Ontology Engineering and Reasoning to Support Real World Human Behavior Recognition

A dissertation submitted to

Universidad de Granada

for the
Title of Doctor

presented by

Claudia Villalonga Palliser

Programa de Doctorado en
Tecnologías de la Información y la Comunicación

Editor: Universidad de Granada. Tesis Doctorales
Autora: Claudia Villalonga Palliser
ISBN: 978-84-9163-061-6
URI: http://hdl.handle.net/10481/44536

El doctorando / *The doctoral candidate* D.ª Claudia Villalonga Palliser y los directores de la tesis / *and the thesis supervisors* D. Héctor Pomares Cintas y D. Oresti Baños Legrán.

Garantizamos, al firmar esta tesis doctoral, que el trabajo ha sido realizado por el doctorando bajo la dirección de los directores de la tesis y hasta donde nuestro conocimiento alcanza, en la realización del trabajo, se han respetado los derechos de otros autores a ser citados, cuando se han utilizado sus resultados o publicaciones.

/

Guarantee, by signing this doctoral thesis, that the work has been done by the doctoral candidate under the direction of the thesis supervisors and, as far as our knowledge reaches, in the performance of the work, the rights of the other authors to be cited (when their results or publications have been used) have been respected.

Lugar y fecha / *Place and date*:

Granada, a 18 de octubre de 2016.

Directores de la Tesis / *Thesis supervisors*:

Firma / *Signed*

Doctorando / *Doctoral candidate*:

Firma / *Signed*

Table of Contents

Resumen

Introducción

El reconocimiento de la conducta humana [1] ha despertado mucho interés recientemente debido a su aplicación para impulsar cambios de comportamiento en los dominios de la salud y el bienestar [2]. No obstante, la mayoría de los sistemas de reconocimiento del comportamiento humano disponibles hasta la fecha sufren dos limitaciones que los hacen poco adecuados para trabajar en el mundo real: están diseñados para trabajar sobre una configuración predeterminada de sensores y para ser usados en el dominio de una aplicación específica. Sin embargo, los sistemas de reconocimiento que trabajan en condiciones reales están sujetos a fallos o defectos de los sensores [3] y a cambios en el despliegue [4] que son imprevisibles durante la fase de diseño y que afectan directamente al buen funcionamiento del sistema de reconocimiento. Asimismo, el reconocimiento de expresiones más informativas sobre el contexto humano requiere analizar múltiples componentes del comportamiento tales como los aspectos físicos y mentales [1], los cuales por el momento solo se han investigado de forma aislada.

En vista de estas limitaciones, existe la necesidad de (1) describir exhaustivamente el conjunto heterogéneo de recursos que participan en el sistema de reconocimiento de la conducta humana, (2) seleccionar dinámicamente los sensores de recambio para garantizar la continuidad del reconocimiento, (3) describir de forma íntegra la información sobre el contexto humano, y (4) inferir automáticamente expresiones descriptivas del contexto para el análisis del comportamiento. Las ontologías [5] son descripciones formales que tienen una semántica implícita, lo que proporciona interoperabilidad y permite el razonamiento. Además las ontologías superan los modelos no-semánticos en términos de flexibilidad, extensibilidad, generalidad, expresividad y en el desacoplamiento del conocimiento del código. Por lo tanto, existe una clara oportunidad de mejorar los sistemas de reconocimiento de la conducta mediante el uso de ontologías y razonamiento ontológico. Por eso, el objetivo de esta tesis es investigar sobre la posible aplicación de ontologías y razonamiento ontológico con el fin de resolver algunas de las limitaciones más importantes a las que los sistemas de reconocimiento del comportamiento humano se ven sujetos durante su operación en condiciones reales.

Selección de sensores basada en ontologías para el reconocimiento continuo del comportamiento

Esta tesis presenta MIMU-Wear, una ontología OWL 2 [6] que proporciona interoperabilidad sintáctica y compatibilidad semántica para los sistemas de reconocimiento del comportamiento. MIMU-Wear describe exhaustivamente las plataformas vestibles (también conocidas como "wearables") equipadas con sensores inerciales y magnéticos (conocidos como MIMU por su nombre en inglés "magnetic and inertial measurement units"). Esta ontología describe las capacidades de los MIMUs por ejemplo sus propiedades de medición y las características de las plataformas vestibles sensorizadas, incluyendo su localización en el cuerpo y sus propiedades de supervivencia. MIMU-Wear ofrece una semántica implícita que permite la interpretación automática de las descripciones de los recursos, la abstracción de la tecnología subyacente y la abstracción del método de selección de sensores de la infraestructura del sistema de reconocimiento.

La ontología MIMU-Wear construye sobre SSN [7], una ontología estándar del W3C, y está diseñada de una manera modular conectando varias ontologías para dominios específicos: la MIMU Ontology describe las características de los MIMUs; la MIMU Capabilities Ontology modela las capacidades de medición o sensado de los MIMUs; la MIMU Magnitudes Ontology representa las diferentes magnitudes observadas por los MIMUs; la MIMU Units Ontology representa las unidades de medida necesarias para describir las capacidades de los MIMUs; la Wearable Sensor Platform Ontology modela las características de las plataformas vestibles sensorizadas; la Human Body Ontology modela las partes del cuerpo humano que representan las localizaciones donde se portan las plataformas vestibles; y la Wearable Survival Range Ontology modela las condiciones de supervivencia los sistemas vestibles.

La modularidad de MIMU-Wear permite que esta ontología seareutilizable en otros dominios. La Wearable Sensor Platform Ontology se podría usar para describir la localización en el cuerpo humano de cualquier sensor vestible, no sólo de MIMUs, por ejemplo la ubicación en el tórax de una banda para medir electrocardiograma. Del mismo modo, la MIMU Ontology se podría utilizar para describir cualquier MIMU, es decir no sólo los vestibles sino también los incluidos en las plataformas de inteligencia ambiental. Por ejemplo, las características de un MIMU integrado en una taza o una puerta en un escenario de inteligencia ambiental podrían modelarse fácilmente usando la MIMU Ontology. Además, el hecho de que MIMU-Wear monte sobre SSN, una ontología estándar del W3C y ampliamente utilizada por la comunidad científica, facilita la adopción generalizada de MIMU-Wear, ya que podría integrarse directamente con cualquier otra ontología que haga uso de SSN.

Esta tesis propone un nuevo método que permite seleccionar dinámicamente sensores de recambio entre los disponibles en las plataformas vestibles para cuando un MIMU perteneciente al sistema de reconocimiento del comportamiento sufre alguna anormalidad y necesita ser remplazado. Este método de selección de sensores se basa en la ontología MIMU-Wear y aplica técnicas de razonamiento ontológico y de consultas a la ontología. Las reglas SWRL [8] definen las características de los sensores candidatos a remplazar un MIMU en el sistema de reconocimiento y permiten inferir que sensores son buenos candidatos y cuáles no. El método de selección de sensores establece el proceso de ejecución iterativa de diferentes consultas SPARQL [9] sobre las descripciones ontológicas de los MIMUs con el fin de seleccionar el MIMU más adecuado para la sustitución del defectuoso. El método de consultas iterativas permite que si no se encuentra ningún resultado para una consulta, se ejecute otra menos restrictiva o con un criterio de búsqueda diferente.

La evaluación del método de selección de sensores en un escenario realista en el área de reconocimiento de la actividad demuestra que la sustitución de un MIMU anómalo asegura la continuidad de reconocimiento. Es decir, la fiabilidad del sistema de reconocimiento se recupera con respecto a la situación de fallo después de la sustitución del sensor anómalo. Para el caso de estudio, la fiabilidad del sistema de reconocimiento cae más de un tercio con respecto a su valor de referencia cuando uno de los sensores falla. La sustitución del sensor afectado con el sensor seleccionado a través del método ontológico muestra una mejoría, que en el mejor de los casos permite restaurar prácticamente las capacidades de reconocimiento del sistema y en el peor de los casos consigue al menos que la fiabilidad del sistema supere la que se obtendría en caso de seleccionar el sensor de remplazo de forma arbitraria.

