetometerMeasurementCapability, where GyroscopeMeasurementCapability and MagnetometerMeasurementCapability are subclasses of the class MimuMeasurementCapability defined in the MIMU Capabilities Ontology. From these assertions and the declared knowledge in the SSN Ontology, it can be inferred that the class AccelerometerMeasurementCapability is related via the property ssn:forProperty to the individual acceleration, the class GyroscopeMeasurementCapability is related via the property ssn:forProperty to the individual rate_of_turn, and the class MagnetometerMeasurementCapability is related to the individual magnetic_field.

3.1.1. MIMU capabilities ontology

The MIMU Capabilities Ontology models the sensing capabilities of MIMUs. The main class of this ontology is the class MimuMeasurementCapability which is a subclass of the class ssn:MeasurementCapability and represents the measurement capabilities of a MIMU in specific conditions (see Fig. 3). A sensor might have several capability descriptions such as its accuracy or resolution, and these are modeled in the SSN Ontology through the class ssn:MeasurementProperty. Thus, each measurement capability of a MIMU is described through a set of measurement properties represented by the class MimuMeasurementProperty which is a subclass of the class ssn:MeasurementProperty. The class MimuMeasurementCapability is defined to be a subclass of the anonymous class ssn:hasMeasurementProperty only MimuMeasurementProperty. Therefore, the class MimuMeasurementCapability and the class MimuMeasurementProperty are linked via the property ssn:hasMeasurementProperty.

Using existential and universal restrictions, the class MimuMeasurementProperty is further specified for the particular measurement properties of MIMUs (see Fig. 3). The class MimuMeasurementProperty is defined to be a subclass of the anonymous classes hasQuantityValue only xsd:float, hasRangeMaxValue only xsd:float and hasRangeMinValue only xsd:float. The data properties hasQuantityValue, hasRangeMaxValue and hasRangeMinValue are functional, this means, properties that can have only one unique value. These properties take as value a xsd:float which is a datatype of the W3C XML Schema Definition Language (XSD) [38]. The universal restrictions on these properties indicate that not all these data properties are mandatory to define the MIMU measurement property, in some cases asserting the value of one of them might be enough; however, if they are asserted, they can only take as value a float. The class MimuMeasurementProperty is also defined to be a subclass of the anonymous classes hasUnit only UnitOfMeasure and hasUnit some UnitOfMeasure, where UnitOfMeasure is the main class of the MIMU Units Ontology (see Section 3.1.3), and hasUnit is a functional object property used to define the units in which the value of the specific measurement property is represented. In this case, the existential and universal restrictions on the functional object property hasUnit indicate that this property needs to be always asserted and to take a single value of the class UnitOfMeasure.

This generic definition of the class MimuMeasurementProperty is not enough. Therefore, the subclasses of the class ssn:MeasurementProperty are further specified to define in detail the most common measurement properties of the MIMUs. Particularly, the disjoint classes MimuMeasurementRange, MimuSensitivity, MimuResolution, MimuFrequency, MimuDrift and MimuNoise are here defined. For each of these classes, existential restrictions are asserted for the properties hasQuantityValue, hasRangeMaxValue and hasRangeMinValue, and restricted property values are asserted for the property hasUnit.

The class MimuMeasurementRange is the subclass of the class ssn:MeasurementRange and the class MimuMeasurementProperty which particularizes the concept of measurement range for the case of MIMUs. The measurement range of a MIMU is defined as the set of values comprised between an upper limit and a lower limit which can be measured by the MIMU. Therefore, the measurement range is described as a pair of values, i.e., the maximum value and the minimum value of the interval in which the MIMU can measure. In order to model these two values, the class MimuMeasurementRange is defined to be a subclass of the anonymous classes hasRangeMaxValue some xsd:float and hasRangeMinValue some xsd:float. The values defining the measurement range are provided in the appropriate units for each of the types of MIMUs (m/s^2 for the accelerometer, deg/s for the gyroscope, and gauss for the magnetometer). Therefore, the class MimuMeasurementRange is defined to have three disjoint subclasses: the class AccelerometerMeasurementRange, the class GyroscopeMeasurementRange, and the class MagnetometerMeasurementRange. These classes model the different measurement ranges for each of the types of MIMUs and define the corresponding units for each of them. The class AccelerometerMeasurementRange is asserted to be a subclass of the anonymous class hasUnit value m_per_square_s (see Fig. 4). The class GyroscopeMeasurementRange is asserted to be a subclass of the anonymous class hasUnit value deg_per_s. The class MagnetometerMeasurementRange is asserted to be a subclass of the anonymous class hasUnit value gauss.

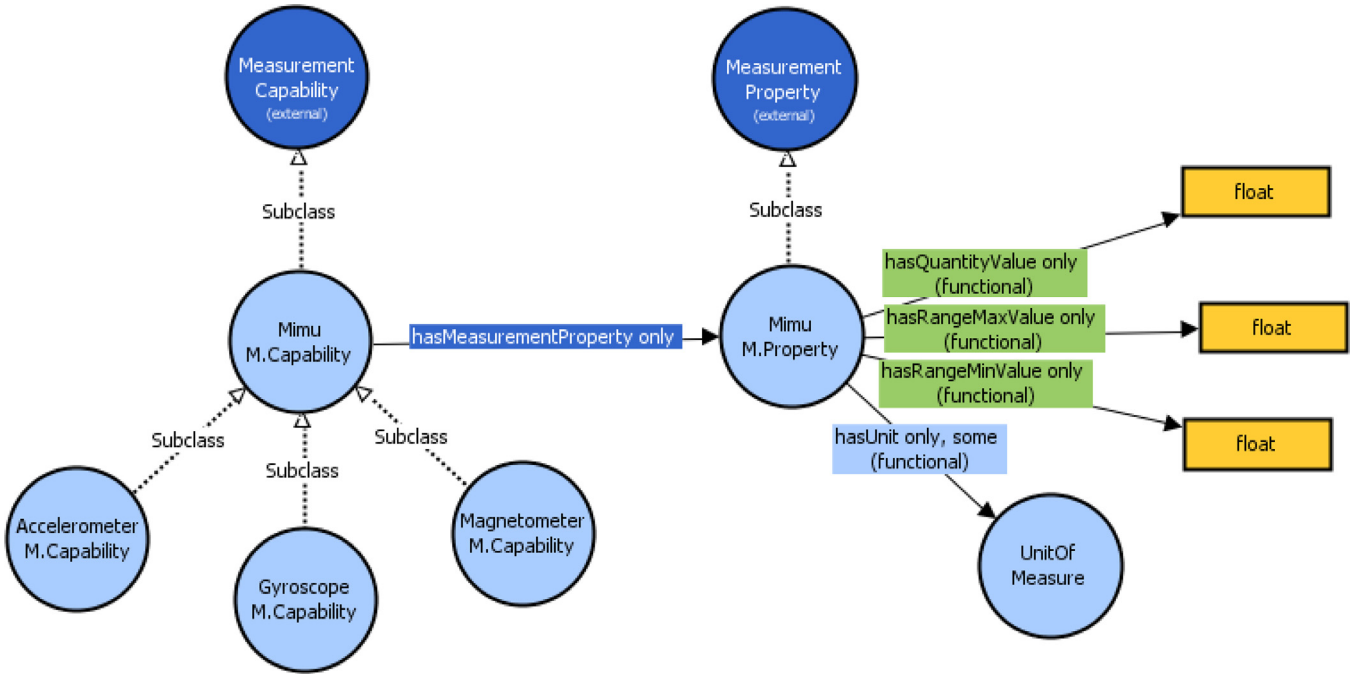

Fig. 3. MIMU Capabilities Ontology: overview of the class `MimuMeasurementCapability` and the class `MimuMeasurementProperty`, and the relation between them.

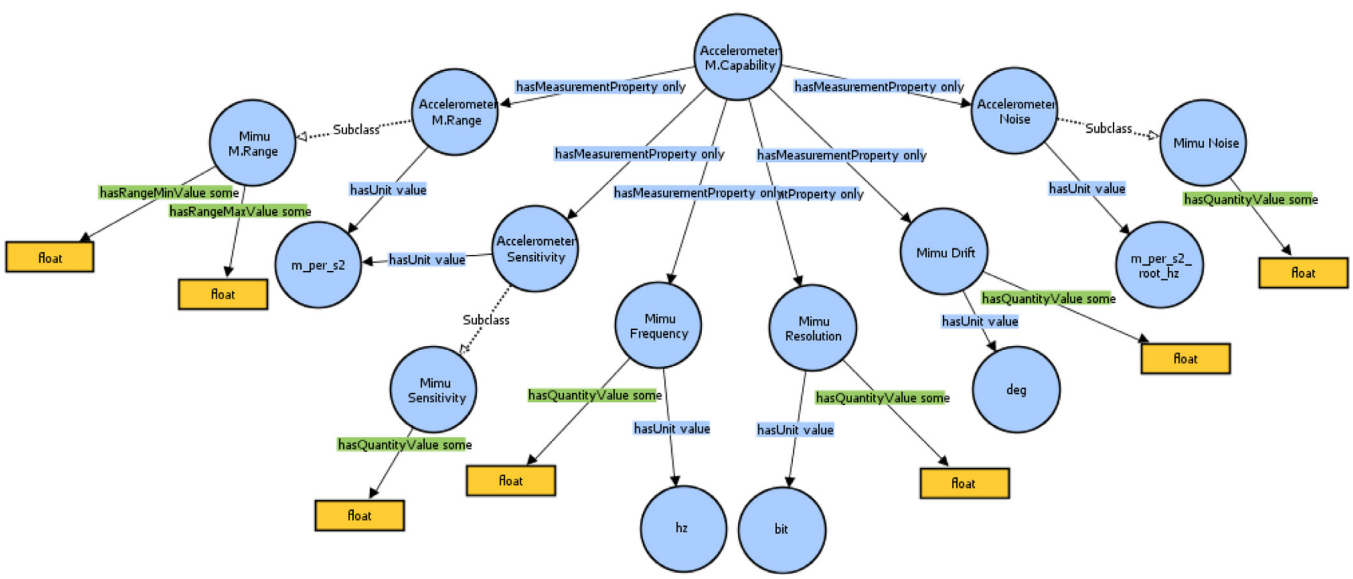

Fig. 4. MIMU Capabilities Ontology: Overview of the class `AccelerometerMeasurementCapability` and the subclasses of the class `MimuMeasurementProperty` which define it.

The class `MimuSensitivity` is the subclass of the class `ssn:Sensitivity` and the class `MimuMeasurement Property` which particularizes the concept of sensitivity for the case of MIMUs. The sensitivity of a MIMU, also known as linearity, measures the calibration of the MIMU, and is here modeled as its value in the measurement unit. Thus, the class `MimuSensitivity` is defined to be a subclass of the anonymous class `hasQuantityValue some xsd:float`, and this value of the sensitivity is provided in the appropriate units for each type of MIMU. Therefore, the class `MimuSensitivity` is defined to have three disjoint subclasses: the class `AccelerometerSensitivity`, the class `GyroscopeSensitivity`, and the class `Magnetometer Sensitivity`. The class `AccelerometerSensitivity` is defined to be a subclass of the anonymous class `hasUnit value m_per_square_s` (see Fig. 4), the class `Gyroscope Sensitivity` of the class `hasUnit value deg_per_s`, and the class `MagnetometerSensitivity` of the class `hasUnit value gauss`.

The class `MimuResolution` is the subclass of the class `ssn:Resolution` and the class `MimuMeasurementProperty` which particularizes the concept of resolution for the case of MIMUs. The numeric resolution of a MIMU is defined as the number of bits that represent the measurement of the MIMU. Thus, the class `MimuResolution` is defined to be a subclass of the anonymous classes `hasQuantityValue some xsd:float` and `hasUnit value bit` (see Fig. 4).

The class `MimuFrequency` is the subclass of the class `ssn:Frequency` and the class `MimuMeasurementProperty` which particularizes the concept of frequency for the case of MIMUs. The frequency of a MIMU is defined as the rate in which the measurements are executed and is represented as a value in

Hz. Thus, the class MimuFrequency is defined to be a subclass of the anonymous classes hasQuantityValue some xsd:float and hasUnit value hz (see Fig. 4).

The class MimuDrift is the subclass of the class ssn:Drift and the class MimuMeasurementProperty which particularizes the concept of drift for the case of MIMUs. The alignment error which appears on the data sheets of MIMUs could be interpreted as a drift and measures the misalignment between the axes of the MIMUs. This alignment error is represented as a value in degrees. Thus, the class MimuDrift is defined to be a subclass of the anonymous classes hasQuantityValue some xsd:float and hasUnit value deg (see Fig. 4).

The class MimuNoise is the subclass of the class MimuMeasurementProperty which represents the noise of the MIMU. This is an important measurement property of the MIMUs which is not part of the SSN Ontology. The class MimuNoise is defined to be a subclass of the anonymous class hasQuantityValue some xsd:float. The value of the density of noise is provided in the appropriate units for each type of MIMU ($m/s^2/\sqrt{Hz}$ for the accelerometer, $deg/s/\sqrt{Hz}$ for the gyroscope, and $gauss/\sqrt{Hz}$ for the magnetometer). Therefore, the class MimuNoise is defined to have three disjoint subclasses: the class AccelerometerNoise, the class GyroscopeNoise and the class MagnetometerNoise. The class AccelerometerNoise is asserted to be a subclass of the anonymous class hasUnit value m_per_square_s_and_root_hz (see Fig. 4). The class GyroscopeNoise is asserted to be a subclass of the anonymous class hasUnit value deg_per_s_and_root_hz. The class MagnetometerNoise is asserted to be a subclass of the anonymous class hasUnit value gauss_per_root_hz.

Finally, to conclude with the modeling of the measurement capabilities and the measurement properties of the MIMUs, the class MimuMeasurementCapability needs to be linked to the subclasses of the class MimuMeasurementProperty, namely MimuMeasurementRange, MimuSensitivity, MimuResolution, MimuDrift, MimuFrequency and MimuNoise. In fact, the class MimuMeasurementCapability and the class MimuMeasurementProperty are already linked via the property ssn:hasMeasurementProperty. However, some of the subclasses of the class MimuMeasurementProperty are more particular and only apply to some specific types of MIMU. For this reason, the classes AccelerometerMeasurementCapability, GyroscopeMeasurementCapability and MagnetometerMeasurementCapability, which are the three disjoint subclasses of the class MimuMeasurementCapability, are created to define the capabilities of the different types of MIMUs. The class AccelerometerMeasurementCapability is asserted to be a subclass of the anonymous class ssn:hasMeasurementProperty only (AccelerometerMeasurementRange or AccelerometerSensitivity or MimuResolution or MimuFrequency or MimuDrift or AccelerometerNoise) (see Fig. 4). Similarly, the class GyroscopeMeasurementCapability is asserted to be a subclass of the anonymous class ssn:hasMeasurementProperty only (GyroscopeMeasurementRange or GyroscopeSensitivity or MimuResolution or MimuFrequency or MimuDrift or GyroscopeNoise), and the class MagnetometerMeasurementCapability is asserted to be a subclass of the anonymous class ssn:hasMeasurementProperty only (MagnetometerMeasurementRange or MagnetometerSensitivity or MimuResolution or MimuFrequency or MimuDrift or MagnetometerNoise).

3.1.2. MIMU magnitudes ontology

The class MimuMagnitude is the main class of the MIMU Magnitudes Ontology and represents the different magnitudes or physical properties that can be observed by a MIMU. For the class MimuMagnitude three different individuals are defined: acceleration, rate_of_turn, and magnetic_field. The name of the individuals indicate the magnitude the MIMU is able to measure. These members are asserted to be different one from each other since they represent different concepts.

In this work, the magnitudes measured by the MIMU have been defined in a simple domain ontology. The MIMU Ontology could be easily extended to include any available ontology which describes magnitudes. For example, the MyMobileWeb Measurement Units Ontology (MUO) could be used to represent the acceleration and the individual muo:acceleration of the class muo:PhysicalQuality would be the equivalent to the presented individual acceleration. If multiple magnitude ontologies would be used for the same scenario description, ontology matching would be required to map the concepts of different domain ontologies into the proposed MIMU Magnitudes Ontology.

3.1.3. MIMU units ontology

The class UnitOfMeasure is the main class of the MIMU Units Ontology and represents the different measurement units required to describe the capabilities of a MIMU. For the class UnitOfMeasure several individuals are defined: m_per_square_s, gauss, deg_per_s, hz, bit, deg, m_per_square_s_and_root_hz, gauss_per_root_hz, deg_per_s_and_root_hz. All these individuals are asserted to be different from each other since they represent different concepts. The name of the members of the class UnitOfMeasure indicate the name of the unit of the International System of Units which they represent.

The units modeled in the MIMU Units Ontology and used in the MIMU Capabilities Ontology are the only ones required for this specific domain. However, this simple ontology could be extended in the future to include other unit systems. The extension of this ontology would imply creating new subclasses of the class UnitOfMeasure and establishing the conversion between the different measurement systems and units. Moreover, external ontologies like the MyMobileWeb Measurement Units Ontology (MUO) or the SysML-QUDV could be plugged into the MIMU Ontology to describe the units. In the case of coexisting more than one units ontology, the concepts should be matched into the proposed MIMU Units Ontology.

3.2. Wearable sensor platform ontology

The Wearable Sensor Platform Ontology models the characteristics of wearable sensor platforms. In order to describe the survival conditions of wearable systems and the localization of the wearable sensor platform on the body of the user, the SSN Ontology is here extended. The Wearable Sensor Platform Ontology neatly defines vocabularies to model the survival range of the wearables and their locations, which are not part of the SSN Ontology.

The main class of the Wearable Sensor Platform Ontology is the class WearableSensorPlatform (see Fig. 5). This class is a subclass of the class ssn:Platform and particularizes the concept of platform for the case of wearable sensor platforms. The platform (ssn:Platform) as described in the SSN Ontology is the entity that hosts a system (ssn:System), and a system is any part of the sensing infrastructure. In other words, the system may be mounted or deployed on a platform, here the entity to which the system is attached. For example, a bracelet that tracks the user activity would be the platform into which the sensing system composed of some accelerometers is embedded. The wearable system is modeled through the class WearableSystem which is the subclass of the class ssn:System and of the anonymous class ssn:onPlatform only WearableSensorPlatform.

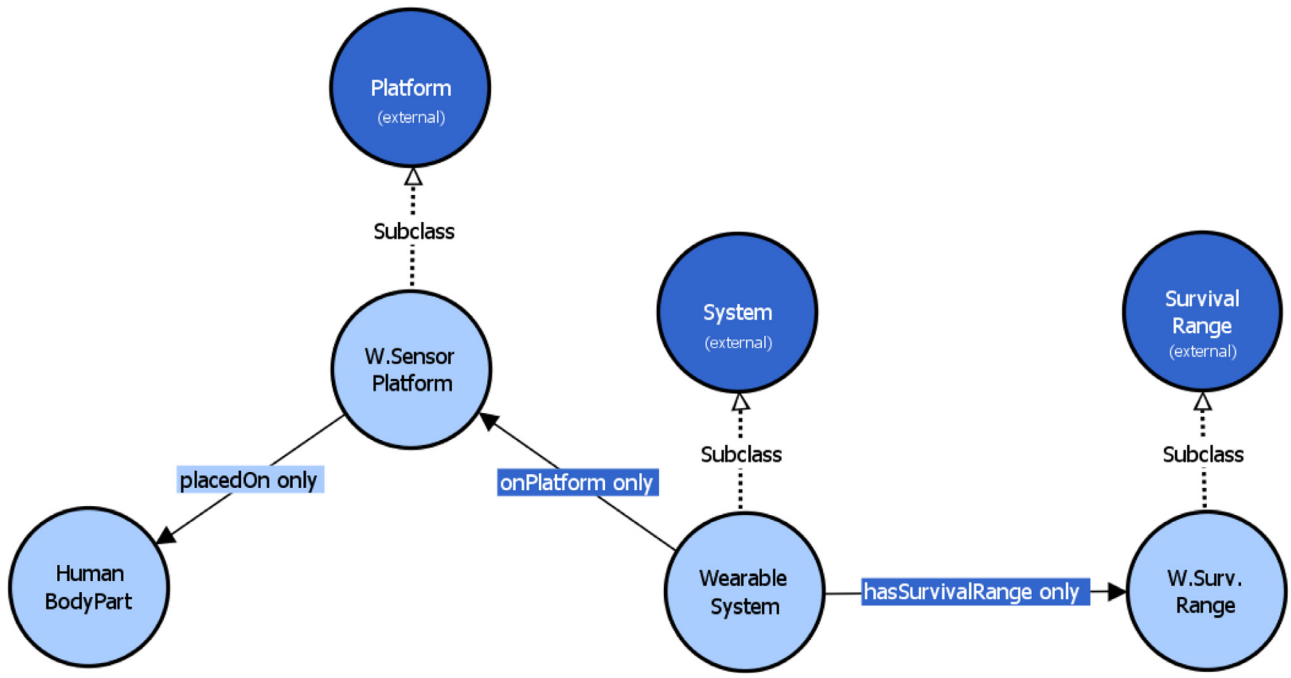

Fig. 5. Wearable Sensor Platform Ontology: overview of the class `WearableSensorPlatform` and its relation to the class `WearableSystem` and the class `HumanBodyPart`.

One of the most important characteristics of wearable sensor platforms is that they are worn or located on the body of the user. Thus, the class `WearableSensorPlatform` is asserted to be a subclass of the anonymous class `placedOn only HumanBodyPart`, where the property `placedOn` is used to define the spatial attributes of the wearable platform and the class `HumanBodyPart` is the main class of the Human Body Ontology (see Section 3.2.1).

Finally, to represent the survival conditions of a wearable system such as its battery lifetime, the class `WearableSystem` is declared to be a subclass of the anonymous class `ssn:hasSurvivalRange only WearableSurvivalRange`. The class `WearableSurvivalRange` is the subclass of the class `ssn:SurvivalRange` which describes the survival conditions of wearables and is defined in the Wearables Survival Range Ontology (see Section 3.2.2). The property `ssn:hasSurvivalRange` links the survival conditions to the system.

3.2.1. Human body ontology

The Human Body Ontology models the human body parts representing the potential locations where the wearable sensor platforms are worn. The main class of the Human Body Ontology is the class `HumanBodyPart` which represents each one of the body parts. The main division of the body is done in four parts: head, trunk, upper limbs and lower limbs. Therefore, four classes are defined as subclasses of the class `HumanBodyPart`: `Head`, `Trunk`, `UpperLimb` and `LowerLimb`. Moreover, each of the main body parts can be further partitioned into subdivisions, which are also parts of the human body and therefore subclasses of the class `HumanBodyPart`. The class `HeadSubdivision` has been specified to define the subdivisions of the head: face and scalp. The class `TrunkSubdivision` has been specified to define the subdivisions of the trunk: thorax, abdomen and back. The class `UpperLimbSubdivision` has been specified to define the subdivisions of the upper limbs: shoulder, arm, elbow, forearm, wrist, and hand. The class `LowerLimbSubdivision` has been specified to define the subdivisions of the lower limbs: hip, thigh, knee, leg, ankle, and foot.

In order to set the links between each of the main body parts and their corresponding subdivisions, the object property `hasPart` has been defined as well as its inverse property `partOf` which relates the subdivisions to their containing main body part. The link between the class `HeadSubdivision` and the class `Head` is created by using the property `partOf` and asserting that the class `HeadSubdivision` is a subclass of the anonymous class `partOf only Head`. From this assertion, it can be inferred that the inverse property `hasPart` links the class `Head` to the class `HeadSubdivision`, i.e., the class `Head` is a subclass of the anonymous class `hasPart only HeadSubdivision`. Moreover, it can also be inferred that the class `Face` and the class `Scalp`, which are subclasses from the class `HeadSubdivision`, are also subclasses of the anonymous class `partOf some Head`. Finally, cardinality restrictions have been asserted to complete the definition of the relation between the main body parts and their subdivisions. Cardinality restrictions are used to constrain the number of values of a particular property, for example, a head has exactly one face. Therefore, the class `Head` has been defined as being a subclass of the anonymous class `(hasPart exactly 1 Face) and (hasPart exactly 1 Scalp)` (see Fig. 6). The relations between the rest of body parts and their subdivisions have been established using the same modeling principle (see Figs. 7–9).

Not only are the different body parts subdivided in a hierarchical manner, they are also connected to other parts. Several object properties have been defined in the Human Body Ontology to describe the connections among body parts and their subdivisions. The top property is `connectedTo` and it has eight subproperties defining the connections of body parts according to the standard human directional terms: superior or inferior, anterior or posterior, medial or lateral, proximal or distal. The property `superiorlyConnectedTo` relates a body part with another one located towards the top of the body or human head, and has as inverse the property `inferiorlyConnectedTo`. The property `anteriorlyConnectedTo` relates a body part with another one located towards the front of the body, and has as inverse the property `posteriorlyConnectedTo`. The

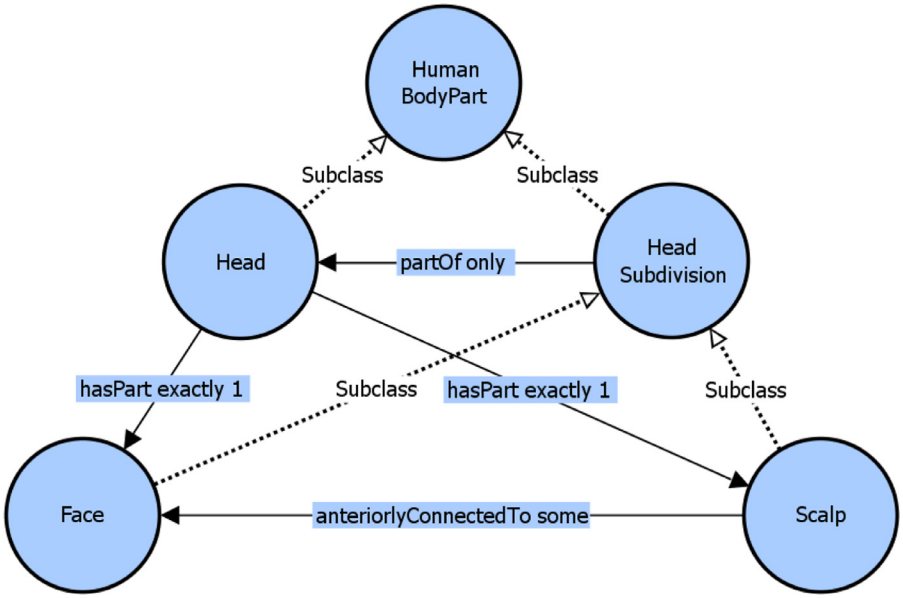

Fig. 6. Human Body Ontology: overview of the class `Head` and the class `HeadSubdivision`.

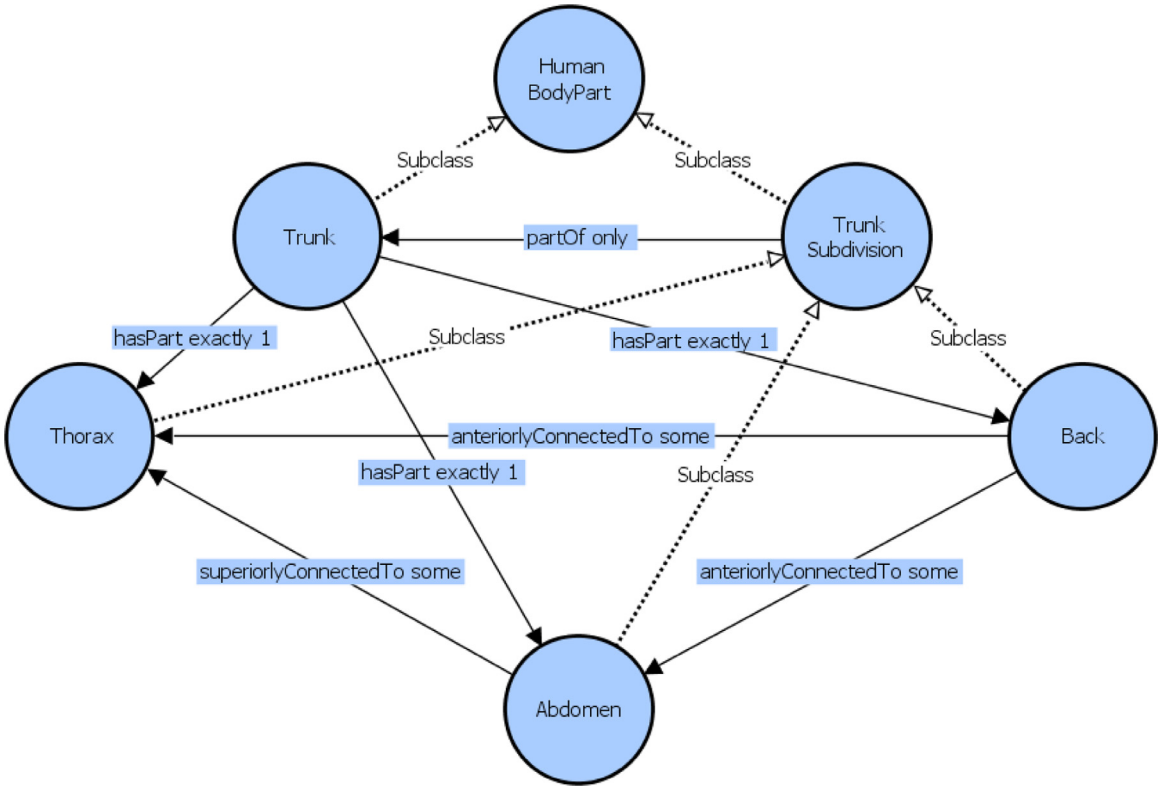

Fig. 7. Human Body Ontology: overview of the class `Trunk` and the class `TrunkSubdivision`.

property `laterallyConnectedTo` relates a body part with another one located towards the lateral of the body, and has as inverse the property `mediallyConnectedTo`. The property `proximallyConnectedTo` relates a body part with another one located towards the main mass of the body, and has as inverse the property `distallyConnectedTo`.

To complete the ontology definition, the connections among the body parts need to be established using the subproperties of `connectedTo`. For example, in the case of the trunk (see Fig. 7), this is modeled via the class `Trunk` and it has three subdivisions represented through the classes `Thorax`, `Abdomen` and `Back`. The thorax and the abdomen conform the anterior part of the trunk and the back the posterior part of it. Therefore, the class `Back` is defined to be a subclass of the anonymous class `anteriorlyConnectedTo some Thorax` and the anonymous class `anteriorlyConnectedTo some Abdomen`. The connection between the class `Thorax` and the class `Back` can be directly inferred from the inverse properties. Thus, the class `Thorax` is inferred to be a subclass of the anonymous class `posteriorlyConnectedTo some Back`.

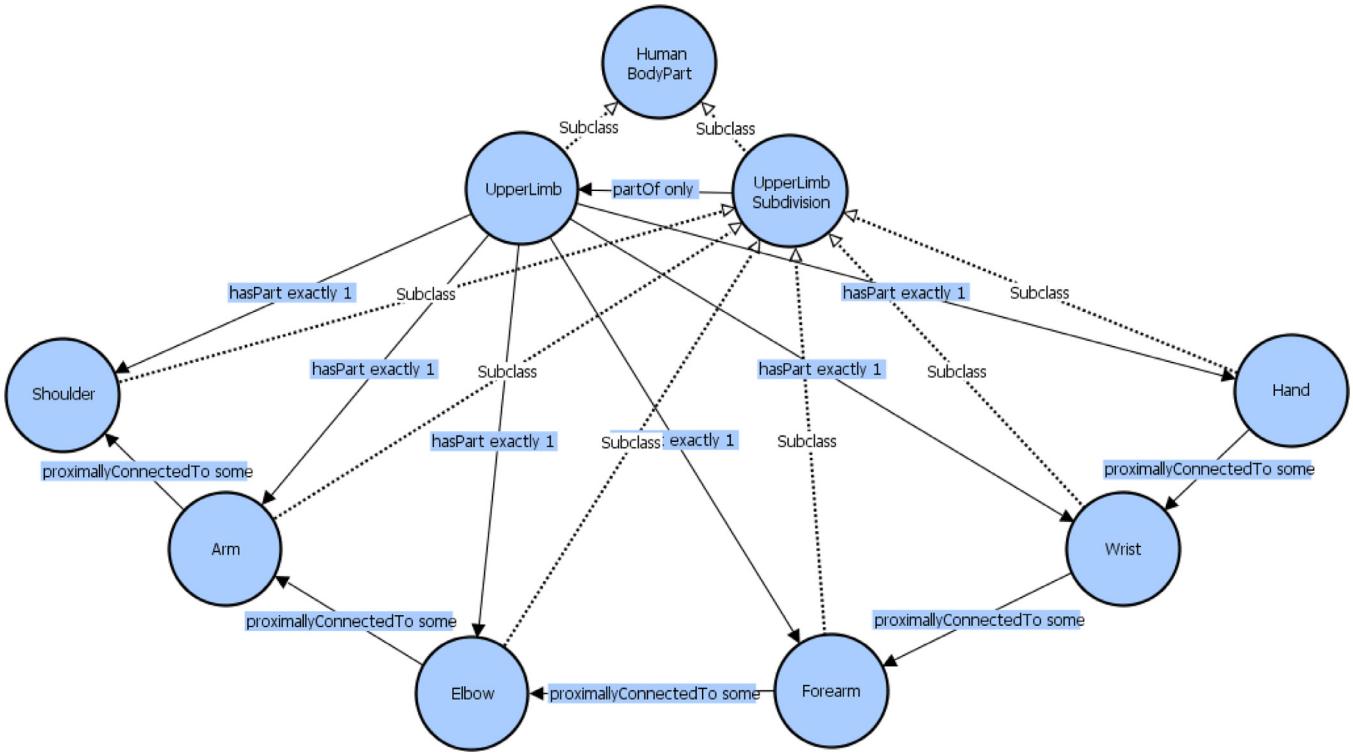

Fig. 8. Human Body Ontology: overview of the class `UpperLimb` and the class `UpperLimbSubdivision`.