El método de selección de sensores propuesto en esta tesis ayuda a mantener el sistema de reconocimiento funcionando de forma continua aunque los sensores sufran alguna anomalía. Sin embargo, este es sólo uno de los posibles escenarios en los que se puede aplicar la ontología MIMU-Wear y el método basado en consultas ontológicas. MIMU-Wear también se podría utilizar al arranque del sistema de reconocimiento para identificar qué sensores deben ser activados en función de la fiabilidad esperada del sistema y del rendimiento pretendido. Del mismo modo, la ontología se podría utilizar para la auto-calibración de algunos parámetros de la red de sensores de acuerdo con las restricciones de energía u objetivos de eficiencia. En todos estos escenarios, un método basado en el razonamiento ontológico y la consulta de la ontología similar al que se propone en esta tesis se podría aplicar de forma sencilla.

Inferencia de contexto basada en ontologías para el análisis del comportamiento humano

Esta tesis presenta la Mining Minds Context Ontology, una ontología OWL 2 para modelar de forma exhaustiva expresiones descriptivas de contexto. Esta ontología está diseñada para modelar los contextos más comunes en escenarios de salud y bienestar y que se dan en estilos de vida sedentarios y activos. Por lo tanto, esta ontología modela múltiples primitivas de contexto, tales como la actividad física, la locación y la emoción, así como contextos más abstractos, tales como inactividad, hacer ejercicio, trabajo en la oficina o comer, los cuales pueden derivarse de la combinación de estas primitivas.

La Mining Minds Context Ontology permite representar cualquier combinación de primitivas de contexto (contextos de bajo nivel), incluso para diferentes dominios, con el fin de inferir representaciones más abstractas de contexto (contextos de alto nivel). La incorporación, sin precedentes hasta la fecha, de las emociones en la definición del contexto permite representar contextos de alto nivel que sólo pueden identificarse cuando la persona evidencia una emoción específica. No obstante, para asegurar su aplicabilidad en múltiples escenarios, la ontología ha sido definida de forma que se permita la identificación de algunos contextos de alto nivel incluso en ausencia de información sobre las emociones.

Esta tesis presenta un método basado en ontologías para derivar información de contexto de alto nivel de la combinación de varios contextos de bajo nivel. Este nuevo método se basa en la Mining Minds Context Ontology y aplica razonamiento OWL 2 DL para inferir contexto de alto nivel a partir de primitivas básicas de contexto de bajo nivel. La High-Level Context Architecture es la arquitectura del sistema que implementa el método de identificación de contexto y que permite inferir automáticamente y en tiempo real expresiones descriptivas del contexto. La High-Level Context Architecture consta de cuatro componentes: el High-Level Context Builder que genera los conceptos ontológicos para representar el contexto del usuario; el High-Level Context Reasoner que verifica y clasifica el contexto de alto nivel; el High-Level Context Notifier que notifica a terceros sobre la identificación de un nuevo contexto de alto nivel; y el Context Manager que almacena la información de contexto.

La evaluación del método de inferencia de información más descriptiva del contexto permite demostrar que con esta información se puede mejorar el funcionamiento de los sistemas de reconocimiento de la conducta. El método propuesto no sólo resulta eficaz para derivar nueva información contextual sino que también es robusto a los errores introducidos por los sistemas de reconocimiento de los contextos de bajo nivel. De hecho, los estudios desarrollados en este trabajo demuestran que el error introducido en el contexto de bajo nivel tiene un menor impacto en los contextos de alto nivel.

La High-Level Context Architecture es el motor del proceso de inferencia de comportamiento abstracto en Mining Minds [10], una plataforma digital en el ámbito de la salud y el bienestar. A pesar de que el método de inferencia de contexto ha sido ideado para esta plataforma, la High-Level Context Architecture se ha definido de una manera que permite su uso en cualquier otro ámbito. De hecho, en caso de que el método de inferencia del contexto tuviera que aplicarse a un nuevo dominio y eso requiriera identificar nuevos contextos, la High-Level Context Architecture no necesitaría modificación alguna y sólo se tendría que extender la ontología. Esto se debe a una de las propiedades principales de las ontologías: el desacoplamiento del conocimiento y del código.

Conclusiones

El propósito de esta tesis fue investigar sobre la posible aplicación de ontologías y razonamiento ontológico con el fin de resolver algunas de las limitaciones más importantes de los sistemas de reconocimiento del comportamiento humano cuando son operados en condiciones reales. Para ello se definieron cuatro objetivos que se han se han alcanzado con éxito. Concretamente, las contribuciones de esta tesis son las siguientes:

1. MIMU-Wear: Una ontología OWL 2 modular que describe exhaustivamente las plataformas vestibles equipadas con sensores MIMU.

2. MIMU Ontology: Una ontología OWL 2 que describe las características de los MIMUs, por ejemplo sus propiedades de medición.

3. Wearable Sensor Platform Ontology: Una ontología OWL 2 que modela las características de las plataformas vestibles sensorizadas, incluyendo su localización en el cuerpo y sus propiedades de supervivencia.

4. Human Body Ontology: Una ontología OWL 2 que modela las partes del cuerpo humano.

5. Un método basado en la ontología MIMU-Wear, las reglas SWRL y las consultas SPARQL que permite seleccionar sensores de forma dinámica para ayuda a mantener el sistema de reconocimiento funcionando de forma continua.

6. Mining Minds Context Ontology: Una ontología OWL 2 para modelar de forma exhaustiva expresiones descriptivas de contexto. Esta ontología está disponible en http://www.miningminds.re.kr/icl/context/context-v2.owl.

7. Un método basado en la Mining Minds Context Ontology y en el razonamiento OWL 2 DL para la inferencia automática de contexto más descriptivo que permite explicar mejor el comportamiento.

8. HLCA: Una arquitectura del sistema que implementa el método de identificación de contexto y su realización en Java.

Estas contribuciones suponen un primer paso hacia una nueva generación de sistemas de reconocimiento del comportamiento humano para el mundo real. Algunas de las posibles líneas de trabajo futuras versan sobre la incorporación de los sensores disponibles en textiles inteligentes al proceso de selección, la consideración de la incertidumbre en el reconocimiento del comportamiento humano y la provisión de interoperabilidad entre diversos sistemas de reconocimiento.

Abstract

Human behavior recognition has attracted much attention during the recent years due to its multiple applications in the health and wellness domain. Despite their popularity, most existing behavior recognition systems suffer two main constrains which make them practically unsuitable to work in real-world conditions: they are bounded to a specific sensor deployment setup and they are defined to operate in a single application domain. Human behavior recognition systems may certainly undergo changes unforeseen at system design time, with sensors subject to diverse types of anomalies such as failures or deployment changes; thus, a pre-defined, well-known and steady sensor setup cannot be guaranteed. Moreover, the categories of behavior recognized by these systems tend to be quite primitive and with limited applicability; however, their appropriate combination could lead to more meaningful and richer expressions of context for human behavior analysis. In the light of these limitations, there is a clear necessity of comprehensively describing the set of heterogeneous resources involved in the human behavior recognition system, dynamically selecting replacement sensors to ensure continuity of recognition, exhaustively describing human context information, and automatically inferring meaningful and rich expressions of context for human behavior analysis.

This thesis investigates novel mechanisms to solve the above limitations of human behavior recognition systems in order to facilitate their seamless, robust and accurate use in realistic conditions. Ontologies are considered here to be the cornerstone technology to realize this idea. The extraordinary characteristics of ontologies, which provide implicit semantics, support interoperability and enable automatic reasoning, fit particularly well with the necessities posed by the problem considered here. Besides, ontologies largely exceed other similar and non-semantic models in terms of flexibility, extensibility, generality, expressiveness, and decoupling of the knowledge from the implementation, thus making it a perfect option to create more advanced behavior-aware systems.