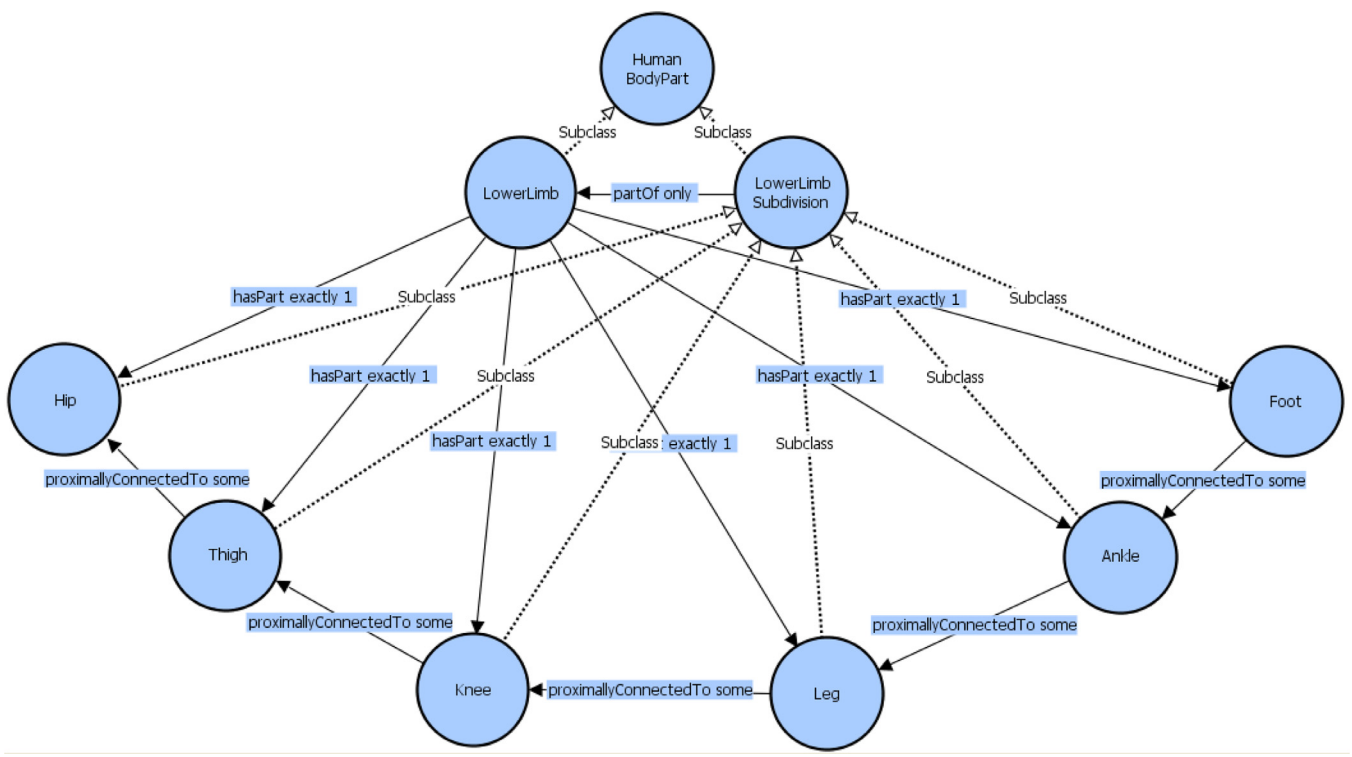

Fig. 9. Human Body Ontology: overview of the class `LowerLimb` and the class `LowerLimbSubdivision`.

Similarly, the class `Abdomen` is inferred to be as a subclass of the anonymous class `posteriorlyConnectedTo some Back`. Moreover, the thorax is located on top of the abdomen, and the class `Abdomen` is asserted to be a subclass of the anonymous class `superiorlyConnectedTo some Thorax`. The corresponding inverse links can be directly inferred. Then, the class `Thorax` is inferred to be a subclass of the anonymous class `inferiorlyConnectedTo some Abdomen`. The connections between the subdivisions of the head are established through the same subproperties of `connectedTo` (see Fig. 6). In the case of the upper and lower limbs, the same modeling principle applies but the property `proximallyConnectedTo` and its in-

verse distallyConnectedTo are used to establish the connections (see Fig. 8 and Fig. 9). For example, the Forearm is a subclass of the anonymous class proximallyConnectedTo some Elbow, since the forearm is more distant from the trunk than the elbow. The connections between the main body parts can also be established through the eight subproperties of the property connectedTo. The main difference with the previous examples is the usage of the property lateralyConnectedTo. The trunk is in the middle of the body and the upper limbs are in a lateral position from the trunk. Thus, the connection between the class Trunk and the class UpperLimb is created by using the property lateralyConnectedTo and defining the class Trunk as a subclass of the anonymous class lateralyConnectedTo some UpperLimb.

The property symmetricTo is used to model the relations between body parts symmetrically located on the human body. For example, the individual user_left_upperlimb of the class UpperLimb is related to the individual user_right_upperlimb of the class UpperLimb via a the property symmetricTo.

The Human Body Ontology models the human body so that the location of a wearable sensor platform on specific body parts can be exhaustively described. The Wearable Sensor Platform Ontology could be easily extended to include any available ontology modeling the human body. Two possible candidates are the Foundational Model of Anatomy ontology (FMA) [39], one of the most complete knowledge source for bioinformatics which represents the phenotypic structure of the human body, and the Uber anatomy ontology (Uberon) [40], an anatomy ontology that integrates any type of animal species. These ontologies are much more extensive than the Human Body Ontology and cover many concepts which are not required to model the location of wearable sensor platforms on the human body. However, the appropriate concepts could be mapped into the Human Body Ontology to enable their coexistance in the Wearable Sensor Platform Ontology.

The Human Body Ontology is extensible and new concepts can be added in a simple fashion. In future versions of this ontology, some characteristics of the body parts could extend the definition of the class HumanBodyPart. These new concepts would be directly inherited by all the subclasses representing the different body parts, such as the UpperLimb or Back. One example of the new characteristics that could be modeled in the Human Body Ontology is the mobility level of a body part. A system taking into account possible injuries of the user in the selection of the wearable sensor platforms would require this information. In such a scenario, marking the injured body parts and with reduced mobility would be relevant to avoid selecting a replacement sensor platform worn on these body parts.

3.2.2. Wearables survival range ontology

The Wearables Survival Range Ontology models the survival conditions of a wearable system. The main class of this ontology is the class WearableSurvivalRange which is a subclass of the class ssn:SurvivalRange and represents the survival range of wearable systems (see Fig. 10). The survival conditions of a wearable system are described through a set of survival properties represented by the class WearableSurvivalProperty, which is a subclass of the class ssn:SurvivalProperty. Moreover, the class WearableSurvivalRange is declared to be a subclass of the anonymous class ssn:hasSurvivalProperty only WearableSurvivalProperty.

The class WearableSurvivalProperty is further specified to model the most common survival properties of the wearable systems. Particularly, the class WearableBatteryLifetime is defined to represent the lifetime of the battery in a wearable system. The class WearableBatteryLifetime is a subclass of the class WearableSurvivalProperty and the class ssn:BatteryLifetime for which some restrictions are asserted. The class WearableBatteryLifetime is a subclass of the anonymous classes hasQuantityValue some xsd:float and hasQuantityValue only xsd:float, where hasQuantityValue is a functional data property, and xsd:float is the datatype of the W3C XML Schema Definition Language (XSD) [38]. These universal and existential restrictions on the property hasQuantityValue indicate that there must be a value for this property and it has to be of type float. Moreover, the class WearableBatteryLifetime is also asserted to be a subclass of the anonymous class hasUnit value s, where hasUnit is a functional object property used to define the units in which the battery lifetime is measured, and s is the individual of the class UnitOfMeasure which represents the seconds.

The Wearables Survival Range Ontology only defines the lifetime of the battery as a property of the wearable system. However, this ontology could be easily extended in the future to model more survival properties of the wearable systems, such as memory resources or processing power. These properties are certainly important for the self-configuration of the wearable sensor system at runtime. In this scenario, sensors need to be associated to a wearable systems and knowing the system memory is crucial to ensure that the sensors can be supported. Thus, in the future the class WearableSurvivalProperty could be subclassed to model these concepts.

3.3. Description of MIMU-based wearable platforms using MIMU-Wear

The presented MIMU-Wear Ontology models the basic concepts to describe wearable sensor platforms consisting of MIMUs. However, in order to describe a specific model of a MIMU or a precise wearable sensor platform, the definition of some more restrictive classes is required. Moreover, to describe the particular operation mode of a MIMU or the location of wearable sensor platform, instances of the classes described in the MIMU-Wear Ontology need to be generated. In the following, the use of the MIMU-Wear Ontology to describe MIMU-based sensor platforms is presented. An example of its application in the description of a real scenario is analyzed in Section 5.2.

To represent a specific model and brand of MIMU, the threes subclasses of the class MIMU - Accelerometer, Gyroscope and Magnetometer - can be further subclassed in order to group the MIMUs with common properties. In the definition of a particular model of MIMU there is no need to assert the value of the property ssn:observes since this is directly inferred from its superclass. However, further definition of the subclasses of the class MimuMeasurementCapability is required to model the MIMU measurement capabilities. For example, an accelerometer of a precise brand and model is described as a subclass of the class Accelerometer. The axiom ssn:observes value acceleration is directly inferred for this subclass. This means that the class grouping these type of accelerometers is inferred to be as a subclass of the MIMUs which measure the acceleration magnitude. However, in the description of this accelerometer is necessary to define its specific measurement range as one of its capabilities. To model the measurement range as a capability of the accelerometer, a subclass of the class AccelerometerMeasurementCapability should be created and linked to a new member of the class AccelerometerMeasurementRange via the property ssn:hasMeasurementProperty. For the member of the class AccelerometerMeasurementRange, the values of the data properties hasMaxValue and hasMinValue should be asserted to indicate the value in m/s^2 that define the

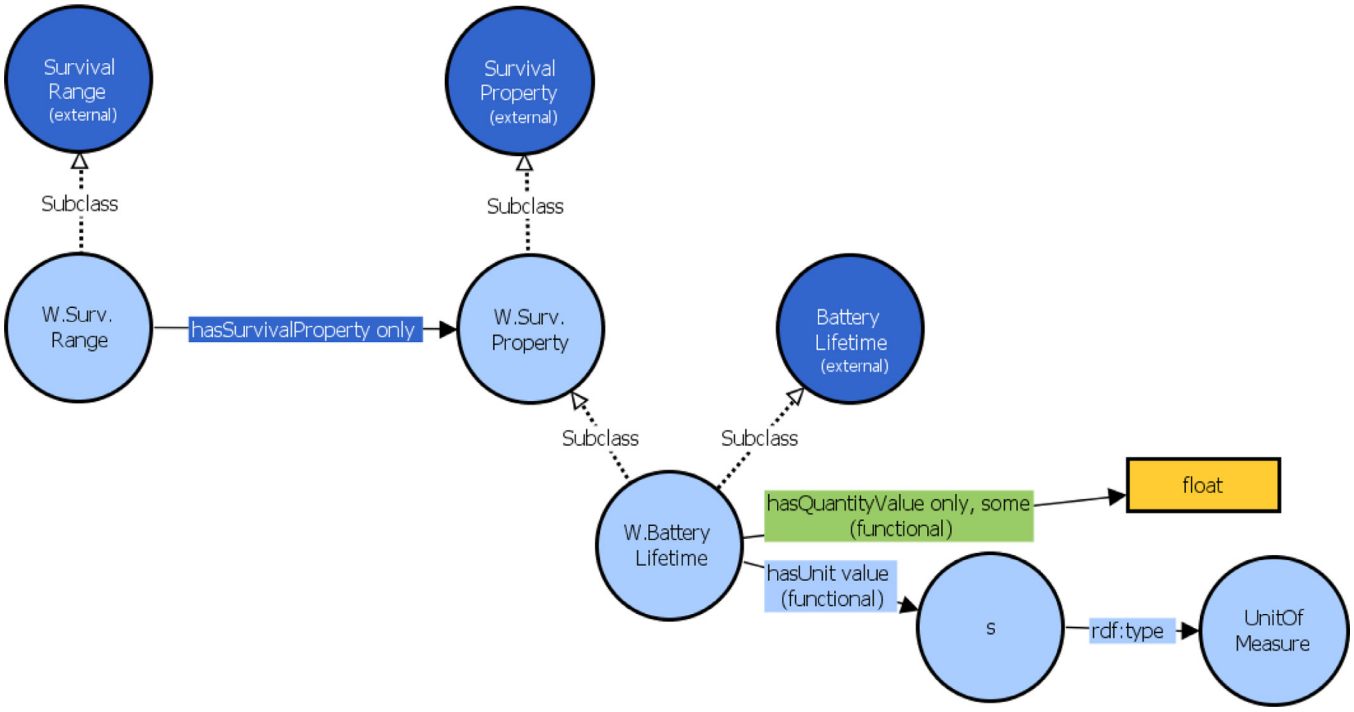

Fig. 10. Wearables Survival Range Ontology: overview of the class `WearableSurvivalRange` and the class `WearableSurvivalProperty`.

interval of the accelerometer dynamic range. The units do not need to be asserted since the axiom `hasUnit value m_per_square_s` is directly inferred from the superclass `AccelerometerMeasurementRange`.

The individuals of the class `MIMU` and the individuals of its subclasses represent the particular MIMUs. The description of a precise MIMU is created by defining the corresponding individual of the appropriate subclass of the class `MIMU`. Several general axioms defining the MIMU are already inferred from the class definition when creating the individual. However, the capabilities of the MIMU for the current working mode need to be specified. This means, asserting the specific value taken by the property `hasMeasurementCapability`.

In a working scenario, it could be the MIMU vendor the one that provides the ontological description of their particular model of MIMU. This means that the vendor provides the description of the appropriate subclass of the class `MIMU - Accelerometer`, `Gyroscope` or `Magnetometer` -. The MIMU description would be created using the MIMU Ontology part of MIMU-Wear in an approach based on Linked Data [41]. This would enable to create the MIMU descriptions in a distributed fashion, make them available to the public and reuse them in different scenarios. Moreover, it would be quite efficient to generate the descriptions for any specific MIMU by only creating an individual of the class representing the particular model of MIMU and which has already been provided by the vendor.

Similarly, the wearable sensor platforms can be further defined to represent specific models. In this case, a subclass of the class `WearableSensorPlatform` is created and linked to the corresponding new subclass of the class `WearableSystem`. For a specific model and brand of wearable sensor system, universal restrictions and cardinality restrictions on the property `ssn:hasSubsystem` have to be asserted to link to the particular models of MIMU integrated in the wearable platform. These assertions relate the subclasses of the class `WearableSensorPlatform` in the Wearable Sensor Platform Ontology with the subclasses of the class `MIMU` in the MIMU Ontology; thus, linking the two main parts of MIMU-Wear. The vendor of the wearable sensor platform could create the ontological description of their particular model of wearable and make it publicly available. In fact, this vendor could directly use the description provided by the vendor of the MIMU and particularize it for the specific setup in a strategy based on the Linked Data concept.

The individuals of the class `WearableSensorPlatform` and the individuals of its subclasses represent the specific wearable sensor platforms. Therefore, a description of a specific wearable sensor platforms is created by defining the corresponding individual of the appropriate subclass of the class `WearableSensorPlatform`. The user of a wearable sensor platform would be able to generate the description of their particular wearable device by simply creating an instance of the class describing the wearable which is provided by the vendor. The description of the wearable should be completed for the deployment scenario, including the location of the sensor. This means that the property `placedOn` which links the class `WearableSensorPlatform` with the class `HumanBodyPart`, should be asserted for the specific wearable. For example, in the case of a bracelet, the value of the property `placedOn` would be an individual of the class `Wrist`, particularly the one representing the wrist of the user wearing the bracelet. In order to link a wearable sensor platform to a particular body part, the model of the body of the user should be previously created. This means defining the individuals for the different body parts of the user, so that they could be integrated in the description of the wearable.

The deployment is a dynamic condition of the wearable sensor platform. The user can wear on the wearable, or take it off and leave it resting aside. Therefore, the property `placedOn` in the description of the wearable should be updated every time there is a change to reflect the real time situation.

4. A method for sensor selection based on MIMU-Wear

This section presents a novel sensor selection method to enable the replacement of anomalous MIMU sensors in wearable activity recognizers whenever a sensor is detected to have suffered some anomaly. The goal of the sensor selection method is to ensure con-

tinuity in the performance of the wearable activity recognizer. Due to the failure or the fault of one of the MIMU sensors, the performance of the wearable activity recognizer might decrease. Using a replacement sensor which can provide the activity recognizer with a similar sensing functionality, the performance of the wearable activity recognizer could in principle get restored to its original value.

The proposed sensor selection method is based on the MIMU-Wear Ontology and requires that all the available MIMU-based wearable sensor platforms are described using this ontology. Several rules are established in order to define the candidate replacement MIMU sensors to be used in the wearable activity recognizer (see Section 4.1). The rules might depend on the application scenario and need to be particularized and prioritized for each specific case. The combination of the rules and the MIMU-Wear Ontology provides the inference features required to determine the replacement sensors. Posing the adequate queries on the descriptions of the available MIMU-based wearable sensor platforms will allow selecting the best MIMU sensors which could replace the ones suffering anomalies in a wearable activity recognizer (see Section 4.2).

4.1. Rules for candidate sensor replacements

Rules are required to determine the MIMU sensors which are possible candidates for the replacement of an anomalous sensor in a wearable activity recognizer. In the proposed sensor selection method, SWRL rules are utilized to define the candidate replacement sensors. SWRL [42] is characterized to integrate with OWL 2 and therefore, it benefits from the full potential of ontological reasoning offered by OWL 2.