This work proposes MIMU-Wear, an OWL 2 ontology which comprehensively describes mainstream wearable sensor platforms consisting of magnetic and inertial measurement units (MIMUs), including the MIMUs capabilities and the characteristics of the wearable sensor platform. This ontology provides implicit semantics enabling the automatic interpretation of the resource descriptions, their abstraction from

the underlying technology, and the abstraction of the sensor selection method from the actual sensing infrastructure. The dynamic selection of sensors is enabled through ontology reasoning and querying. The proposed sensor selection method builds on the MIMU-Wear Ontology, applies ontological reasoning to infer candidate replacement sensors from a set of heuristic SWRL rules, and iteratively poses SPARQL queries on the ontological sensor descriptions to select the most appropriate MIMU for the replacement of an anomalous one. The proposed ontology-based sensor selection method proves to ensure continuity of recognition as it helps recovering the system capabilities after the replacement takes place. MIMU-Wear could also serve at system startup to identify which sensors should be activated based on the necessities of the behavior recognition system or 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.

This thesis further proposes the Mining Minds Context Ontology, an OWL 2 ontology for exhaustively modeling rich and meaningful expressions of context. This ontology enables any combination of cross-domain behavior primitives, also referred to as low-level contexts, in order to infer more abstract human context representations, also called high-level contexts. The context ontology extends beyond the state-of-the-art while uniting emotion information as a novel behavioral component together with activity and location data to model new contextual information. An ontological method based on descriptive logic is developed for deriving high-level context information out of the combination of cross-domain low-level context primitives, namely activities, locations and emotions. The proposed method not only proves efficient while deriving new contextual information but also robust to potential errors introduced by low-level contexts misrecognitions. This method can be used for determining any type of high-level context information from diverse sources of low-level context data. Thus, it can be easily applied to any new domain, only requiring the extension of the ontology itself.

The proposed models and methods enable comprehensive descriptions and dynamic selection mechanisms for heterogeneous sensing resources to support the continuous operation of behavior recognition systems; likewise, exhaustively descriptions and automatic inference of abstract human context information is supported to enhance the operation of behavior-aware systems. Hence, these ontologies and ontology reasoning-based methods pave the path to a new generation of behavior recognition systems readily available for their use in the real-world.

1

Introduction

1.1. Human Behavior Recognition

From a behavioral science perspective, the behavior of an organism, also a human being, can be defined as “everything it does, including covert actions like thinking and feeling” [1]. According to this definition, to explain human behavior, one must understand the interactions between the individual and their environment, including physical, cognitive and social aspects. Behavior analysis is a comprehensive approach to the study of the behavior of organisms and its primary objectives are “the discovery of principles and laws that govern behavior, the extension of these principles over species, and the development of an applied technology” [1]. Some reasons to study human behavior are applying behavioral principles to raising and educating children, building humanoid robots able to adjust their behavior based on consequences or feedback, and modifying aspects of human behavior in self-control and coaching scenarios. In fact, human behavior change has particularly drawn the attention of research and industry lately due to its enormous application potential in areas such as health and wellbeing [2].

Human behavior recognition, the task of detecting and identifying the people’s conducts, is an essential step for the analysis of human behavior. The use of information and communication technologies has increasingly been fostered during recent years to facilitate the automatic and seamless recognition of human behavior. Some commercial behavior-aware products assess the number of steps the user took, their intake and burnt of calories, the quality of their sleep or their stress level. Fitbit Surge [11], Jawbone® UP3™ [12], Garmin vívofit®3 [13], and Empatica Embrace [14] are some examples of behavior tracking solutions that recently made their journey to the market. However, the most prominent behavior-aware systems have been provided at research level. These prototypes tackle more advanced and complex problems of behavior-awareness in the health and wellness domains, such as detecting cardiovascular illnesses [15], alerting about physical conditions [16], tracking changes in the physiological responses of patients with chronic diseases [17], persuading people to change their unhealthy life habits [18], and coaching them in fitness and sports [19].

Behavior recognition systems normally consist of two different parts embedded into a single or multiple hardware components: (1) electronic sensor devices capable of measuring and translating human physical and physiological responses into digital data; and (2) digital processing systems in charge of the gathering, storage and analysis of the data.

Diverse sensing technologies have been explored in behavior recognition systems to identify some primitive components of human behavior. Video systems have been considered for the recognition of physical activity [20] and the analysis of facial expressions [21]. Audio recorders and microphones have been used to detect speech and recognize some emotional states from speech [22]. Positioning technologies, like GPS and indoor navigation systems, have been utilized to track the user location and derive movement patterns [23]. Wearable sensors measuring physical and physiological human characteristics, such as body motion or vital signs, have extensively been considered for the recognition of physical activity [24]. In fact, 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. MIMUs have been used to determine the physical activity from body motion [25, 26] and to measure cardiac and respiratory parameters, which could indicate cognitive load and stress, from body motion [27, 28].

The automatic processing and analysis of the measured signals to recognize some primitive components of human behavior can follow different approaches: data-driven, knowledge-driven and hybrid methods. In data-driven approaches, signal processing and machine learning techniques are used to detect patterns matching some known categories of behavior. For example, extracting places and activities from GPS traces using hierarchical conditional random fields [29], recognizing activities of daily living from acceleration data based on statistical feature quality group selection [30] and recognizing emotions from speech using support vector machines [31]. In knowledge-driven approaches, logical representations of knowledge like ontologies and rules are utilized to model and infer different human behaviors. For example, detecting emotions using a multimodal approach based on rules [32] and recognizing activities of daily living based on ontologies [33]. Both data-driven and knowledge driven techniques are further combined in hybrid methods to determine some components of human behavior like activity [34] or location [35].

1.2. Limitations of Real World Human Behavior Recognition Systems

Although several systems recognizing human behavior have been proposed to date, as presented in Section 1.1, most of these systems do not fulfill the requirements posed by their use in real-world scenarios. In fact, existing behavior recognition system suffer two main limitations: (1) these systems are bounded to a specific sensor deployment setup; and (2) these systems are applied to a specific application domain.

1.2.1. Sensor Setup-Specific Systems

Behavior recognition systems are mainly 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 the recognition system may undergo changes unforeseen at system design time and sensors may be subject to diverse types of anomalies. Sensors may suffer some failures, i.e., they may be subject to damage due to their regular use or the environmental conditions, and in worst case they may break or stop working [3]. Sensors may also undergo deployment changes, i.e., variations introduced by the people normal usage of the sensor such as displacements on the position where the wearable sensor platform is located on the user's body [36, 4]. Consequently, methods supporting the dynamic selection and replacement of sensors are required in order to ensure the fully functional operation of human behavior recognition systems in realistic conditions.

Very few solutions have been proposed for the dynamic sensor selection problem mainly combining probabilistic and machine-learning driven strategies to identify the best combination of sensors. For example, fusion techniques are used to select a subset of sensor data streams to minimize power consumption while keeping a certain level of recognition [37]. Similarly, stochastic methods are applied to tune the recognition system through sensor selection [38]. Other approaches, not only explore the number of sensors to be used, but also the number of samples to be collected from them by using scheduling and down-sampling approaches [39]. 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 the sensors and devices. 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 [40].

Ensuring the operational continuity of human behavior recognition systems working in dynamic environments requires advanced sensor replacement functionalities. Mechanisms to abstract the selection of the most adequate resources are needed to enable the sensor replacement functionalities. Precisely, a comprehensive and interoperable description of the heterogeneous resources is required, including aspects such as the sensors' characteristics, their deployment and their availability. These resource descriptions in combination with sophisticated search techniques could support the selection of the best replacement for an anomalous sensor.

1.2.2. Domain-Specific Systems

The vast majority of existing solutions relying on digital technologies for automatic human behavior recognition are domain specific and apply to a sole dimension of behavior. In other words, most human behavior recognition systems are only capable of identifying some physical activity-related parameters, such as step counts or calories burned; some location-related information like movement patterns; or some mental-related parameters, such as emotions or stress. While these primitives could be considered in isolation for a preliminary analysis of a person's behavior, their appropriate combination can lead to more meaningful and richer expressions of context for human behavior analysis. Consequently, there is a need for developing new methods for the automatic identification of richer human context information which better describes human behavior and which may enhance the operation of behavior-aware systems in realistic conditions.

Efforts are being put towards the creation of commercial frameworks capable of digesting different types of behavior-related contextual information, such as Google Fit [41] and Apple HealthKit [42]. However, these initiatives rely on third party applications and systems for inferring behavior information. Some other attempts have been made in research towards a more comprehensive and multifaceted recognition of human behavior. For example, a middleware that recognizes activities and indoor mobility has been proposed to support pervasive elderly homecare [43]. An approach combining motion reconstruction, location detection and activity identification has been developed for profiling the user's daily life [44]. A platform to gather users' psychological, physiological and activity information has been proposed for analyzing their mental health [45]. Despite the availability of solutions which combine different primitive components of human behavior, these systems are still far away from identifying rich human context information. Even more important is the fact that these solutions are bound to a specific application domain since the information they provide cannot be in principle automatically interpreted and used in different behavior-aware systems.

Recognizing meaningful and rich expressions of human context, which cover multiple primitive components of behavior such as physical and mental aspects and which can be used by any behavior-aware system working in realistic conditions, requires an extensive and interoperable description of human context information. Such description is needed to enable novel mechanisms which automatically infer richer human context out of more basic context primitives. Thus, the combination of the comprehensive and interoperable description of human context information and the advanced methods for the automatic recognition of rich human context could enhance the real-world operation of behavior-aware systems.

1.3. Motivation for the Use of Ontologies to Support Real World Human Behavior Recognition

In the light of the limitations of human behavior recognition systems presented in Section 1.2, there is a clear necessity of (1) comprehensively describing the set of heterogeneous resources involved in the human behavior recognition process, (2) dynamically selecting replacement sensors to ensure continuity of recognition, (3) exhaustively describing human context information, and (4) automatically inferring meaningful and rich expressions of context for human behavior analysis.

The most simplistic approach for describing heterogeneous resources in human behavior recognition systems would consist in using text, like the descriptions of the sensor characteristics in specification sheets. However, such a solution is only viable if the dynamic selection of replacement sensors is done by humans. Likewise, free-text tags describing the resources are also insufficient for any machine-based interaction, where the dynamic selection of replacement sensors has to be executed automatically. In this case, the syntax and semantics of the resource 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 provides implicit semantics which enable the automatic interpretation of the data. An *ontology* is a formal explicit specification of a shared conceptualization or model of the world, i.e., a machine-readable representation of consensual knowledge [5]. Thus, an ontology is a predefined specification of terms that define concepts, relationships and constraints within a domain, in this case in the human behavior domain.

Using ontological models, the resource descriptions for heterogeneous sensors of different vendors are sufficiently rich to be automatically interpreted. This property of ontologies ensures interoperability among diverse human behavior recognition systems. Moreover, these features of the ontological descriptions enable the abstraction of the heterogeneous resources from their underlying technology. Hence, mechanisms to automatically select the most adequate resources, which match several given conditions, can be easily applied. Actually, the ontological descriptions of the available resources can be searched to find a replacement for an anomalous sensor by applying ontology querying mechanisms. In the process of answering the ontological query, reasoning is intrinsically applied to infer new knowledge from pre-defined rules and to verify its consistency. Therefore, ontological models in combination with rules and ontological querying can in theory support the dynamic selection of replacement sensors to ensure continuity of recognition. In conclusion, ontologies seem to be the perfect candidate to comprehensively describe heterogeneous resources abstracting them from the underlying technology and to support sensor replacement functionalities in human behavior recognition systems.

Following the same line of argument, ontologies can also be considered to exhaustively describe context information for human behavior analysis. The implicit semantics provided by ontologies enable the derivation of new context information from existing one, a key characteristic for the recognition of richer human context out of more basic context primitives. Moreover, ontological context information indentified by diverse third party recognition systems can be automatically interpreted and used to infer more meaningful context. This is of utmost importance to provide interoperability among different behavior-aware systems. Ontological reasoning can be inherently applied on ontology-based models; thus, simplifying the inference task. Several reasoners are available and can be used with the ontologies modeling human context.

In the automatic recognition of human context, the reasoner using the defined ontology can automatically validate for conflicts and inconsistencies in the definition of some human contexts. Moreover, the reasoner can also automatically classify an unknown type of context into some of the categories defined in the ontology. For all these reasons, and considering the recognized potential for ontologies to model and infer context [46], ontologies and ontological reasoning seem to be the appropriate means to comprehensively describe human context information and to automatically identify meaningful and rich expressions of human context which could enhance the real-world operation of behavior-aware systems.

There are also many other motivations for the use of ontologies and ontological reasoning in the description of heterogeneous resources in human behavior recognition systems, the dynamic selection of replacement sensors, the description of human context information, and the automatic inference of rich expressions of context for human behavior analysis. Ontologies surpass non-semantic models in terms of flexibility, extensibility, generality, expressiveness, and decoupling of the knowledge from the code. The hierarchical structure of ontologies, with subclasses inheriting the properties from their ascendant classes, facilitate its evolvability and maintenance. In fact, new concepts can be easily added to the ontology and related to the existing ones. Moreover, multiple methods exist for the automatic validation of conflicts and semantic inconsistencies of the newly added concepts. Using ontologies is also beneficial from the implementation perspective since no changes are required in the implementation of an architecture whenever the model is extended; thus, only requiring the adaptation of the ontology itself. Finally, one should not forget to mention the downside of using ontologies: the overhead in the knowledge representation and the complexity of defining the models. Nevertheless, all the presented advantages fairly overcome these two drawbacks of ontologies.

1.4. Thesis Goal and Objectives

Taking into consideration the limitations of human behavior recognition systems presented in Section 1.2 and the benefits provided by ontologies described in Section 1.3, there is a clear opportunity to create more advanced behavior-aware systems by using ontologies and ontological reasoning. Thus, the goal of this thesis is to investigate on the application of ontology engineering and reasoning in order to solve some of the most prominent limitations of human behavior recognition systems working in realistic conditions. In this way, this work seeks to contribute to paving the path to a new generation of behavior recognition systems readily available for their use in the real-world.

This thesis aims at achieving this goal via the following supporting objectives:

Objective 1: Design and development of an ontology for the comprehensive and interoperable description of sensing technologies used in behavior recognition systems.

The first objective of this thesis is designing and developing an ontology for the comprehensive and interoperable description of heterogeneous resources composing behavior recognition systems. Specifically, the objective is describing the magnetic and inertial measurement units (MIMUs) embedded into wearable platforms, which are the mainstream sensing technologies in behavior recognition systems. An ontology has to be proposed to exhaustively model the capabilities of the MIMUs, such as their measurement properties, and the characteristics of the wearable sensor platforms including their on-body location and their survival properties. This ontology will provide implicit semantics enabling the automatic interpretation of the resource descriptions and ensuring interoperability among diverse human behavior recognition systems. Thus, the ontological descriptions of the wearable sensor platforms consisting of MIMUs will abstract the heterogeneous resources from their underlying technology. Consequently, these descriptions will support the abstraction of the the sensor selection method from the actual sensing infrastructure. Therefore, the ontology will enable the dynamic sensor selection functionalities required to ensure the operational continuity of human behavior recognition systems.

Objective 2: Definition and validation of a method based on ontology reasoning and querying to dynamically select sensing technologies to support continuity of behavior recognition.