The rules defined to determine the candidate replacement sensors build on the MIMU-Wear Ontology and utilize the concepts represented in it. In order to model the relation of one sensor with its potential candidate replacement ones, the object property `hasReplacement` has been defined. This property links two individuals of the class `MIMU`; thus, it has the class `MIMU` as part of its domain and its range. Moreover, several subproperties of the property `hasReplacement` have been defined to particularize the conditions in which the candidate sensor has been proposed for replacement. The name of the properties are self-explanatory and describe the characteristics of the candidate replacement sensor. For example, the property `hasReplacementSameType` is utilized to link a MIMU sensor with another one which is a candidate replacement since it measures the same type of magnitude. This means that an accelerometer is proposed as candidate using the property `hasReplacementSameType`, if the faulty sensor is an accelerometer, or a gyroscope in the case of an anomalous gyroscope, or a magnetometer in the case of a failure in a magnetometer.

The rules for candidate sensor replacements depend on the application scenario and the particular requirements of the activity recognizer. Therefore, depending on the characteristics of the activity recognition problem a different set of rules should be used. Several rules which are generic and might apply to multiple scenarios are presented in the following. One should note that the results produced by some of these rules might be contradictory since they tackle different problems. Moreover, the list of rules is not exhaustive and only intends to showcase different possibilities offered by the proposed sensor selection method.

The identification of candidate sensor replacements should be done on the basis of the sensing functionalities offered by the MIMU sensors. The most prominent of these functionalities is the kind of magnitude measured by the sensor which determines the type of MIMU. Therefore, Table 1 shows three rules defined for the identification of candidate sensor replacements based on the type of anomalous MIMU sensor. The first idea one could consider to find a replacement for an anomalous MIMU in a wearable activity recognizer would be trying to get the signal of any other MIMU able to measure the same type of magnitude. Rule#1 describes this situation: if *?s1* and *?s2* are two different MIMUs which observe the same magnitude, represented in the rule as *?m1* and *?m2*, then *?s2* is a candidate replacement for *?s1*. If there is no other MIMU able to measure the same type of magnitude, transfer learning could be applied [43]. In this case, the requirement would be finding a replacement sensor of another modality capable of measuring a different type of magnitude (Rule#2). Moreover, in a more particular case of transfer learning where this technique could only be applied from the acceleration signal to the rate of turn signal, a failing accelerometer could be replaced by a gyroscope (Rule#3).

In the rules presented in Table 1, the kind of magnitude measured by the sensor, which determines the type of MIMU, is only considered for the identification of candidate sensor replacements. However, the sensing functionalities offered by a MIMU sensor are not only represented by the measured magnitude, but also by their measurement capabilities. Therefore, Table 2 presents some rules which enable the identification of candidate sensor replacements based on the measurement capabilities of the MIMUs. These rules extend the rules presented in Table 1 and incorporate restrictions on the measurement properties which define the measurement capabilities of the MIMUs. Rule#4 identifies candidate replacements which are able of measuring the same type of magnitude and have an equal or greater measurement range. Imposing this condition on the measurement range, one can expect that all the values of the signal collected originally will also be registered by the replacement MIMU, i.e., there will be no signal clipping. The rule states that if *?s1* and *?s2* are two different MIMUs which observe the same magnitude (*?m1* and *?m2*), and the upper limit of the measurement range of the second sensor (*?max2*) is greater or equal than the upper limit of the measurement range of the first sensor (*?max1*), and the lower limit of the measurement range of the second sensor (*?min2*) is less or equal than the lower limit of the measurement range of the first sensor (*?min1*), then *?s2* is a candidate replacement for *?s1*. Similarly, Rule#5 identifies candidate replacements which are able of measuring the same type of magnitude and have an equal or greater value for the sensitivity. In the case of Rule#6 the condition is imposed on the resolution of the candidate replacement which needs to be equal or greater than the original one. In Rule#7 and Rule#8 the candidate replacement MIMUs of the same type need to have, respectively, less or equal drift, and less or equal noise levels. Finally, Rule#9 identifies candidate replacements which belong to a different MIMU type but execute the measurements at the same rate or frequency. All these rules could be merged in any combination in order to pose simultaneously several conditions in more than one of the measurement properties which define the measurement capabilities of the MIMUs.

Candidate sensor replacements can also be identified on the basis of the characteristics of the wearable sensor platform such as its location on the body of the user [9]. In fact, the location where the wearable sensor platform is placed on the human body is of utmost importance for the performance of the wearable activity recognizer. Therefore, some rules incorporating the location of the wearable sensor platform hosting the MIMU need to be defined. Table 3 presents some rules for the identification of candidate sensor replacements based on the locations of the wearable sensor platform hosting the MIMU. The first option would be finding a candidate replacement for an anomalous MIMU in the same wearable sensor platform. This means that the two MIMUs coexist in the same physical device and it could be expected that they provide similar signals (Rule#10). Another option would be identifying as a candidate replacement a MIMU sensor hosted on a

Table 1
Rules for identification of candidate sensor replacements based on the MIMU types.

ID	Description	Rule
1	Same MIMU type	MIMU(?s1) ∧ MIMU(?s2) ∧ ssn:observes(?s1,?m1) ∧ ssn:observes(?s2,?m2) ∧ sameAs(?m1, ?m2) ∧ differentFrom(?s1,?s2) → hasReplacementSameType(?s1,?s2)
2	Different MIMU type	MIMU(?s1) ∧ MIMU(?s2) ∧ ssn:observes(?s1,?m1) ∧ ssn:observes(?s2,?m2) ∧ differentFrom(?m1, ?m2) → hasReplacementDiffType(?s1,?s2)
3	Gyroscope to replace an Accelerometer	Accelerometer(?s1) ∧ Gyroscope(?s2) → hasReplacementAccGyro(?s1,?s2)

Table 2
Rules for identification of candidate sensor replacements based on the measurement capabilities of the MIMU.

ID	Description	Rule
4	Same MIMU type with equal or greater Measurement Range	MIMU(?s1) ∧ MIMU(?s2) ∧ ssn:observes(?s1,?m1) ∧ ssn:observes(?s2,?m2) ∧ sameAs(?m1, ?m2) ∧ ssn:hasMeasurementCapability(?s1,?c1) ∧ ssn:hasMeasurementCapability(?s2,?c2) ∧ ssn:hasMeasurementProperty(?c1,?p1) ∧ ssn:hasMeasurementProperty(?c2,?p2) ∧ MimuMeasurementRange(?p1) ∧ MimuMeasurementRange(?p2) ∧ hasRangeMaxValue(?p1,?max1) ∧ hasRangeMaxValue(?p2,?max2) ∧ hasRangeMinValue(?p1,?min1) ∧ hasRangeMinValue(?p2,?min2) ∧ swrlb:greaterThanOrEqual(?max2, ?max1) ∧ swrlb:lessThanOrEqual(?min2, ?min1) ∧ differentFrom(?s1,?s2) → hasReplacementSameTypeRange(?s1,?s2)
5	Same MIMU type with equal or greater Sensitivity	MIMU(?s1) ∧ MIMU(?s2) ∧ ssn:observes(?s1,?m1) ∧ ssn:observes(?s2,?m2) ∧ sameAs(?m1, ?m2) ∧ ssn:hasMeasurementCapability(?s1,?c1) ∧ ssn:hasMeasurementCapability(?s2,?c2) ∧ ssn:hasMeasurementProperty(?c1,?p1) ∧ ssn:hasMeasurementProperty(?c2,?p2) ∧ MimuSensitivity(?p1) ∧ MimuSensitivity(?p2) ∧ hasQuantityValue(?p1,?v1) ∧ hasQuantityValue(?p2,?v2) ∧ swrlb:greaterThanOrEqual(?v2, ?v1) ∧ differentFrom(?s1,?s2) → hasReplacementSameTypeSens(?s1,?s2)
6	Same MIMU type with equal or greater Resolution	MIMU(?s1) ∧ MIMU(?s2) ∧ ssn:observes(?s1,?m1) ∧ ssn:observes(?s2,?m2) ∧ sameAs(?m1, ?m2) ∧ ssn:hasMeasurementCapability(?s1,?c1) ∧ ssn:hasMeasurementCapability(?s2,?c2) ∧ ssn:hasMeasurementProperty(?c1,?p1) ∧ ssn:hasMeasurementProperty(?c2,?p2) ∧ MimuResolution(?p1) ∧ MimuResolution(?p2) ∧ hasQuantityValue(?p1,?v1) ∧ hasQuantityValue(?p2,?v2) ∧ swrlb:greaterThanOrEqual(?v2, ?v1) ∧ differentFrom(?s1,?s2) → hasReplacementSameTypeRes(?s1,?s2)
7	Same MIMU type with equal or less Drift	MIMU(?s1) ∧ MIMU(?s2) ∧ ssn:observes(?s1,?m1) ∧ ssn:observes(?s2,?m2) ∧ sameAs(?m1, ?m2) ∧ ssn:hasMeasurementCapability(?s1,?c1) ∧ ssn:hasMeasurementCapability(?s2,?c2) ∧ ssn:hasMeasurementProperty(?c1,?p1) ∧ ssn:hasMeasurementProperty(?c2,?p2) ∧ MimuDrift(?p1) ∧ MimuDrift(?p2) ∧ hasQuantityValue(?p1,?v1) ∧ hasQuantityValue(?p2,?v2) ∧ swrlb:lessThanOrEqual(?v2, ?v1) ∧ differentFrom(?s1,?s2) → hasReplacementSameTypeDrift(?s1,?s2)
8	Same MIMU type with equal or less Noise	MIMU(?s1) ∧ MIMU(?s2) ∧ ssn:observes(?s1,?m1) ∧ ssn:observes(?s2,?m2) ∧ sameAs(?m1, ?m2) ∧ ssn:hasMeasurementCapability(?s1,?c1) ∧ ssn:hasMeasurementCapability(?s2,?c2) ∧ ssn:hasMeasurementProperty(?c1,?p1) ∧ ssn:hasMeasurementProperty(?c2,?p2) ∧ MimuNoise(?p1) ∧ MimuNoise(?p2) ∧ hasQuantityValue(?p1,?v1) ∧ hasQuantityValue(?p2,?v2) ∧ swrlb:lessThanOrEqual(?v2, ?v1) ∧ differentFrom(?s1,?s2) → hasReplacementSameTypeNoise(?s1,?s2)
9	Different MIMU type with same Frequency	MIMU(?s1) ∧ MIMU(?s2) ∧ ssn:observes(?s1,?m1) ∧ ssn:observes(?s2,?m2) ∧ differentFrom(?m1, ?m2) ∧ ssn:hasMeasurementCapability(?s1,?c1) ∧ ssn:hasMeasurementCapability(?s2,?c2) ∧ ssn:hasMeasurementProperty(?c1,?p1) ∧ ssn:hasMeasurementProperty(?c2,?p2) ∧ MimuFrequency(?p1) ∧ MimuFrequency(?p2) ∧ hasQuantityValue(?p1,?v1) ∧ hasQuantityValue(?p2,?v2) ∧ swrlb:equal(?v2, ?v1) ∧→ hasReplacementDiffTypeFreq(?s1,?s2)

wearable sensor platform located on the same body part where the anomalous sensor is (Rule#11). If the two sensors are worn on the same body part, one could expect that the measurements they provide would be very similar. In case no MIMU sensor hosted on a wearable sensor platform is located on the body part where the anomalous sensor is, it would be logical trying to identify a candidate replacement located on any of the adjacent body parts (Rule#12). If two body parts are connected, for example the forearm with the elbow, one could expect that their movements are similar and the MIMUs worn on them are candidate sensors for replacement. Rule#12 presents this idea: if two MIMUs (*?s1* and *?s2*) are part of two wearable sensor platforms (*?w1* and *?w2*) located on two connected body parts (represented in the rule as *connectedTo(?l1, ?l2)* where *?l1* and *?l2* are the body parts), then *?s2* is a candidate replacement for *?s1*. In case no sensor is available on the adjacent body parts, one could think of the identification of a replacement MIMU sensor hosted on a wearable sensor platform located on a body part directly connected to the adjacent body

Table 3
Rules for identification of candidate sensor replacements based on the location of the wearable sensor platform hosting the MIMU.

ID	Description	Rule
10	On same platform	MIMU(?s1) ∧ MIMU(?s2) ∧ ssn:hasSubsystem(?ws1,?s1) ∧ ssn:hasSubsystem(?ws2,?s2) ∧ ssn:OnPlatform(?ws1,?w1) ∧ ssn:OnPlatform(?ws2,?w2) ∧ sameAs(?w1, ?w2) ∧ differentFrom(?s1,?s2) → hasReplacementSamePlatf(?s1,?s2)
11	On same body part	MIMU(?s1) ∧ MIMU(?s2) ∧ ssn:hasSubsystem(?ws1,?s1) ∧ ssn:hasSubsystem(?ws2,?s2) ∧ ssn:OnPlatform(?ws1,?w1) ∧ ssn:OnPlatform(?ws2,?w2) ∧ placedOn(?w1,?l1) ∧ placedOn(?w2,?l2) ∧ sameAs(?l1, ?l2) ∧ differentFrom(?s1,?s2) → hasReplacementSamePart(?s1,?s2)
12	On an adjacent body part	MIMU(?s1) ∧ MIMU(?s2) ∧ ssn:hasSubsystem(?ws1,?s1) ∧ ssn:hasSubsystem(?ws2,?s2) ∧ ssn:OnPlatform(?ws1,?w1) ∧ ssn:OnPlatform(?ws2,?w2) ∧ placedOn(?w1,?l1) ∧ placedOn(?w2,?l2) ∧ connectedTo(?l1, ?l2) → hasReplacementAdjPart(?s1,?s2)
13	On a body part directly connected to an adjacent one	MIMU(?s1) ∧ MIMU(?s2) ∧ ssn:hasSubsystem(?ws1,?s1) ∧ ssn:hasSubsystem(?ws2,?s2) ∧ ssn:OnPlatform(?ws1,?w1) ∧ ssn:OnPlatform(?ws2,?w2) ∧ placedOn(?w1,?l1) ∧ placedOn(?w2,?l2) ∧ differentFrom(?l1,?l2) ∧ connectedTo(?l1, ?l3) ∧ connectedTo(?l2, ?l4) ∧ sameAs(?l3, ?l4) ∧ differentFrom(?s1,?s2) → hasReplacementConnAdjPart(?s1,?s2)
14	On the same body division	MIMU(?s1) ∧ MIMU(?s2) ∧ ssn:hasSubsystem(?ws1,?s1) ∧ ssn:hasSubsystem(?ws2,?s2) ∧ ssn:OnPlatform(?ws1,?w1) ∧ ssn:OnPlatform(?ws2,?w2) ∧ placedOn(?w1,?l1) ∧ placedOn(?w2,?l2) ∧ partOf(?l1,?d1) ∧ partOf(?l2,?d2), sameAs(?d1, ?d2) ∧ differentFrom(?s1,?s2) → hasReplacementSameDiv(?s1,?s2)
15	On the symmetric body part	MIMU(?s1) ∧ MIMU(?s2) ∧ ssn:hasSubsystem(?ws1,?s1) ∧ ssn:hasSubsystem(?ws2,?s2) ∧ ssn:OnPlatform(?ws1,?w1) ∧ ssn:OnPlatform(?ws2,?w2) ∧ placedOn(?w1,?l1) ∧ placedOn(?w2,?l2) ∧ symmetricTo(?l1, ?l2) → hasReplacementSymPart(?s1,?s2)
16	On the symmetric body division	MIMU(?s1) ∧ MIMU(?s2) ∧ ssn:hasSubsystem(?ws1,?s1) ∧ ssn:hasSubsystem(?ws2,?s2) ∧ ssn:OnPlatform(?ws1,?w1) ∧ ssn:OnPlatform(?ws2,?w2) ∧ placedOn(?w1,?l1) ∧ placedOn(?w2,?l2) ∧ partOf(?l1,?d1) ∧ partOf(?l2,?d2), symmetricTo(?d1, ?d2) → hasReplacementSymDiv(?s1,?s2)

Table 4
Rules for identification of candidate sensor replacements based on the survival range of the wearable sensor platform hosting the MIMU.

ID	Description	Rule
17	Battery lifetime greater than a certain limit (e.g., 3600 s)	MIMU(?s1) ∧ MIMU(?s2) ∧ ssn:hasSubsystem(?ws2,?s2) ∧ ssn:hasSurvivalRange(?ws2,?r2) ∧ ssn:hasSurvivalProperty(?r2,?b2) ∧ WearableBatteryLifetime(?b2) ∧ hasQuantityValue(?b2,?v2) ∧ swrlb:greaterThan(?v2, "3600"∧∧float) ∧ differentFrom(?s1,?s2) → hasReplacementBat(?s1,?s2)

part (Rule#13). According to this rule, a MIMU sensor hosted on a wearable sensor platform located on the arm would be a candidate to replace an anomalous sensor on the forearm. A more general option would be identifying as a candidate replacement a MIMU sensor hosted on a wearable sensor platform located on the same body division (Rule#14). For example, if the anomalous sensor is located on the forearm, any other MIMU sensor hosted on a wearable sensor platform located on the same upper limb would be a candidate sensor for replacement. Alternatively, another approach would be finding a candidate replacement for an anomalous MIMU in a wearable sensor platform located on the symmetric body part. Frequently, daily living activities equally involve both sides of the body; therefore, it might be reasonable to imagine that the signal produced by the sensor on the symmetric body part could be mirroring the actual one (Rule#15). Under these conditions, the MIMU sensor hosted on a wearable sensor platform located on left forearm would be a candidate to replace the anomalous sensor on the right forearm. A similar but more general option would identifying as a candidate replacement a MIMU sensor hosted on a wearable sensor platform located on the symmetric body division (Rule#16). According to this rule, any sensor on the left upper limb would be a candidate to replace an anomalous sensor on the right forearm.