The second objective of this thesis is defining and validating a method based on ontology reasoning and querying to dynamically select the sensing technologies used in behavior recognition systems. Specifically, the objective is defining a method for selecting some of the available MIMUs, embedded into wearable platforms, whenever a MIMU part of the human behavior recognition system suffers some abnormality and demands a replacement. The selection method will build on the ontology which thoroughly describes the MIMUs, including their capabilities and the characteristics of the wearable sensor platforms in which they are embedded (as defined in Objective 1). Rules have to be established to define the characteristics of the candidate replacement MIMUs which could be used in the human behavior recognition system. Ontology query mechanisms will be used to search in the descriptions of the available MIMUs for the ones which match the required characteristics. In the processing of the ontological query, reasoning will be applied to infer the knowledge about the candidate replacements from the defined rules. Thus, the appropriate ontology queries have to be defined to allow the selection of the best MIMU which could replace the one suffering from anomalies. Finally, the novel ontology-based selection method has to be validated to prove that the dynamic selection and posterior replacement of an anomalous MIMU ensure the continuity of recognition, i.e., the reliability of the human behavior recognition system holds after the replacement takes place.

Objective 3: Design and development of an ontology for the exhaustive modeling of rich and meaningful expressions of context for human behavior analysis.

The third objective of this thesis is designing and developing an ontology for the exhaustive modeling of rich and meaningful expressions of context building on cross-domain information. Specifically, the objective is modeling the most commonplace contexts for health and wellness scenarios which involve sedentary and active lifestyles. The proposed ontology has to thoroughly model multiple primitive components of context, such as the physical activity, the location and the emotion, as well as more abstract daily contexts which can be derived from the combination of these primitives. This ontology will provide the implicit semantics required for the derivation of new richer context information from basic existing context. Thus, this ontology will enable the inference of meaningful human context information better describing human behavior and which can be useful to enhance behavior-aware systems.

Objective 4: Definition and validation of a method based on ontology reasoning to automatically infer rich and meaningful human context to enhance the operation of behavior-aware systems.

The fourth objective of this thesis is defining and validating a method based on ontology reasoning to automatically infer rich and meaningful human context to enhance the operation of behavior-aware systems. Therefore, a method applying ontological reasoning has to be defined to automatically identify richer human context out of more basic context primitives. The context inference method will build on the context model defined by the ontology which comprehensively describes rich and meaningful expressions of context based on primitive cross-domain contextual information (as defined in Objective 3). Ontology reasoning will be applied to the knowledge modeled in the context ontology to automatically validate for conflicts and inconsistencies in the definition of the contexts. Moreover, ontology reasoning will also be used to automatically classify an unknown type of context into some of the categories defined in the ontology. Finally, the ontological reasoning-based method for the inference of rich context has to be validated to prove that it can enhance the operation of behavior-aware systems.

1.5. Thesis Outline

This thesis is structured in five chapters.

Chapter 1 introduces human behavior recognition, analyzes the principal limitations of human behavior recognition systems under realistic conditions, motivates the use of ontologies and ontological reasoning to support real-world human behavior recognition, presents the thesis goal, and details the supportive objectives to achieve this goal.

Chapter 2 presents background in prior research about ontologies for modeling sensor networks, methods for semantic sensor selection, ontologies for human context modeling, and approaches for semantic context inference.

Chapter 3 proposes several ontologies to comprehensively model mainstream sensing technologies in behavior recognition systems, defines a method based on ontology reasoning and querying to dynamically select the sensing technologies to be used in behavior recognition systems, and evaluates the proposed ontology-based sensor selection method to prove that it ensures the operational continuity of human behavior recognition systems working in dynamic environments.

Chapter 4 proposes an ontology for the exhaustive modeling of rich and meaningful expressions of context, defines a method based on ontological reasoning to automatically identify abstract human context out of more basic context primitives, and evaluates this context inference method to prove that it can enhance the operation of behavior-aware systems in realistic conditions.

Finally, Chapter 5 presents the achievements and the contributions of this thesis, as well as possible future extensions of the presented research work and some final remarks.

2

State of the Art

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 [47] 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 [48], and for sensor data in Observations and Measurements (O&M) [49, 50]. These standards provide syntactic interoperability but lack semantic compatibility. Therefore, semantic web technologies are used to augment the OGC SWE standards in what is know as the Semantic Sensor Web [51].

OntoSensor [52] is an ontology which builds on the ideas of the OGC SensorML standard and extends the Suggested Upper Merged Ontology (SUMO) [53]. 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 [54] 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 [7] 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 [55, 56], the Semantic Gateway as Service (SGS) [57] and GeoSMA [58].

The SSN Ontology has been extended in the Wireless Semantic Sensor Network (WSSN) ontology [59]. 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 [60]. 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 [61]. 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 [62] 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 subontologies: 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 [63] 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 [64]. 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 [65]. 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. [66] 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 [67], 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.

2.3. Ontologies for Human Context Modeling and Inference

A number of surveys have reviewed the use and foundations of ontologies for context modeling. For example, a survey on context-aware systems [68] describes the basic design principles of context-aware architectures and depicts the different context models. Special focus is placed in this survey on the analysis and comparison of several approaches using ontologies. Another review of context modeling and reasoning techniques [69] discusses the requirements for modeling different context information and introduces the concept of high-level context abstractions. This survey describes and compares several ontology-based models of context information. Finally, a more recent survey on context-aware computing for the Internet of Things [70] evaluates 50 projects including the majority of research and commercial solutions proposed in the field of context-aware computing from 2001 to 2011. An extensive evaluation of research prototypes, systems and approaches building on ontology-based modeling and reasoning solutions is presented in this survey.

Many ontologies have been specifically proposed to model and recognize user context. The most well-known context ontologies and ontology-based context frameworks are described in the following. One of the most prominent ontologies for modeling context in pervasive environments is SOUPA (Standard Ontologies for Ubiquitous and Pervasive Applications) [71]. The core of the SOUPA ontology defines generic vocabularies for several domains: person, agent, belief-desire-intention, action, policy, time, space, and event. Similarly, CONON (CONtext ONtology) [72] is a noticeable ontology for smart home environments. The CONON upper ontology captures the general features of different context entities: person, activity, computational entity and location. Both SOUPA and CONON ontologies are generic and can be extended to describe the context in the application-specific domain. For example, the Context Broker Architecture (CoBrA) [73] adopts the SOUPA ontology, whereas the SOCAM (Service-oriented Context-Aware Middleware) [74] builds on the CONON ontology. The CoBrA ontology describes places, agents, events in an intelligent meeting room. The ontology proposed in SOCAM models persons, activities, locations and devices for smart home and vehicle environments.

Apart from these early well-known solutions, more recent context ontologies and ontology-based context frameworks have been proposed.

The Pervasive Information Visualization Ontology (PIVOn) [75] is composed of four ontologies for the description of intelligent environments: user, device, environment and service. The user model describes the static characteristics of the users, their agenda, and their situation including the user location, the current task and goals. The mIO! ontology [76] models context-related knowledge for the adaptation of applications in mobile environments. This ontology defines concepts like information on location and time, user information and its current or planned activities, as well as devices located in his surroundings. The Context Aggregation and REasoning (CARE) middleware [77] performs ontological and statistical reasoning to support the context-aware adaptation of internet services in a mobile computing environment. The ontology, which models the user context within the CARE middleware, describes the user activities (actions and movements), interests, contacts, calendar items and places. For example, the context *business meeting* is defined as including any activity performed in a conference room within a company building, and having at least two actors, each of which is an employee. Thus, the ontology in the CARE middelware models context based on activities and locations.