The characteristics of the wearable sensor platform do not restrict to its location, but also include its survival range. Thus, a rule has been defined in Table 4 to identify candidate sensor replacements based on the survival range of the wearable platform. Rule#17 determines that a candidate replacement (*?s2*) is a MIMU sensor hosted on a wearable sensor platform (*?ws2*) which has a survival range represented by its battery lifetime (*?b2*), which takes a value (*?v2*) greater than a certain limit, for example 3600 s (*swrlb:greaterThan(?v2, "3600"∧∧float)*).

All the presented rules can be combined in order to obtain more meaningful descriptions for the identification of candidate sensor replacements. Table 5 shows some examples of these more complex rules. Rule#18 combines Rule#9 with Rule#10 and states that a candidate replacement is the sensor which belongs to a different MIMU type, executes the measurements at the same frequency and is hosted on the same wearable sensor platform. This rule would be typically used in a transfer learning scenario, where the two different types of MIMUs execute the measurements at the same rate, coexist on the same wearable platform, and the recognition model of the anomalous sensor can be transferred to the candidate replacement, for example from an accelerometer to a gyroscope. Rule#19 combines Rule#1 with Rule#11 and Rule#17 in order to identify as a candidate replacement a MIMU sensor of the same type hosted on a wearable sensor platform located on the same body part and which has a expected battery lifetime greater than a certain limit, for example 3600 s. Rule#20 combines

Table 5
Rules for identification of candidate sensor replacements based on the combinations of other rules.

ID	Description	Rule
18	Different MIMU type with same Frequency and on same platform	MIMU(?s1) ∧ MIMU(?s2) ∧ ssn:observes(?s1,?m1) ∧ ssn:observes(?s2,?m2) ∧ differentFrom(?m1, ?m2) ∧ ssn:hasMeasurementCapability(?s1,?c1) ∧ ssn:hasMeasurementCapability(?s2,?c2) ∧ ssn:hasMeasurementProperty(?c1,?p1) ∧ ssn:hasMeasurementProperty(?c2,?p2) ∧ MimuFrequency(?p1) ∧ MimuFrequency(?p2) ∧ hasQuantityValue(?p1,?v1) ∧ hasQuantityValue(?p2,?v2) ∧ swrlb:equal(?v2, ?v1) ∧ ssn:hasSubsystem(?ws1,?s1) ∧ ssn:hasSubsystem(?ws2,?s2) ∧ ssn:OnPlatform(?ws1,?w1) ∧ ssn:OnPlatform(?ws2,?w2) ∧ sameAs(?w1, ?w2) → hasReplacementDiffTypeFreqSamePlat(?s1,?s2)
19	Same MIMU type and on same body part and with a battery lifetime greater than a certain limit	MIMU(?s1) ∧ MIMU(?s2) ∧ ssn:observes(?s1,?m1) ∧ ssn:observes(?s2,?m2) ∧ sameAs(?m1, ?m2) ∧ ssn:hasSubsystem(?ws1,?s1) ∧ ssn:hasSubsystem(?ws2,?s2) ∧ ssn:OnPlatform(?ws1,?w1) ∧ ssn:OnPlatform(?ws2,?w2) ∧ placedOn(?w1,?l1) ∧ placedOn(?w2,?l2) ∧ sameAs(?l1, ?l2) ∧ ssn:hasSurvivalRange(?ws2,?r2) ∧ ssn:hasSurvivalProperty(?r2,?b2) ∧ WearableBatteryLifetime(?b2) ∧ hasQuantityValue(?b2,?v2) ∧ swrlb:greaterThan(?v2, "3600"∧∧float) ∧ differentFrom(?s1,?s2) → hasReplacementSameTypeSamePartBat(?s1,?s2)
20	Same MIMU type with equal or greater Measurement Range and on a symmetric body part	MIMU(?s1) ∧ MIMU(?s2) ∧ ssn:observes(?s1,?m1) ∧ ssn:observes(?s2,?m2) ∧ sameAs(?m1, ?m2) ∧ ssn:hasMeasurementCapability(?s1,?c1) ∧ ssn:hasMeasurementCapability(?s2,?c2) ∧ ssn:hasMeasurementProperty(?c1,?p1) ∧ ssn:hasMeasurementProperty(?c2,?p2) ∧ MimuMeasurementRange(?p1) ∧ MimuMeasurementRange(?p2) ∧ hasRangeMaxValue(?p1,?max1) ∧ hasRangeMaxValue(?p2,?max2) ∧ hasRangeMinValue(?p1,?min1) ∧ hasRangeMinValue(?p2,?min2) ∧ swrlb:greaterThanOrEqual(?max2, ?max1) ∧ swrlb:lessThanOrEqual(?min2, ?min1) ∧ ssn:hasSubsystem(?ws1,?s1) ∧ ssn:hasSubsystem(?ws2,?s2) ∧ ssn:OnPlatform(?ws1,?w1) ∧ ssn:OnPlatform(?ws2,?w2) ∧ placedOn(?w1,?l1) ∧ placedOn(?w2,?l2) ∧ symmetricTo(?l1, ?l2) → hasReplacementSameTypeRangeSymPart(?s1,?s2)

```
SELECT ?replacementsensor
WHERE { <sensor-id> hasReplacement ?replacementsensor. }
```

Listing 1. SPARQL query to retrieve all the candidate sensors to replace a MIMU with identifier `<sensor-id>`.

Rule#4 with Rule#15 and identifies that a candidate replacement is a MIMU which is able of measuring the same type of magnitude with an equal or greater measurement range, and is part of a wearable sensor platform located on the symmetric body part.

The rules depend on the application scenario and the specific requirements for the replacement sensor. Therefore, the presented rules have to be particularized depending on the application requirements.

4.2. Queries for sensor selection

The novel selection method for the replacement of anomalous sensors proposed in this work is based on an iterative query process triggered once a sensor is detected to have failed. Posing the adequate queries on the descriptions of the available MIMU-based wearable sensor platforms will allow selecting the best MIMU sensors which could replace the ones suffering from anomalies in a wearable activity recognizer.

The query method builds on the MIMU-Wear ontology and depends on the set of rules which are defined for each application scenario. The priorities assigned to the outcomes of each of the rules are pretty important for the effectiveness of the query method. Therefore, a particular sequence order should be established for the execution of the queries depending on the specific problem.

SPARQL [44], a query language for RDF, is utilized in the sensor selection method because of its fully potential to query OWL 2 data. Listing 1 shows a query which retrieves all the candidate sensors to replace an anomalous MIMU. In fact, the string `<sensor-id>` in the query must be replaced with the actual identifier of the anomalous MIMU, which is the name of the individual of the class `MIMU` representing this very MIMU in the ontology. The query is very abstract and applies to any MIMU independently of its characteristics and the wearable sensor platform in which it is hosted. This generality avoids having to know the actual characteristics of the anomalous sensor in order to pose the query. These characteristics are inferred from the ontology and the rules and implicitly used in the query execution. Thus, the main benefit is that in the query method only the identifier of the anomalous sensor is needed.

The query presented in Listing 1 should be particularized in order to obtain a more reduced set of MIMU sensors which are possible candidates for the replacement of an anomalous sensor in a wearable activity recognizer. The restriction of the results is based on querying for a specific subproperty of the property `hasReplacement`, instead of using the generic one. For example, if the expected result is a set of candidate MIMUs which are able of measuring the same type of magnitude with an equal or greater measurement range, and which are part of a wearable sensor platform located on the symmetric body part (Rule#20), the SPARQL query in Listing 2 should be executed.

The SPARQL solution modifier `ORDER BY` could be utilized to order the query results depending one criteria, so that the selection between the candidates replacement sensors is facilitated. For example, the SPARQL result could order the candidate MIMU sensors depending on the battery lifetime of the wearable sensor platform. Moreover, the potential of the SPARQL algebra could enable obtaining as result only one candidate replacement sensor which is the one that maximizes or minimizes one search criteria. As an example, the result could be the candidate MIMU sensor hosted in the wearable sensor platform which has the longest expected battery lifetime.

The sensor selection method is based on an iterative query process triggered once a sensor is detected to behave anomalously.

```
SELECT ?replacementsensor
WHERE { <sensor-id> hasReplacementSameTypeRangeSymPart ?replacementsensor. }
```

Listing 2. SPARQL query for the identification of candidate replacement sensors able of measuring the same type of magnitude with an equal or greater measurement range and which are part of a wearable sensor platform located on the symmetric body part.

The iterative method ensures that if no result is provided for a query, another less restrictive query or with another criteria is executed in order to obtain as result a candidate replacement sensor. For example, in a particular scenario, the logic could be that the first option in order to replace an anomalous MIMU is trying to find a replacement sensor hosted on a wearable sensor platform located on the same body part. Therefore, it executes a SPARQL query for the results of Rule#11, i.e., on the property `hasReplacementSamePart`. In case no sensor is found on the same body part, it tries to find a candidate replacement hosted on a wearable sensor platform located on any of the adjacent body parts by querying for the results of Rule#12 on the property `hasReplacementAdjPart`. In case no sensor is found on the adjacent parts, the closest sensor could be searched, or in the case of a MIMU sensor is hosted on a wearable sensor platform located on a limb, it could try to find a sensor in the symmetric part. This is executing the query for the results of Rule#15 on the property `hasReplacementSymPart`.

5. An example of sensor replacement using MIMU-Wear and the sensor selection method

In order to provide a clearer view of the MIMU-Wear Ontology and the functioning of the proposed sensor selection method, this section presents an exemplary application scenario. The sensor selection method is here used to ensure continuity in real-world wearable activity recognition. In a real world scenario, the wearable sensors might suffer from anomalies, such us failures or faults. Whenever this happens, the performance of the activity recognizer decreases due to the processing of a corrupted signal, or in the worst case, not receiving any signal at all. Once detecting such a situation, it would be desirable to replace the anomalous sensor with another one which provides the same sensing functionality. The final goal is that after the sensor replacement, the performance of the wearable activity recognizer gets restored to its original value, or at least it increases with respect to the failure situation. The proposed sensor selection method should be triggered when a sensor is detected to fail [45,46] in order to identify an appropriate candidate sensor replacement. After the candidate has been identified using the proposed sensor selection method, the actual replacement should take place. The replacement process is not part of this work neither the detection of the sensor failure.

5.1. Deployment scenario

Nine MIMU-based wearable platforms are considered in this exemplary scenario (see Fig. 11). The wearable sensor platforms are symmetrically distributed all over the body and worn on the four limbs and the trunk. Specifically, the MIMU-based wearable platforms are deployed in this scenario are the MTx, a 3DOF inertial Orientation Tracker developed by Xsens [47]. The MTx wearable sensor platforms are composed of three MIMU sensors: a 3D accelerometer, a 3D gyroscope and 3D magnetometer. More details on the specifications of these sensors are provided in the following section.

5.2. Scenario description using MIMU-Wear

The MIMU-Wear Ontology is here used to describe the nine MTx wearable sensor platforms in this exemplary scenario. The logic for the user of MIMU-Wear is presented in Section 3.3, but here the process is described through an example. First, the generic descriptions of the three MIMUs embedded into MTx wearable sensor platform are created. Second, the description of the MTx wearable sensor platform is generated and the links to the embedded MIMUs are established. Finally, the specific descriptions for the nine particular instances of the MTx wearable sensor platforms are defined.

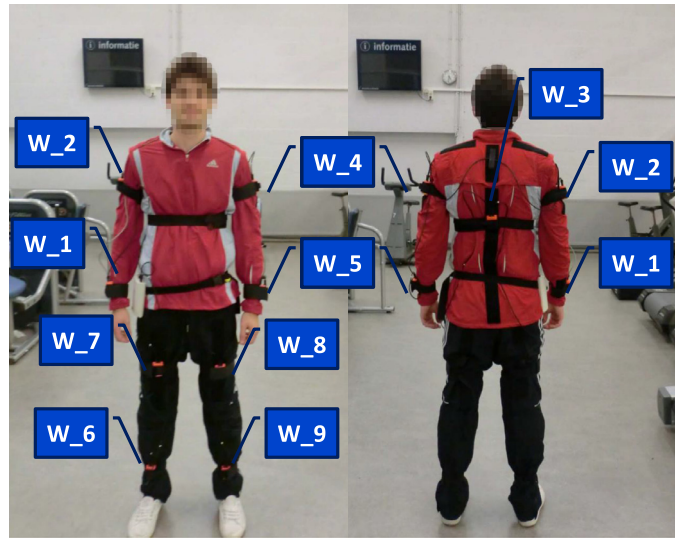

Fig. 11. Sensor deployment for the experimental scenario. Nine MIMU-based wearable platforms (W_1,...,W_9) are placed on different parts of the user body.

5.2.1. Generic description of the MTx wearable sensor platform

In a practical application, where the MIMU-Wear ontology would be widely adopted, the creation of the generic description of the sensors and wearable sensor platforms would performed only once by the manufacturer. In this case Xsens would make available the ontological classes describing their MIMU sensors and wearable sensor platforms. The process of creating these descriptions is explained in the following.

In order to represent the accelerometer embedded into MTx, the class `MTxAcc` is defined as a subclass of the class `Accelerometer` and of the anonymous class `ssn:hasMeasurementCapability only MTxAccMeasurementCapability`. The class `MTxAccMeasurementCapability` is defined here to represent the actual measurement capabilities of the MTx accelerometer and is a subclass of the class `AccelerometerMeasurementCapability`. The MTx accelerometer has three different configurations which determine the different values of its measurement properties such as its measurement range, sensitivity, or noise. Each one of these configurations is modeled as an individual of the class `MTxAccMeasurementCapability`, namely `standard_MTxAcc_capability`, `costum17_MTxAcc_capability` and `costum100_MTxAcc_capability`. The values of the measurement properties of the MTx can be obtained from its specification sheet [47] and are ontologically modeled as follows.

The measurement range for the standard configuration of the MTx accelerometer is $\pm$ 50 m/s^2. In order to represent this range, the individual `standard_MTxAcc_mrange` of the class

AccelerometerMeasurementRange has asserted for the property hasMaxValue the value ‘‘50’’∧∧float and for the property hasMinValue the value ‘‘-50’’∧∧float. The sensitivity of the MTx accelerometer is 0.5% of the full scale or measurement range, which means 0.5 m/s^2 for the standard configuration. The individual standard_MTxAcc_sensitivity of the class AccelerometerSensitivity has asserted for the property hasQuantityValue the value ‘‘0.5’’∧∧float. Since the frequency of the MTx accelerometer is 30 Hz, the individual standard_MTxAcc_frequency of the class MimuFrequency has asserted for the property hasQuantityValue the value ‘‘30’’∧∧float. The drift of the MTx accelerometer is 0.1° and is represented as the individual standard_MTxAcc_drift of the class MimuDrift which has asserted for the property hasQuantityValue the value ‘‘0.1’’∧∧float. Finally, the noise of the MTx accelerometer is 0.002 m/s^2/$\sqrt{}$Hz for the standard configuration. Thus, the individual standard_MTxAcc_noise of the class AccelerometerNoise has asserted for the property hasQuantityValue the value ‘‘0.002’’∧∧float. In one of the costume configurations, the measurement range of the MTx accelerometer is ± 17 m/s^2. Thus, the individual custom17_MTxAcc_mrange of the class AccelerometerMeasurementRange has asserted for the property hasMaxValue the value ‘‘17’’∧∧float and for the property hasMinValue and the value ‘‘-17’’∧∧float. In this configuration, the sensitivity of 0.5% of the measurement range is equivalent to 0.17 m/s^2. This sensitivity is represented as the individual custom17_MTxAcc_sensitivity of the class AccelerometerSensitivity which has asserted for the property hasQuantityValue the value ‘‘0.17’’∧∧float. The rest of measurement properties - frequency, drift and noise - take the same values for this configuration than for the standard case. Therefore, no new individuals are created to represent them but the ones already defined for the standard configuration are reused. In the other costume configuration, the measurement range of the MTx accelerometer is ± 100 m/s^2. Thus, the individual custom100_MTxAcc_mrange of the class AccelerometerMeasurementRange has asserted for the property hasMaxValue the value ‘‘100’’∧∧float and for the property hasMinValue the value ‘‘-100’’∧∧float. In this configuration, the sensitivity of 0.5% of the measurement range is equivalent to 1 m/s^2. This sensitivity is represented as the individual custom100_MTxAcc_sensitivity of the class AccelerometerSensitivity which has asserted for the property hasQuantityValue the value ‘‘1’’∧∧float. Moreover, the noise of the MTx accelerometer for this configuration is 0.003 m/s^2/$\sqrt{}$Hz and to represent it the individual custom100_MTxAcc_noise of the class AccelerometerNoise has asserted for the property hasQuantityValue the value ‘‘0.003’’∧∧float. The rest of measurement properties for this configuration take the same values than in the case of the standard configuration.