Some other works focus on the detection of a specific category of context, mainly activities, sometimes utilizing in their definition other type of contexts such as locations. ActivO is the ontology used in COSAR [78], an activity recognition system that supports hybrid statistical and ontological reasoning. The ActivO ontology models a set of activities and the context data required to recognize them (the person performing the activity, the location of the activity and the time extent in which the activity takes place). The authors of the ActivO ontology have also proposed a very similar approach but using OWL2 for modeling and reasoning [34]. Furthermore, some activities involve the interaction with objects. Thus, contextual information about the interaction (time and location) can be used to model and infer the activities. An ontology-based approach is used to model activities for smart homes in [79]. The proposed ontology models activities based on a sequence of user-object interactions and the location of the objects. For instance, the activity *making tea* is composed of the primitives *get cup*, *get tea*, *pour water*, *get milk* and *get sugar*, which take place in the *kitchen*. Composite activities in smart homes are modeled and recognized in [80]. Ontological and temporal knowledge modeling formalisms are combined to describe composite activities, like for example *make tea and then wash hands*. The work in [81] describes an ontology-based technique for multilevel activity recognition. The proposed ontology models atomic gestures (actions which cannot be decomposed), manipulative gestures (execution of simple atomic gestures), simple activities (temporal sequences of manipulative gestures) and complex activities (concurrent execution of simple activities). One example of complex activity could be *clean up* which is composed of the simple activities *put in dishwasher* and *clean table*. Finally, [82] proposes a fuzzy ontology for the representation of activity and the reasoning on vague, incomplete, and uncertain knowledge. The ontology core models three domains: users, environment including locations, and actions, activities and behaviors. Actions are atomic events, activities can be a single action or a composed set of actions, and behaviors are a sequence of activities and/or actions. For example, the behavior *coffee break* includes the action *exit office*, the activity *make coffee* or *take coffee*, and the action *enter office*.

3

Ontology-based Sensor Selection for Continuous Behavior Recognition

3.1. Overview

An enormous effort has been made during the recent years towards the recognition of human behavior 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 human behavior 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 behavior recognition systems must be fairly abstracted from the actual underlying network infrastructure. Section 3.2 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 presented in Section 3.3. 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 in 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 presented in Section 3.4 to evaluate the potential of MIMU-Wear for supporting the seamless operation of wearable human behavior recognition systems.

3.2. 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 [6] designed in a modular manner with an upper ontology and several plugable domain ontologies (see Figure 3.1). The MIMU-Wear Ontology builds on the standard W3C Semantic Sensor Network (SSN) Ontology [7], 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

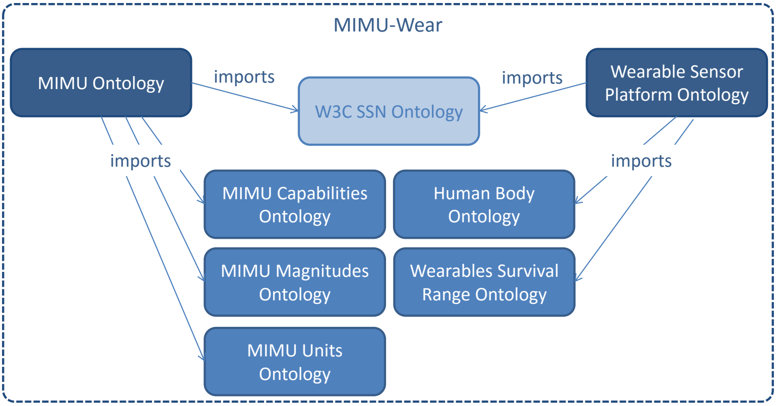


Figure 3.1: *Structure of the MIMU-Wear Ontology for the description of MIMU-based wearable platforms.*

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 2.1), 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 3.2.1) and the Wearable Sensor Platform Ontology (see Section 3.2.2). 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.

3.2.1. 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 Figure 3.2). 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

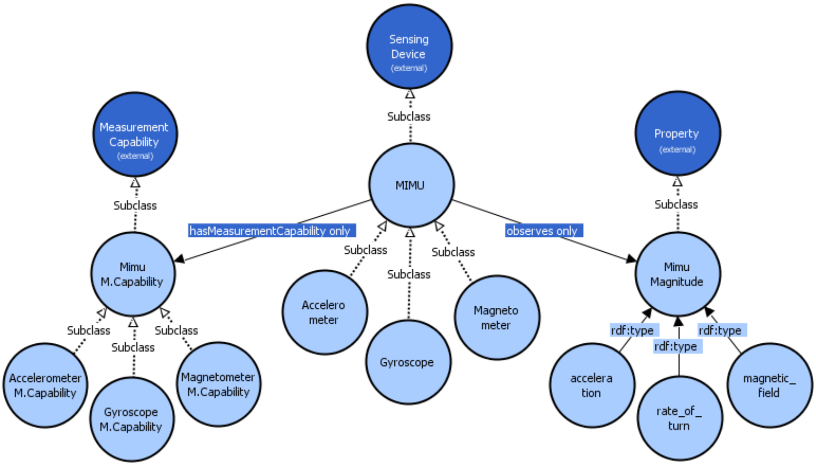


Figure 3.2: *MIMU Ontology: overview of the class* `MIMU` *and its relation to the class* `MimuMeasurementCapability` *and the class* `MimuMagnitude`*.*

`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 3.2.1). 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 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 [83], 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 article.

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:hasMeasurementCapability only MimuMeasurementCapability`. The class `MimuMeasurementCapability` is a subclass of the class `ssn:MeasurementCapability` defined in the MIMU Capabilities Ontology (see Section 3.2.1). From these assertions and the declared knowledge in the SSN Ontology, it can be inferred that all the members of the class `MimuMeasurementCapability` are related along 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 subclass of `ssn:hasMeasurement-`

Capability 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 along the property ssn:forProperty to the individual acceleration, the class GyroscopeMeasurementCapability is related along the property ssn:forProperty to the individual rate_of_turn, and the class MagnetometerMeasurementCapability is related to the individual magnetic_field.

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 Figure 3.3). 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 Figure 3.3). The class MimuMeasurementProperty is defined to be a subclass of the anonymous classes hasQuantityValue only xsd:float, hasRangeMaxValue only

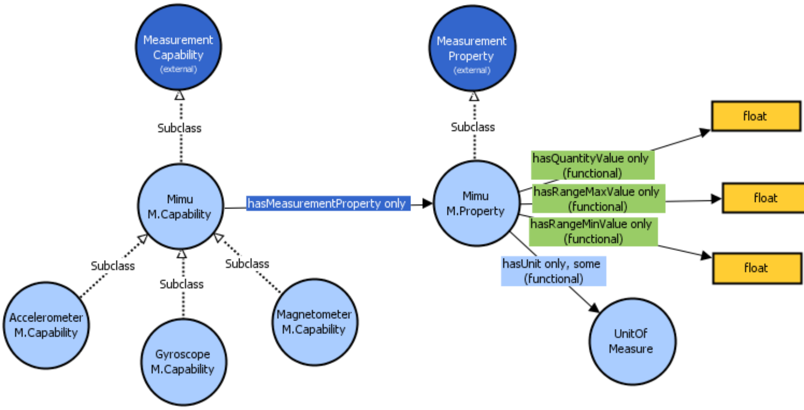


Figure 3.3: *MIMU Capabilities Ontology: overview of the class* `MimuMeasurementCapability` *and the class* `MimuMeasurementProperty`*, and the relation between them.*

`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) [84]. 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 3.2.1), 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

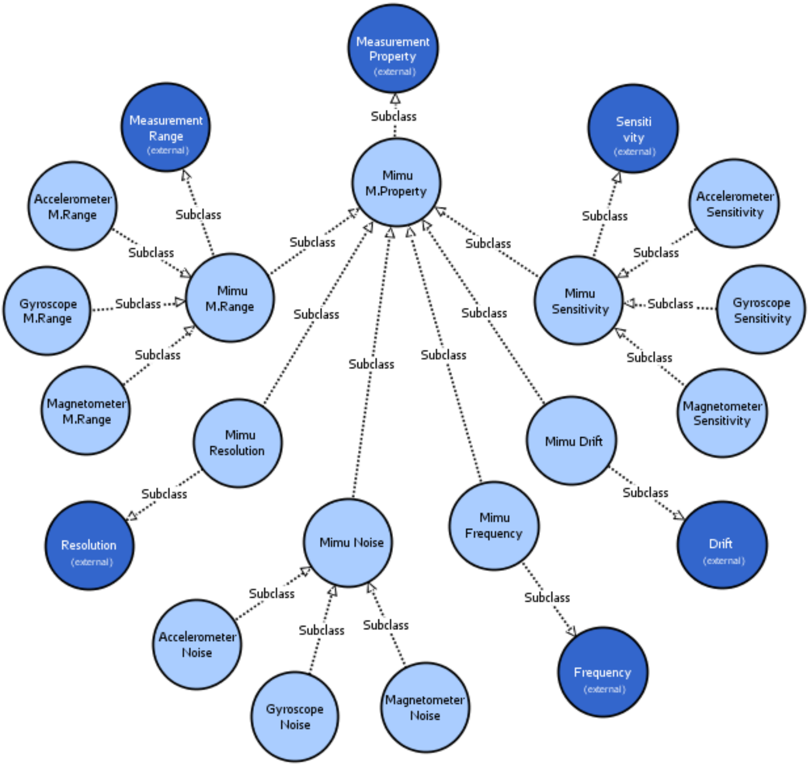


Figure 3.4: *MIMU Capabilities Ontology: overview of the subclasses of the class `MimuMeasurementProperty`.*

classes `MimuMeasurementRange`, `MimuSensitivity`, `MimuResolution`, `MimuFrequency`, `MimuDrift` and `MimuNoise` are here defined (see Figure 3.4). 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 Figure 3.5). 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`.

The class `MimuSensitivity` is the subclass of the class `ssn:Sensitivity` and the class `MimuMeasurementProperty` 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 `MagnetometerSensitivity`. The class `AccelerometerSensitivity` is defined to be a subclass of the anonymous class `hasUnit value m_per_square_s` (see Figure 3.5), the class `GyroscopeSensitivity` 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

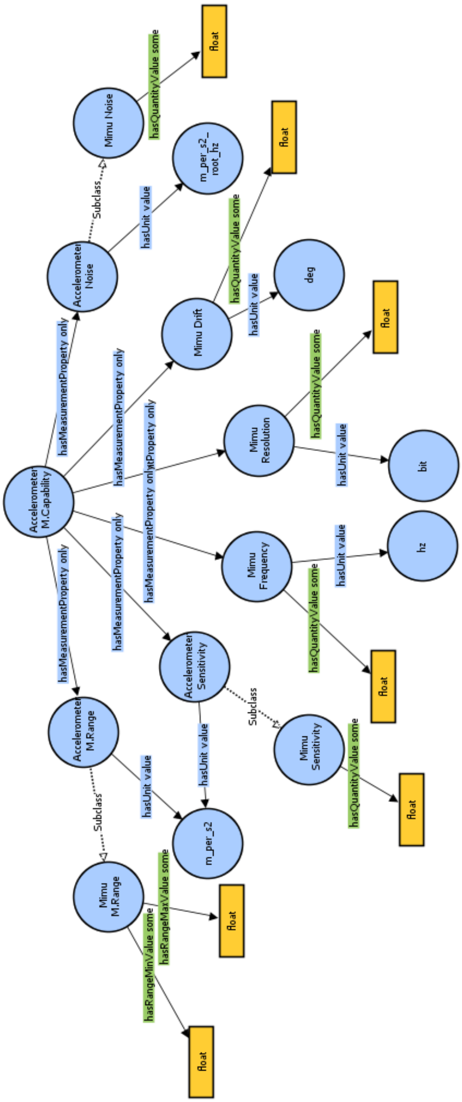


Figure 3.5: *MIMU Capabilities Ontology: Overview of the class* `AccelerometerMeasurementCapability` *and the subclasses of the class* `MimuMeasurementProperty` *which define it.*

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 Figure 3.5).

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 Figure 3.5).

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 Figure 3.5).

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 ($\mathrm{m/s^2}/\sqrt{\mathrm{Hz}}$ for the accelerometer, $\mathrm{deg/s}/\sqrt{\mathrm{Hz}}$ for the gyroscope, and $\mathrm{gauss}/\sqrt{\mathrm{Hz}}$ 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 Figure 3.5). 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 Figure 3.5). 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)`.

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 ontolgies 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.

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` (representing m/s^2), `gauss` (gauss), `deg_per_s` (deg/s), `hz` (hertz), `bit` (bit), `deg` (degree), `m_per_square_s_and_root_hz` ($m/s^2/\sqrt{Hz}$), `gauss_per_root_hz` ($gauss/\sqrt{Hz}$), `deg_per_s_and_root_hz` ($deg/s/\sqrt{Hz}$). 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.

3.2.2. 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 Figure 3.6). This class is a subclass of the class `ssn:Platform` and particularizes the concept of platform

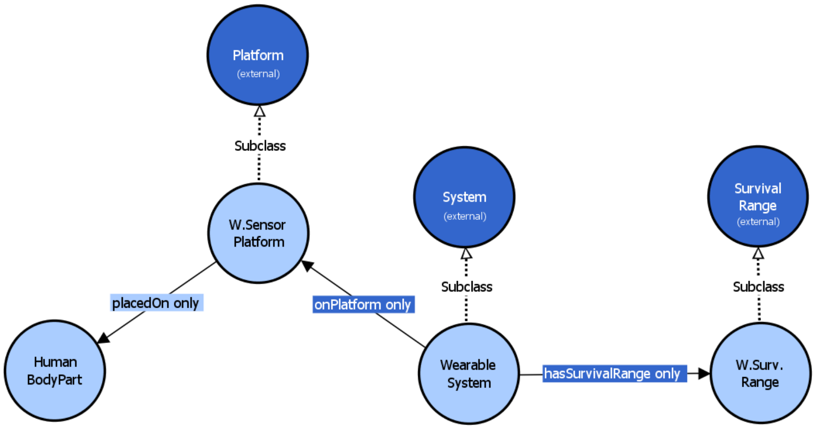


Figure 3.6: *Wearable Sensor Platform Ontology: overview of the class* `WearableSensorPlatform` *and its relation to the class* `WearableSystem` *and the class* `HumanBodyPart`*.*

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`.

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 3.2.2).

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 Wearable-SurvivalRange`. 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.

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 (see Figure 3.7). 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

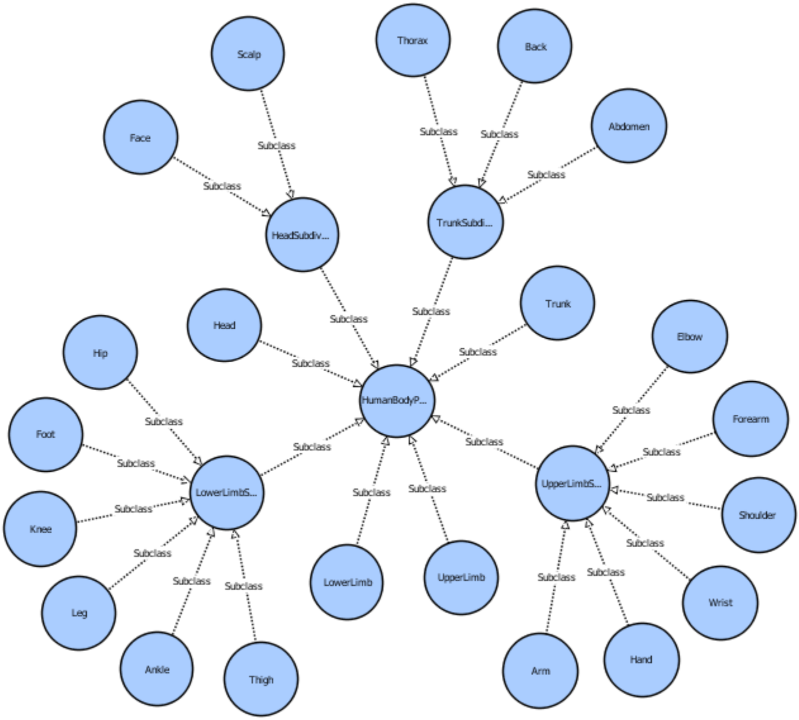


Figure 3.7: *Human Body Ontology: overview of the hierarchy of subclasses for the class* `HumanBodyPart`*, including its direct subclasses* `Head`*,* `Trunk`*,* `UpperLimb`*,* `LowerLimb`*,* `HeadSubdivision`*,* `TrunkSubdivision`*,* `UpperLimbSubdivision` *and* `LowerLimbSubdivision`*, and their subclasses.*

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 Figure 3.8). The relations between the rest of body parts and their subdivisions have been established using the same modeling principle (see Figure 3.9 for the definition of the class `Trunk`, Figure 3.10 for `UpperLimb` and Figure 3.11 for `LowerLimb`).