Having defined the different values for the measurement properties, these can be linked to the actual measurement capabilities which represent each one of the MTx accelerometer configurations. In order to represent the capabilities of the standard configuration, the individual standard_MTxAcc_capability has asserted for the property ssn:hasMeasurementProperty the following individuals standard_MTxAcc_mrange, standard_MTxAcc_sensitivity, standard_MTxAcc_frequency, standard_MTxAcc_drift, and standard_MTxAcc_noise. For one of the custom configurations, the individual costum17_MTxAcc_capability has asserted for the property ssn:hasMeasurementProperty the individuals custom17_MTxAcc_mrange, custom17_MTxAcc_sensitivity, standard_MTxAcc_frequency, standard_MTxAcc_drift, and standard_MTxAcc_noise. For the other custom configuration, the individual costum100_MTxAcc_capability has asserted for the property ssn:hasMeasurementProperty the individuals custom100_MTxAcc_mrange, custom100_MTxAcc_noise, custom100_MTxAcc_sensitivity, standard_MTxAcc_frequency, and standard_MTxAcc_drift.

Following a similar approach, the class MTxGyro could be defined to represent the gyroscope embedded into MTx and the class MTxMag to represent the magnetometer. The generic description of the wearable sensor platform can be created when the description of the three types of MIMUs is already available.

The MTx wearable sensor platform is represented via the class MTxPlat which is a subclass of the class WearableSensorPlatform. The class MTxSystem is defined to be a subclass of the class WearableSystem and of the anonymous class ssn:onPlatform only MTxPlat. Moreover, the class MTxSystem is asserted to be a subclass of the anonymous classes ssn:hasSubsystem exactly 1 MTxAcc, ssn:hasSubsystem exactly 1 MTxGyro and ssn:hasSubsystem exactly 1 MTxMag. These cardinality restrictions state that the MTx wireless sensor platform is composed of one MTx accelerometer, one MTx gyroscope and one MTx magnetometer.

5.2.2. Description of the particular MTx elements in the scenario

Let now imagine that the manufacturer has created the generic ontological description of the MTx and has made it is available online. Then, the particular description of the wearable sensor platforms and their embedded MIMUs can be easily created. These definitions could be created by the final user when utilizing the application, but it would be more common that they would be automatically generated at application setup. Anyway, this would require that the designer of the activity recognition application provides an interface to create the descriptions. In the following, the manual creation of the MTx descriptions is explained.

The description of the accelerometer embedded into the MTx is already defined via the class MTxAcc. Nine individuals of this class - MTxAcc_1, MTxAcc_2,..., MTxAcc_9 - can be created in order to represent the accelerometers in each one of the wearable sensor platforms - W_1, W_2,..., W_9. For example, the individual MTxAcc_1 (see Fig. 12) represents the accelerometer hosted on W_1, i.e., on the wearable sensor platform worn on the right forearm. Only one axiom has to be asserted in order to define each individual, since the rest of the definition is directly derived from the class description. Particularly, the value of the property ssn:hasMeasurementCapability has to be asserted in order to model the specific capabilities of the MTx accelerometer for the current working mode. This property can only take as value three individuals standard_MTxAcc_capability, costum17_MTxAcc_capability or costum100_MTxAcc_capability depending on the accelerometer configuration. Since all the accelerometers work in the standard configuration, the nine individuals of the class MTxAcc will have asserted the individual standard_MTxAcc_capability for the property hasMeasurementCapability.

The individuals of the class MTxPlat represent the particular wearable sensor platforms. For example, the individual MTxPlat_1 (see Fig. 13) represents the wearable sensor platform worn on the right forearm (W_1 in the scenario). The corresponding individual MTxSys_1 (see Fig. 14) of the class MTxSystem is created and linked to MTxPlat_1 via the property ssn:onPlatform. Moreover, the specific MIMUs which are part of the wireless sensor platform are asserted as the values of the property ssn:hasSubsystem. For example, in the case of the individual MTxSys_1, the property ssn:hasSubsystem

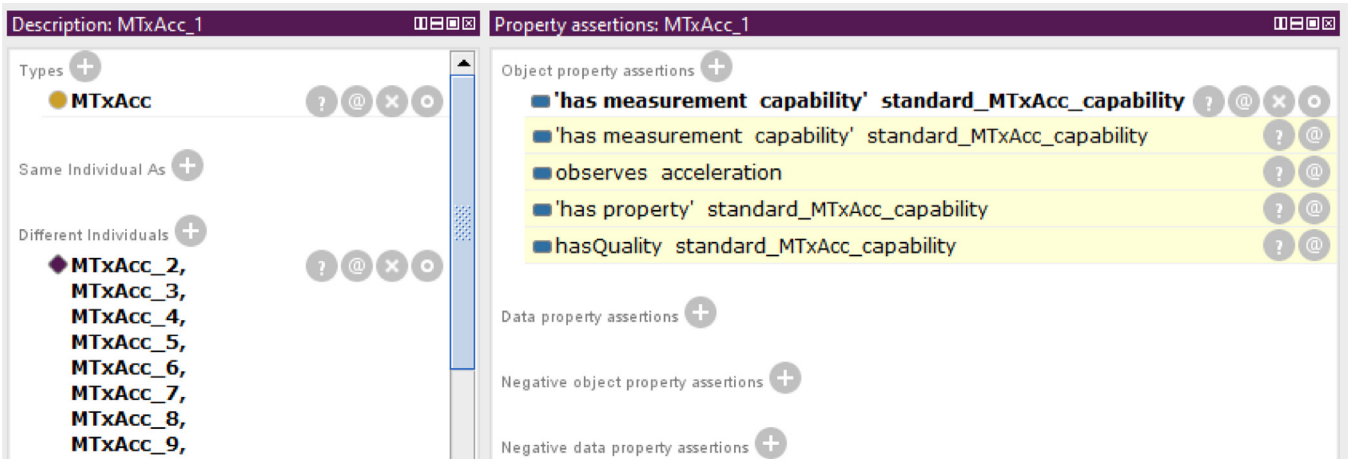

Fig. 12. `MTxAcc_1`: instance of the the class `MTxAcc` which describes the accelerometer embedded into the MTx. The axioms marked in yellow represent the inferred knowledge, whereas the others are asserted axioms. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

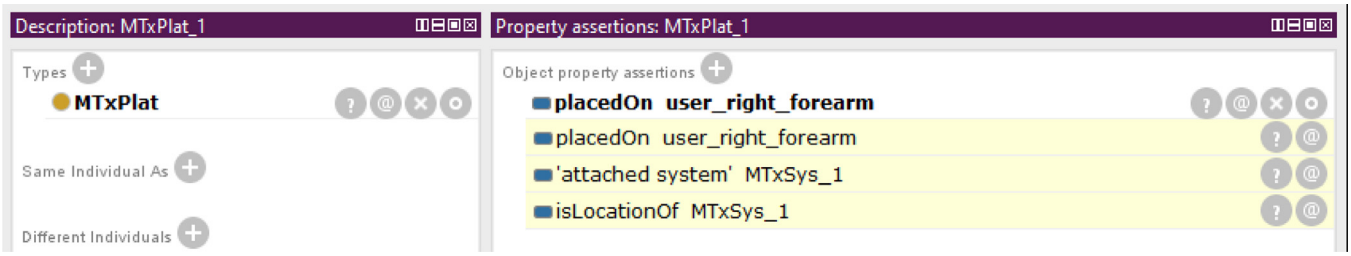

Fig. 13. `MTxPlat_1`: instance of the the class `MTxPlat` which describes the MTx. The axioms marked in yellow represent the inferred knowledge, whereas the others are asserted axioms. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

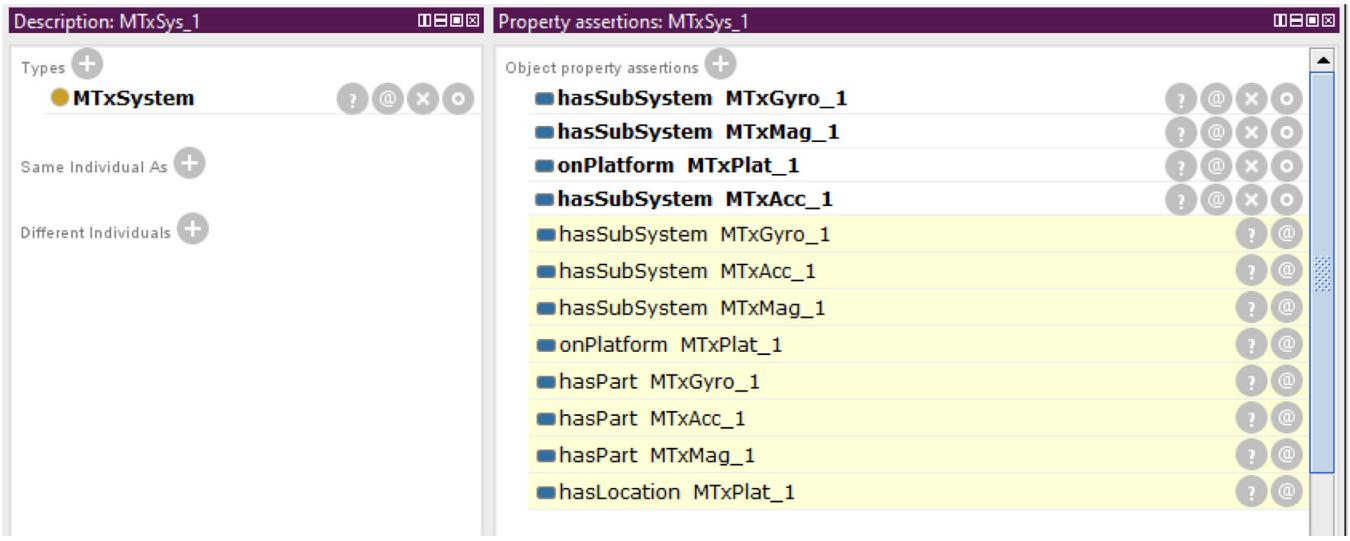

Fig. 14. `MTxSys_1`: instance of the the class `MTxSystem` which describes the MTx. The axioms marked in yellow represent the inferred knowledge, whereas the others are asserted axioms. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

takes as values `MTxAcc_1`, `MTxGyro_1` and `MTxMag_1`, which are individuals of the classes `MTxAcc`, `MTxGyro` and `MTxMag`, respectively. In order to complete the description of the wearable sensor platform, its deployment on the body of the user should be modeled. This is done by asserting the value of the property `placedOn` for the individual of the class `MTxPlat`. For example, in the case of the MTx wearable sensor platform worn on the right forearm, the property `placedOn` of the individual `MTxPlat_1` takes as value the individual `user_right_forearm`, which is a member of the class `Forearm` in the Human Body Ontology and represents the right forearm of the user.

Similarly, the nine individuals of the class `MTxGyro` and nine of the class `MTxMag` should be created in order to represent the nine gyroscopes and the nine magnetometers hosted in the MTx wearable sensor platforms. Moreover, the descriptions of the other eight wearable sensor platforms (`MTxPlat_2`, ... `MTxPlat_9`) could be created. Table 6 summarizes the names of the individuals representing each of the sensors and wearable sensor platforms.

Table 6
Summary of the ontological description of the MIMU-based wearable sensor platforms for the scenario presented in Fig. 11.

ID (Fig 11)	Wearable sensor platform (individual of MTxPlat)	Location (MTxPlat placedOn)	Hosted MIMUs (MTxSystem ssn:hasSubsystem)
W_1	MTxPlat_1	user_right_forearm	MTxAcc_1, MTxGyro_1, MTxMag_1
W_2	MTxPlat_2	user_right_arm	MTxAcc_2, MTxGyro_2, MTxMag_2
W_3	MTxPlat_3	user_back	MTxAcc_3, MTxGyro_3, MTxMag_3
W_4	MTxPlat_4	user_left_arm	MTxAcc_4, MTxGyro_4, MTxMag_4
W_5	MTxPlat_5	user_left_forearm	MTxAcc_5, MTxGyro_5, MTxMag_5
W_6	MTxPlat_6	user_right_leg	MTxAcc_6, MTxGyro_6, MTxMag_6
W_7	MTxPlat_7	user_right_thigh	MTxAcc_7, MTxGyro_7, MTxMag_7
W_8	MTxPlat_8	user_left_thigh	MTxAcc_8, MTxGyro_8, MTxMag_8
W_9	MTxPlat_9	user_left_leg	MTxAcc_9, MTxGyro_9, MTxMag_9

5.3. Application of the sensor selection method

The wearable activity recognition scenario considered in this example seeks to conform as much as possible to a real-world setup. Previous works have proven that using several sensors usually results in a higher level of accuracy and robustness of the wearable activity recognizer [8,9]. On the other hand, the more sensors are utilized, the more computationally and energy expensive the activity recognizer turns to be [10], being the latter particularly critical for wearable systems. For these reasons, a balanced sensor setup is defined for this exemplary scenario. In a balanced setup multiple sensors are available but only a subset of them actively participate in the recognition process. The remaining sensors are kept in an idle or sleep state and can be used as replacement ones.

The nine MIMU-based wearable platforms deployed in this scenario (see Fig. 11) are configured to capture only acceleration since this magnitude proves to work well-enough for the recognition of a variety of activities [48]. Therefore, gyroscopes and magnetometers are not operating and only accelerometers are measuring. Moreover, three out of the nine accelerometers are actually used for the activity recognition process while the rest remain in idle state. The used MIMUs are the accelerometer hosted on W_1 - the wearable sensor platform on the right forearm -, the accelerometer hosted in W_3 - the wearable sensor platform on the back -, and the accelerometer on W_9 - the wearable sensor platform on the left leg -. This setup has been chosen because it has shown to provide a good trade-off between number of sensors and performance and it has been successfully used in some prior applications [49–52].

In order to showcase the functioning of the sensor selection method whenever one of the three accelerometers hosted on the wearable sensor platforms W_1, W_3 or W_9 behaves anomalously, particularly, when it runs out of battery, the generic rules presented in Section 4.1 have to be particularized for the actual application scenario and the queries resented in Section 4.2 prioritized accordingly.

Four rules are envisioned for this application scenario (see Table 7). These rules build on common sense assumptions about which are the best candidate replacement sensors. The first and best option for a candidate replacement would be a MIMU sensor that measures the same type of magnitude and which is hosted on a wearable sensor platform located on the same body part where the anomalous sensor is. This rule which has priority P1 is a combination of Rule#1 in Table 1 and Rule#11 in Table 3. A second option consist in trying to find a candidate replacement of the same MIMU type and hosted in a wearable sensor platform located on the symmetric body part. The rule which reflects this situation has priority P2 and is a combination of Rule#1 in Table 1 and Rule#15 in Table 3. The third rule is a combination of Rule#1 in Table 1 and Rule#12 in Table 3, has priority P3 and seeks to identify a candidate replacement able of measuring the same magnitude and hosted in a wearable sensor platform located on any of the adjacent body parts. Finally, the forth rule which has priority P4, aims at identifying a replacement MIMU sensor which measures the same magnitude and which is hosted on a wearable sensor platform located on a body part directly connected to an adjacent body part. This rule is a combination of Rule#1 in Table 1 and Rule#13 in Table 3.

In order to request the results provided by the rule with priority P1, the associated SPARQL query which is executed is shown in Listing 3. In the same way, to retrieve the results of rule with priority P2 the associated query is shown in Listing 4, for the rule with priority P3 the query is shown in Listing 5, and for the rule with priority P4 the query is shown in Listing 6. In all the queries the string `<sensor-id>` must be replaced with the actual identifier of the anomalous MIMU, for example `MTxAcc_1`, `MTxAcc_3` or `MTxAcc_9`.

Summarizing, the iterative query method for sensor selection works as follows. First the query in Listing 3 is executed. If it provides a result, the search is stopped since a candidate replacement has been identified. Otherwise, the query in Listing 4 is executed and so on. If no results are obtained while executing the last query, i.e., Listing 6, the method finalizes without success.

Let us suppose that the accelerometer in W_1, i.e., the accelerometer hosted on the MTx wearable sensor platform located on the user right forearm, suffers from a failure condition, namely it runs out of battery and stops working. Once this event is identified, e.g., by the network monitor, the sensor selection method is triggered. First, it is executed the SPARQL query presented in Listing 3 while substituting the string `<sensor-id>` with the string `MTxAcc_1`. The latter string corresponds to the name of the ontological individual `MTxAcc_1` of the class `MTxAcc` and represents the accelerometer sensor in W_1. This query does not produce any result because there are no other accelerometer hosted on a wearable sensor platform located on the user right forearm. Then, the iterative query process continues by executing the query presented in Listing 4, where the string `<sensor-id>` is replaced with `MTxAcc_1`. This query returns the individual `MTxAcc_5`, which is the accelerometer in W_5, i.e., the accelerometer hosted on the MTx wearable sensor platform located on the user left forearm. The rationale for this result is the following. Both the individual `MTxAcc_1` and the individual `MTxAcc_5` have inferred for the property `ssn:observes` the individual `acceleration`. Moreover, the individual `MTxSys_1` has asserted for the property `ssn:hasSubsystem` the individual `MTxAcc_1` and for the property `ssn:OnPlatform` the individual `MTxPlat_1`. Also the individual `MTxPlat_1` has asserted for the property `placedOn` the individual `user_right_forearm`. Furthermore, the individual `MTxSys_5` has asserted for the property `ssn:hasSubsystem` the individual `MTxAcc_5` and for the property `ssn:OnPlatform`

Table 7
Prioritized set of rules for identification of candidate sensor replacements in the exemplary application scenario.

Priority	Rule
P1	MIMU(?s1) ∧ MIMU(?s2) ∧ ssn:observes(?s1,?m1) ∧ ssn:observes(?s2,?m2) ∧ sameAs(?m1, ?m2) ∧ ssn:hasSubsystem(?ws1,?s1) ∧ ssn:hasSubsystem(?ws2,?s2) ∧ ssn:OnPlatform(?ws1,?w1) ∧ ssn:OnPlatform(?ws2,?w2) ∧ placedOn(?w1,?l1) ∧ placedOn(?w2,?l2) ∧ sameAs(?l1, ?l2) ∧ differentFrom(?s1,?s2) → hasReplacementSameTypeSamePart(?s1,?s2)
P2	MIMU(?s1) ∧ MIMU(?s2) ∧ ssn:observes(?s1,?m1) ∧ ssn:observes(?s2,?m2) ∧ sameAs(?m1, ?m2) ∧ ssn:hasSubsystem(?ws1,?s1) ∧ ssn:hasSubsystem(?ws2,?s2) ∧ ssn:OnPlatform(?ws1,?w1) ∧ ssn:OnPlatform(?ws2,?w2) ∧ placedOn(?w1,?l1) ∧ placedOn(?w2,?l2) ∧ symmetricTo(?l1, ?l2) → hasReplacementSameTypeSymPart(?s1,?s2)
P3	MIMU(?s1) ∧ MIMU(?s2) ∧ ssn:observes(?s1,?m1) ∧ ssn:observes(?s2,?m2) ∧ sameAs(?m1, ?m2) ∧ ssn:hasSubsystem(?ws1,?s1) ∧ ssn:hasSubsystem(?ws2,?s2) ∧ ssn:OnPlatform(?ws1,?w1) ∧ ssn:OnPlatform(?ws2,?w2) ∧ placedOn(?w1,?l1) ∧ placedOn(?w2,?l2) ∧ connectedTo(?l1, ?l2) → hasReplacementSameTypeAdjPart(?s1,?s2)
P4	MIMU(?s1) ∧ MIMU(?s2) ∧ ssn:observes(?s1,?m1) ∧ ssn:observes(?s2,?m2) ∧ sameAs(?m1, ?m2) ∧ ssn:hasSubsystem(?ws1,?s1) ∧ ssn:hasSubsystem(?ws2,?s2) ∧ ssn:OnPlatform(?ws1,?w1) ∧ ssn:OnPlatform(?ws2,?w2) ∧ placedOn(?w1,?l1) ∧ placedOn(?w2,?l2) ∧ differentFrom(?l1,?l2) ∧ connectedTo(?l1, ?l3) ∧ connectedTo(?l2, ?l4) ∧ sameAs(?l3, ?l4) ∧ differentFrom(?s1,?s2) → hasReplacementSameTypeConnAdjPart(?s1,?s2)

```
SELECT ?replacementsensor
WHERE { <sensor-id> hasReplacementSameTypeSamePart ?replacementsensor. }
```

Listing 3. SPARQL query for retrieving the results of rule with priority P1.

```
SELECT ?replacementsensor
WHERE { <sensor-id> hasReplacementSameTypeSymPart ?replacementsensor. }
```

Listing 4. SPARQL query for retrieving the results of rule with priority P2.

```
SELECT ?replacementsensor
WHERE { <sensor-id> hasReplacementSameTypeAdjPart ?replacementsensor. }
```

Listing 5. SPARQL query for retrieving the results of rule with priority P3.

```
SELECT ?replacementsensor
WHERE { <sensor-id> hasReplacementSameTypeConnAdjPart ?replacementsensor. }
```

Listing 6. SPARQL query for retrieving the results of rule with priority P4.

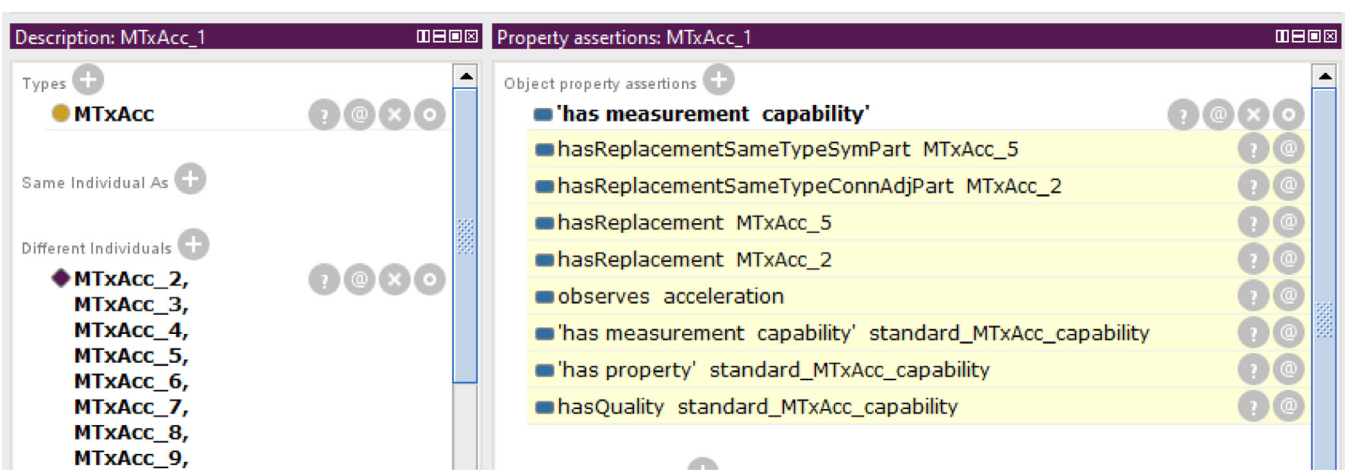

Fig. 15. Instance `MTxAcc_1` of the class `MTxAcc` which shows the candidate replacement sensors via the inferred property `hasReplacement` and its subproperties.

the individual `MTxPlat_5`. Also the individual `MTxPlat_5` has asserted for the property `placedOn` the individual `user_left_forearm`. From the Human Body Ontology it can be inferred that the individuals `user_right_forearm` and `user_left_forearm` are related via the property `symmetricTo`. Therefore, the rule with priority P2 in Table 7 is satisfied and the following axiom is inferred `MTxAcc_1 hasReplacementSameTypeConnAdjPart MTxAcc_5` (see Fig. 15). The SPARQL query which retrieves the value of the property `hasReplacementSameTypeConnAdjPart` for the individual `MTxSys_1` gets then as a result the individual `MTxPlat_5`. In conclusion, the sensor selection method determines that the anomalous accelerometer in W_1 - the accelerometer hosted on the MTx wearable sensor platform located on the user right forearm - could be replaced with the accelerometer in W_5 - the accelerometer hosted on the MTx wearable sensor platform located on the user left forearm -.

Let us now suppose that the MIMU which runs out of battery is the accelerometer in W_9, i.e., the accelerometer hosted on the MTx wearable sensor platform located on the user left leg. The query method would be applied in the same way as in the previous case but replacing in the SPARQL queries the string `<sensor-id>` with `MTxAcc_9` . In this case, the first query (Listing 3) does not produce any result. Then, the second query (Listing 4) is executed and returns as result the individual `MTxAcc_6`, which is the accelerometer in W_6. Therefore, the

anomalous accelerometer in W_9 - the accelerometer hosted on the MTx wearable sensor platform located on the user left leg - could be replaced with the accelerometer in W_6 - the accelerometer hosted on the MTx wearable sensor platform located on the user right leg - according to the results of the sensor selection method.

Finally, let us now suppose that the failure is suffered by the accelerometer in W_3, i.e., the accelerometer hosted on the MTx wearable sensor platform located on the user back. In this case the queries to be applied have replaced the string `<sensor-id>` with the string `MTxAcc_3`. The first step is executing the query in Listing 3. This query does not produce any result since there is no other accelerometer hosted in a platform located on the user back, in fact there is no other wearable sensor platform located on the back. Then, the iterative query process continues and the second step consists on executing the query presented in Listing 4. This query does not return any result because the individual `user_back` in the Human Body Ontology does not have asserted neither inferred the property `symmetricTo`, i.e., there is no symmetric body part for the back. In the third step of the query process, the query presented in Listing 5 is executed. This query does not produce any result because there is no wearable sensor platform located on the body parts adjacent to the back. According to the model of the Human Body Ontology, the body parts which are directly connected to the back are the thorax and the abdomen which are also in the trunk, and the shoulders which are part of the limbs but connect to the trunk. After the failure of the first three queries, the fourth one which is presented in Listing 6 is executed. This query returns two individuals: `MTxAcc_2`, which is the accelerometer in W_2, i.e., the accelerometer hosted on the MTx wearable sensor platform located on the user right arm, and `MTxAcc_4`, which is the accelerometer in W_4, i.e., the accelerometer hosted on the MTx wearable sensor platform located on the user left arm. The logical explanation of this query result is that the back is connected to both shoulders and each shoulder is connected to the arm. Thus, the sensors on both arms are located at a distance of two hops from the back. Two results are obtained since the characteristics of the accelerometers hosted on W_2 and W_4 are the same, and the rules do not state any preference in choosing one instead of the other. Thus, both of them could be used as replacements for W_3, the accelerometer hosted on the MTx wearable sensor platform located on the user back.

5.4. Validation

The MIMU-Wear Ontology has been validated using symbolic methods [53]. The HermiT reasoner has been utilized in protégé to perform the tests. The consistency of the ontology has been verified. Moreover, the ontology is determined to be well classified. The rules for the presented scenario have been also validated. Thus, it can be assured the well functioning of the proposed ontology.

The scalability of the proposed solution might depend on the size of the knowledge base generated when describing the wearable sensor platforms and the MIMU sensors embedded into these platforms. Moreover, the semantic sensor selection method creates inferred knowledge. Based on the proposed scenario, the metrics that define the ontology have been measured using protégé (see Table 8). The MIMU-Wear ontology is composed of 932 axioms. The generic definition of the MTx wearable sensor platforms requires 117 axioms for the definition of the 8 classes and 23 individuals. The particular description of the nine MTx deployed in the scenario requires 214 axioms for the definition of 58 individuals. One can observe that the inferred knowledge reaches almost the same size than the original asserted knowledge. Most of this knowledge is derived from the actual logic encoded in the ontology: 427 inferred class axioms and 331 inferred property axioms. Thus, if only the inferred knowledge associated to the rules is considered, there are only 102 object properties inferred for an scenario with nine wearable sensor platforms containing three different MIMUs each one of them. To determine the knowledge associated to the rules, the knowledge based has been inferred with and without the rules and the difference in the count of the axioms has been calculated.

5.5. Evaluation

The different replacement scenarios described before are here evaluated by using the REALDISP dataset [54]. This dataset comprises acceleration, rate of turn and magnetic field orientation data collected for 17 people while performing 33 fitness activities in an out-of-lab setting. Apart from the huge variety of activities and diversity of body parts involved in their execution, this dataset is well-suited for this evaluation since the sensor deployment matches the one presented in Fig. 11.

An analysis of the classification performance of a pre-trained activity recognizer for the various sensor configuration scenarios is conducted. In the normal scenario (denoted as "ideal") the activity recognizer operates on the acceleration data registered by the accelerometers embedded into W_1 (hereafter, ACC_1), W_3 (hereafter, ACC_3) and W_9 (hereafter, ACC_9) platforms. In the failure scenarios (denoted as "F"), one of the three sensors turns to not work properly, thus leading to three cases respectively: F-ACC_1, F-ACC_3 or F-ACC_9. The anomalous or defective behavior of the failure sensor is here modeled through a residual signal (zero signal), whilst the signals of the remaining unaffected two are kept unaltered. Finally, in the replacement scenarios (denoted as "R"), the failure sensor from the previous scenarios is replaced with one of the sensors in idle state, thus leading to the following cases: R-ACC_2, R-ACC_4, R-ACC_5, R-ACC_6, R-ACC_7 or R-ACC_8.

The activity recognizer is modeled as follows. Each acceleration sensor data stream is partitioned into non-overlaping windows of 2 s duration [55]. Each of these data windows are further characterized through a feature extraction process in which the mean, standard deviation, maximum, minimum, mean crossing rate and kurtosis are computed [56]. These features are used as input to the classifier, which is here defined through a decision tree model [57] for simplicity. Other sets of features and types of classifiers could also be used and similar results would in principle apply. However, this particular activity recognition model configuration has been proven to perform well in a number of prior studies [9,43,52,58].

In practical terms, the assessment consists in a leave-one-subject-out cross validation (LOOXV) in which the model training is performed on the ACC_1, ACC_3 and ACC_9 data of K-1 subjects (with K=17) while the model test is carried out on the sensor data from the remaining subject for the particular scenario under evaluation. The process is repeated for all the users to ensure a reliable average estimate of the performance of the recognizer in each case. Moreover, the *Fscore* [59], a combination of precision and recall measures, is used as performance metric given its robustness to class imbalancement. The *Fscore* ranges between [0,1], where 1 represents optimal recognition capabilities whilst 0 corresponds to a model which is not capable of recognition at all.

The results obtained for each of the scenarios are depicted in Fig. 16. The performance baseline is driven by the ideal case, for which a Fscore average value of 0.85 is attained. Now, in the first case (Fig. 16(a)), once ACC_1 behaves abnormally a performance drop of around 0.25 is observed, thus leading to an overall performance of 0.58. Replacing the affected sensors with another one shows no improvement in general but for the case in which the ACC_5 is used, which indeed shows an improvement of nearly 0.10.

Table 8
Metrics for the MIMU-Wear ontolgy.

	Inferred knowledge due to the rules	Inferred knowledge due to the ontology	MTx instances	MTx definition	MIMU-Wear terminology
Metrics					
Axiom	102	860	214	117	932
Logical axiom count	102	860	152	86	339
Declaration axioms count	0	0	64	31	215
Class count	0	0	0	8	111
Object property count	0	0	6	0	75
Data property count	0	0	0	0	4
Individual count	0	0	58	23	13
Class axioms					
SubClassOf	0	0	0	15	234
EquivalentClasses	0	0	0	0	1
DisjointClasses	0	0	0	0	8
Object property axioms					
SubObjectPropertyOf	0	27	4	0	38
EquivalentObjectProperties	0	59	0	0	0
InverseObjectProperties	0	0	0	0	11
FunctionalObjectProperty	0	0	0	0	8
InverseFunctionalObjectProperty	0	0	0	0	6
AsymmetricObjectProperty	0	6	0	0	0
IrrefexiveObjectProperty	0	6	0	0	0
ObjectPropertyDomain	0	0	0	0	6
ObjectPropertyRange	0	0	0	0	6
SubPropertyChainOf	0	0	0	0	2
Data property axioms					
EquivalentDataProperties	0	4	0	0	0
Individual axioms					
ClassAssertion	0	427	58	23	13
ObjectPropertyAssertion	102	331	88	25	0
DataPropertyAssertion	0	0	0	23	0
DifferentIndividuals	0	0	2	0	2
Annotation axioms					
AnnotationAssertion	0	0	0	0	378

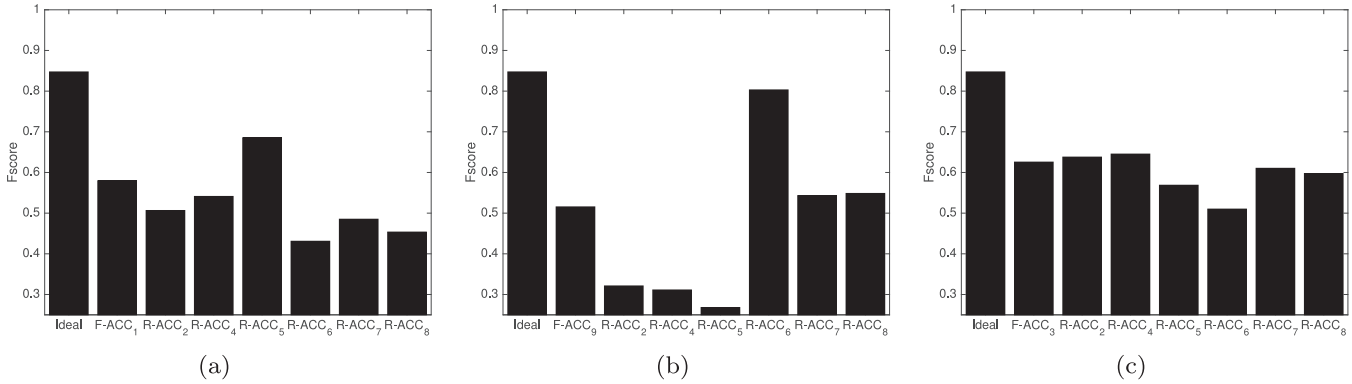

Fig. 16. Performance (Fscore) for the different sensor replacement scenarios when there is a failure of 16(a) the accelerometer in W_1 (ACC_1), 16(b) the accelerometer in W_9 (ACC_9), 16(c) the accelerometer in W_3 (ACC_3). Legend: "Ideal" = configuration ACC_1, ACC_3, ACC_9; "F-ACC_i" = same as the ideal configuration but with the ACC_i not working properly; "R-ACC_k" = same as ideal configuration but with ACC_i replaced with ACC_k.

This sensor is actually the one that would be found for replacement through MIMU-Wear as shown in Section 5.3. In the second case (Fig. 16(b)) the benefit of using an accurate sensor replacement strategy is even clearer. Once ACC_9 gets affected the performance of the recognition system drops to approximately 0.5. A random replacement could lead to even worse values, even below 0.3 as for using ACC_5 as candidate. However, an improvement of more than 0.2 is achieved while using the sensor found from the ontological search, i.e., ACC_6, thus leading to a performance close to the original one. Worse options yet providing better results than for the failure case could be ACC_7 and ACC_8, which would be the sensors suggested by the selection method if ACC_6 was not available. Finally, the third case (Fig. 16(c)) presents a situation in which little improvement is attained even if the anomalous sensor is replaced. An abnormal behavior ACC_3 seems to be not as harmful as for the other sensors, thus likely meaning this sensor is not as informative or relevant as the other two are. Anyhow, some benefit can be obtained while replacing this sensor with ACC_2 or ACC_4, which again coincide with the replacement criteria suggested by the ontological search method.

6. Conclusions

This work has presented the MIMU-Wear Ontology, an OWL 2 ontology which provides syntactic interoperability and semantic compatibility in wearable activity recognition systems. The MIMU-Wear Ontology comprehensively describes wearable sensor platforms consisting of magnetic and inertial measurement units, including the MIMUs capabilities and the characteristics of wearable sensor platforms like their on-body location.

This ontology has been used to enable the semantic sensor selection of the best sensors for the replacement of a given anomalous one. However, this is only one of the potential application scenarios for this ontology. MIMU-Wear could also be used at system startup in order to identify which sensors should be activated depending on the necessities of the activity recognizer and the aimed performance. Similarly, the ontology could be used for the self-calibration of some parameters of the sensing network according to energy constraints or efficiency goals, and based on processing power or memory resources.

The MIMU-Wear Ontology is designed in a modular manner which makes it reusable in other domains. The Wearable Sensor Platform Ontology could be used to describe the location on the human body of any wearable sensors not only of MIMUs. For example, the location of an ECG sensor in a belt could be easily described using this ontology. Similarly, the MIMU Ontology could be used to describe any MIMUs, this means not only the wearable ones but also those embedded into ambient intelligence platforms. As an example, the characteristics of a MIMU integrated into a cup or door in an ambient assisted living scenario could be thoroughly modeled using the MIMU Ontology.

Furthermore, MIMU-Wear builds on the standard W3C SSN Ontology, thus facilitating its widespread adoption since it could be directly integrated with any other ontology using SSN. In fact, the SSN Ontology has been extensively used in the research community, which already opens up a broad spectrum of ontologies in which MIMU-Wear could be integrated.

Future work includes some improvements on the MIMU-Wear Ontology such as defining more characteristics of the wearable sensor platforms, e.g., their weight, crucial for wearable systems and of potential interest for the selection process. Furthermore, the locations on the user body could be extended to make them more granular or the orientation of the MIMU included in the model for applications requiring a precise estimate of the 3D placement of the sensor.

Acknowledgments

This work was supported by the Regional Government of Andalusia Project P12-TIC-2082 and the Spanish Ministry of Economy and Competitiveness “Retos de la Sociedad” Project TIN2015-71873-R.

References

[1] H. Koskimäki, et al., Behavior modeling in industrial assembly lines using a wrist-worn inertial measurement unit, J. Ambient Intell. Hum. Comput. 4 (2) (2013) 187–194.
[2] C. Ladha, et al., Climbax: skill assessment for climbing enthusiasts, in: International Joint Conference on Pervasive and Ubiquitous Computing (Ubicomp), ACM, 2013, pp. 235–244.
[3] F. Aiolli, et al., Climbtheworld: real-time stairstep counting to increase physical activity, in: International Conference on Mobile and Ubiquitous Systems (Mobiquitous), 2014, pp. 218–227.
[4] A. Bulling, et al., A tutorial on human activity recognition using body-worn inertial sensors, ACM Comput. Surv. 46 (3) (2014) 33.
[5] M. Ermes, et al., Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions, IEEE Trans. Inform. Technol. Biomed. 12 (1) (2008) 20–26.
[6] O. Banos, et al., Daily living activity recognition based on statistical feature quality group selection, Expert Syst. Appl. 39 (9) (2012) 8013–8021.
[7] A. Mannini, et al., Activity recognition using a single accelerometer placed at the wrist or ankle, Med. Sci. Sports Exerc. 45 (11) (2013) 2193–2203.
[8] O. Banos, et al., Multi-sensor fusion based on asymmetric decision weighting for robust activity recognition, Neural Process. Lett. 42 (1) (2015) 5–26.
[9] O. Banos, M.-A. Toth, M. Damas, H. Pomares, I. Rojas, Dealing with the effects of sensor displacement in wearable activity recognition, Sensors 14 (6) (2014) 9995–10023.
[10] P. Zappi, et al., Network-level power-performance trade-off in wearable activity recognition: a dynamic sensor selection approach, ACM Trans. Embed Comput. Syst. 11 (3) (2012) 68:1–68:30.
[11] D.-S. Zois, et al., Energy-efficient, heterogeneous sensor selection for physical activity detection in wireless body area networks, IEEE Trans. Signal Process. 61 (7) (2013) 1581–1594.
[12] H. Nguyen, et al., Energy saving for activity recognition through sensor selection, scheduling and sampling rate tuning, in: IEEE Conference Wire and Mobile Networks (WMNC), IEEE, 2014, pp. 1–8.
[13] M. Janicke, et al., Self-adapting multi-sensor systems: a concept for self-improvement and self-healing techniques, in: IEEE International Conference on Self-Adapt and Self-Organ Systems (SASOW), 2014, pp. 128–136.
[14] W3C OWL Working Group, OWL 2 Web Ontology Language: Document Overview (Second Edition), W3C Recommendation, 11 December 2012. Available at http://www.w3.org/TR/owl2-overview/.
[15] M. Botts, et al., OGC Sensor Web Enablement: Overview And High Level Architecture, Technical Report, OGC, 2007.
[16] M. Botts, A. Robin, OpenGIS sensor model language (SensorML) implementation specification, Technical Report, Open Geospatial Consortium,, 2007.
[17] S. Cox, Observations and Measurements Part 1 - Observation schema (OGC 07-022r1), 2007a, (OpenGIS Implementation Standarda).
[18] S. Cox, Observations and Measurements Part 2 - Sampling features (OGC 07-002r3), 2007b, (OpenGIS Implementation Standardb).
[19] A. Sheth, et al., Semantic sensor web, IEEE Internet Comput. 12 (4) (2008) 78–83.
[20] D.J. Russomanno, C.R. Kothari, O.A. Thomas, Building a sensor ontology: a practical approach leveraging iso and ogc models, in: 2005 International Conference on Artificial Intelligence, 2005, pp. 637–643.
[21] I. Niles, A. Pease, Towards a standard upper ontology, in: Proceedings of the International Conference on Formal Ontology in Information Systems, 2001.
[22] S. Nikolić, V. Penca, M. Segedinac, Z. Konjović, Semantic web based architecture for managing hardware heterogeneity in wireless sensor network, Int. J. Comput. Sci. Appl. 8 (2) (2011) 38–58.
[23] M. Compton, et al., The SSN ontology of the W3C semantic sensor network incubator group, J. Web Seman. 17 (0) (2012).
[24] J. Soldatos, et al., Openiot: open source internet-of-things in the cloud, in: FP7 OpenIoT Project Workshop, 9001, 2015, pp. 13–25.
[25] N. Kefalakis, et al., A visual paradigm for iot solutions development, in: FP7 OpenIoT Project Workshop, 9001, 2015, pp. 26–45.
[26] P. Desai, et al., Semantic gateway as a service architecture for iot interoperability, in: IEEE International Conference on Mobile Services (MS), 2015, pp. 313–319.
[27] A. Ibrahim, et al., Geospatial ontology-based mission assignment in wireless sensor networks, in: International Conference on Internet Services of Things (RIoT), 2015.
[28] R. Bendadouche, C. Roussey, G. De Sousa, J.-P. Chanet, K.M. Hou, Extension of the semantic sensor network ontology for wireless sensor networks: the stimulus-wsnnode-communication pattern, in: Proceedings of the 5th International Conference on Semantic Sensor Networks - Volume 904, in: SSN'12, CEUR-WS.org, Aachen, Germany, Germany, 2012, pp. 49–64.
[29] L. Nachabe, et al., Unified data model for wireless sensor network, IEEE Sens. J. 15 (7) (2015) 3657–3667.
[30] L. Nachabe, et al., Semantic smart home system: ontosmart to monitor and assist habitant, Int. J. Comput. Commun. 10 (2016) 78–86.
[31] ETSI TC SmartBAN, Smart Body Area Networks (SmartBAN) Unified data representation formats, semantic and open data model. ETSI TS 103 378 V1.1.1 (2015–12), ETSI, 2015.
[32] X. Wang, H. An, Y. Xu, S. Wang, Sensing network element ontology description model for internet of things, in: International Conference on Information Science and Control Engineering, 2015.
[33] V. Tsiatsis, et al., The sensei real world internet architecture., in: Future Internet Assembly, 2010, pp. 247–256.
[34] C. Villalonga, et al., A resource model for the real world internet, in: European Conference on Smart Sensing and Context (EuroSSC), Springer, 2010, pp. 163–176.
[35] Y.-C. Hsu, et al., Design of a sensing service architecture for internet of things with semantic sensor selection, in: IEEE International Conference on Ubiquitous Intelligence and Computing, 2014, pp. 290–298. Cited By 0.
[36] C. Perera, et al., Sensor search techniques for sensing as a service architecture for the internet of things, IEEE Sens. J. 14 (2) (2014) 406–420.
[37] protégé, (2016), (Available online: http://protege.stanford.edu/ (Accessed: 07.10.16)).
[38] D. Peterson, et al., W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes, W3C Recommendation, 5 April 2012. Available at http://www.w3.org/TR/xmlschema11-2/.
[39] Foundational Model of Anatomy ontology (FMA), http://sig.biostr.washington.edu/projects/fm/AboutFM.html.
[40] C. Mungall, et al., Uberon, an integrative multi-species anatomy ontology, Genome Biol. 13 (1) (2012) 1–20.
[41] Linked data, http://linkeddata.org/.
[42] I. Horrocks, et al., SWRL: A Semantic Web Rule Language Combining OWL and RuleML, W3C Member Submission, 2004. Available at http://www.w3.org/Submission/SWRL/.
[43] O. Banos, et al., Kinect=IMU? Learning MIMO Signal Mappings to Automatically Translate Activity Recognition Systems across Sensor Modalities, in: International Symposium on Wearable Comp (ISWC),, 2012, pp. 92–99.
[44] S. Harris, SPARQL 1.1 (SPARQL Query Language for RDF), W3C Recommendation, 2013. Available at http://www.w3.org/TR/sparql11-query/.

[45] H. Sagha, H. Bayati, J. del R. Millán, R. Chavarriaga, On-line anomaly detection and resilience in classifier ensembles, Pattern Recogn. Lett. 34 (15) (2013a) 1916–1927.
[46] H. Sagha, A. Calatroni, J. del R. Millán, D. Roggen, G. Tröster, R. Chavarriaga, Robust activity recognition combining anomaly detection and classifier retraining, in: Proceedings of the 10th annual conference on Body Sensor Networks (BSN), 2013b.
[47] XM-B Technical Documentation, Xsens Technologies B.V., 2009. http://www.xsens.com.
[48] O. Lara, et al., A survey on human activity recognition using wearable sensors, IEEE Commun. Surv. Tuts 1 (99) (2012) 1–18.
[49] J. Parkka, et al., Activity classification using realistic data from wearable sensors, IEEE Trans. Inf. Technol. Biomed. 10 (1) (2006) 119–128.
[50] K. Altun, et al., Human activity recognition using inertial/magnetic sensor units, in: Human Behavior Understanding, 2010, pp. 38–51.
[51] H. Gjoreski, M. Gams, Accelerometer data preparation for activity recognition, in: International Multiconference Information Society, 2011.
[52] O. Banos, et al., Design, implementation and validation of a novel open framework for agile development of mobile health applications, Biomed. Eng. Online 14 (S2:S6) (2015) 1–20.
[53] S. Tartir, I.B. Arpinar, A.P. Sheth, Ontological Evaluation and Validation, Springer Netherlands, pp. 115–130.
[54] O. Banos, et al., A benchmark dataset to evaluate sensor displacement in activity recognition, in: International Joint Conference on Pervasive and Ubiquitous Computing (Ubicomp), ACM, 2012, pp. 1026–1035.
[55] O. Banos, J.-M. Galvez, M. Damas, H. Pomares, I. Rojas, Window size impact in human activity recognition, Sensors 14 (4) (2014) 6474–6499.
[56] J.R. Kwapisz, et al., Activity recognition using cell phone accelerometers, Conf. Knowl. Disc Data Min. 12 (2) (2011) 74–82.
[57] R.O. Duda, et al., Pattern Classification (2nd Edition), Wiley-Interscience, 2000.
[58] O. Banos, et al., On the use of sensor fusion to reduce the impact of rotational and additive noise in human activity recognition, Sensors 12 (6) (2012) 8039–8054.
[59] M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, Inf. Process. Manag. 45 (4) (2009) 427–437.

Claudia Villalonga (*Master's Degree in Telecommunications Engineering* in 2006 by the Universitat Politècnica de Catalunya, *Master in Business Management (Administration of Organizations in the Knowledge Economy*) in 2016 by the Universitat Oberta de Catalunya, and PhD on Information and Communication Technologies in 2016 by the University of Granada) is a postdoctoral researcher at the Research Center for Information and Communications Technologies of the University of Granada (CITIC-UGR). With more than 10 years in the information technology industry, Claudia Villalonga has a wide range of experience in research and innovation in the areas of ontologies and semantics, knowledge engineering, context awareness and context recognition. Claudia Villalonga has worked in international environments for the research and innovation centers of several major multinational enterprises: as R&D project manager at the Canadian consulting company CGI in Spain, as research associate at the German software company SAP in Switzerland, and as software engineering at the Japanese technological company NEC in Germany. Moreover, Claudia Villalonga has worked for almost four years as a research associate at the prestigious Swiss university ETHZ (Swiss Federal Institute of Technology Zurich) and for one year as senior researcher at the Korean university KHU (Kyung Hee University).

Héctor Pomares (MSc in Electrical Engineering in 1995, MSc in Physics in 1997, PhD from the University of Granada in 2000, all of them with honors) is currently a full professor at the University of Granada. He has published more than 60 articles in the most prestigious scientific (JCR-indexed) journals and contributed with more than 150 papers in international conferences. He has participated in 15 national projects of which he has been the leading researcher in 5. He has been a visiting researcher (longer than 1 month) in the following universities: University of Dortmund (Germany), University of California at Berkeley (USA), University of Texas A & M (USA), University of Applied Sciences Muenster (Germany), Technical University of Graz (Austria) and University of Amsterdam (Netherlands). He has also participated in 8 contracts signed for innovative research through the Office of Transfer of Research Results (TTO) and has been principal investigator in 5 contracts through the University of Granada Foundation Company for an amount totaling more than 1.2 million euros. At the present time, he is the director of the Official Doctoral Program in ICT at the University of Granada.

Ignacio Rojas received the B.Sc. degree in electronic physics in 1992 and the Ph.D. degree in Intelligent Systems with honors in 1996, both from the University of Granada, Spain. He has been visiting professor at the University of Dortmund (Germany) at the Department of Electrical Engineering, as a visiting researcher with the BISC Group of Prof. L.Zadeh, University of California, Berkeley and as visiting professor at the Muenster University of Applied Sciences, Germany. He is currently Full Professor with the Department of Computer Architecture and Computer Technology, University of Granada (he started in 1994 being assistant professor). He is director of the Spanish Chapter of the IEEE Computational Intelligence Society, and actually he is also director of the Information and Communications Technology Centre (CITIC-UGR) University of Granada, Spain.
His research interests include hybrid system, hardware–software implementation, combination of intelligent system for adaptive control, self-organizing neuro-fuzzy systems, neural networks, time series forecasting, e-monitoring, e-health, bioinformatics, biomedical engineering, data mining and architectures for complex optimization problems.

Oresti Baños (MSc in Telecommunications Eng. 2009, MSc in Computer Network Eng. 2010, MSc in Electrical Eng. 2011, PhD in Computer Science 2014, all with honors from the University of Granada, Spain) is currently working as an Assistant Professor of Creative Technology at the University of Twente, Netherlands. He is also a Research Scientist with the Biomedical Signal and Systems group, the Centre for Telematics and Information Technology, the Research Centre for Biomedical Technology and Technical Medicine and the Centre for Monitoring and Coaching. He is a former Research Associate at Kyung Hee University (South Korea) and the Research Center for Information and Communications Technologies of the University of Granada (Spain). He has been Visiting Researcher at several prestigious institutions such as the Swiss Federal Institute of Technology Zurich (Switzerland), the University of Alabama (USA) and the Technical University of Eindhoven (Netherlands). His main research is on wearable, ubiquitous, pervasive and mobile computing with a particular focus on digital health and wellbeing applications. His expertise also covers pattern recognition and machine learning for probabilistic modeling of human behavior from multi-modal sensor data, information fusion and context-awareness, with a special interest in robust, adaptive and opportunistic expert systems.