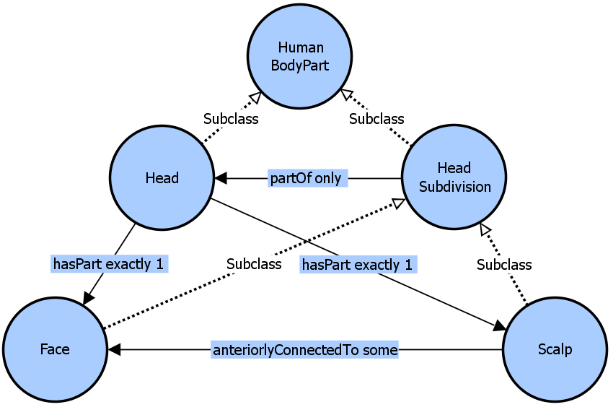


Figure 3.8: *Human Body Ontology: overview of the class* `Head` *and the class* `HeadSubdivision`*.*

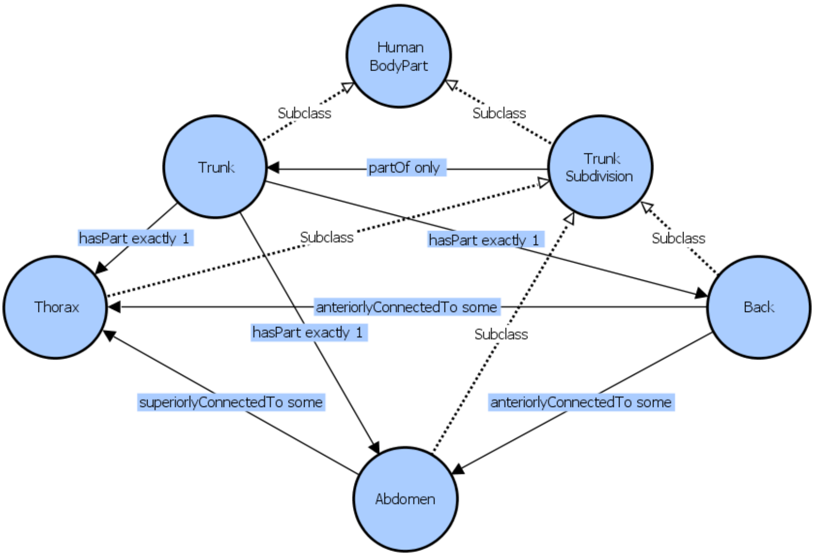


Figure 3.9: *Human Body Ontology: overview of the class* `Trunk` *and the class* `TrunkSubdivision`*.*

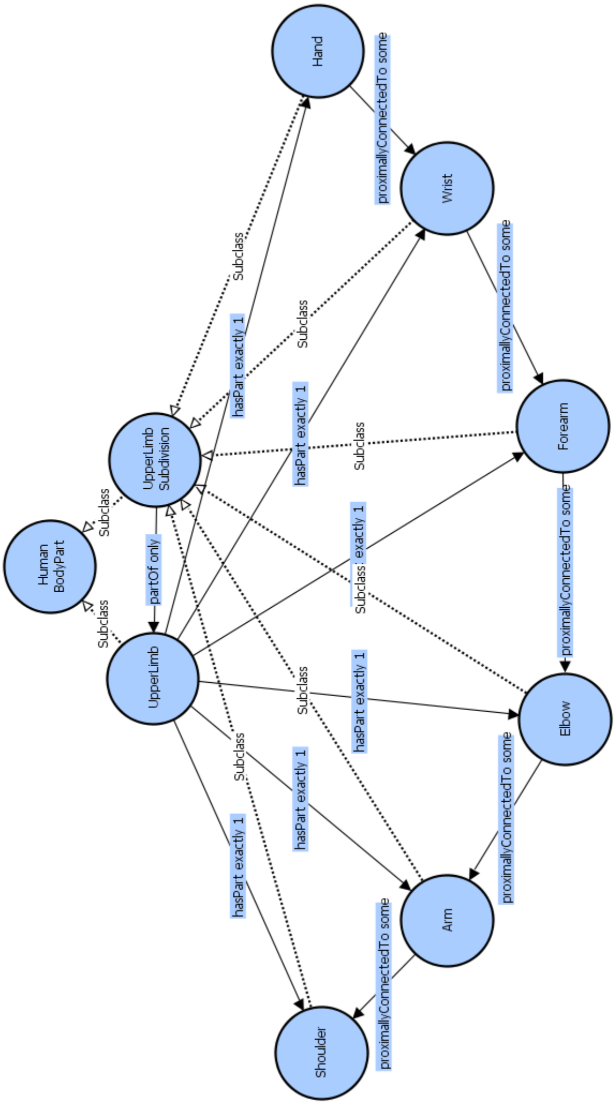


Figure 3.10: *Human Body Ontology: overview of the class `UpperLimb` and the class `UpperLimbSubdivision`.*

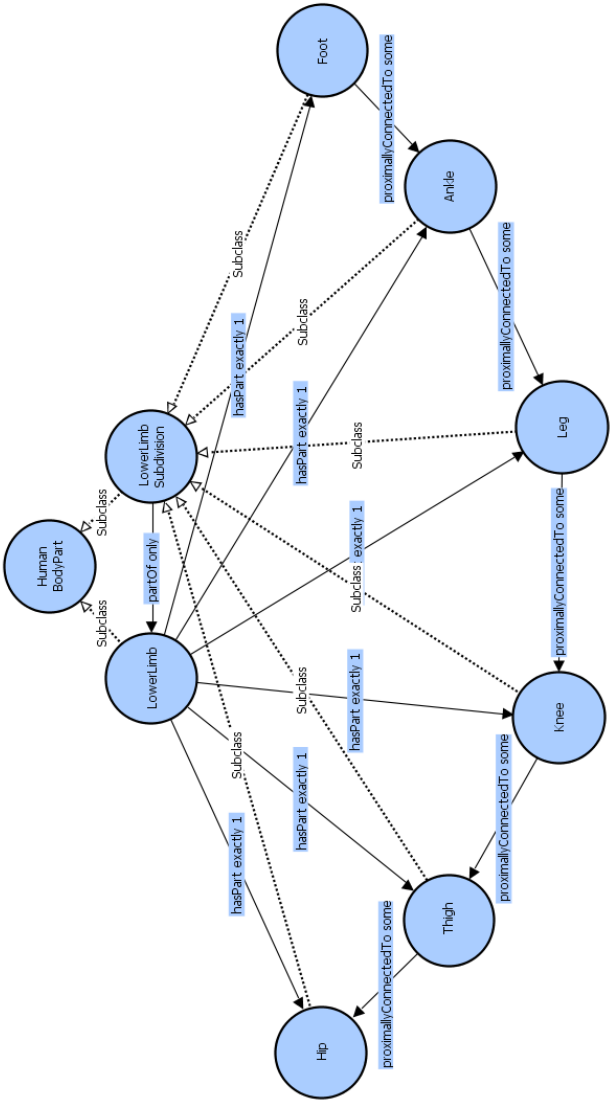


Figure 3.11: *Human Body Ontology: overview of the class* `LowerLimb` *and the class* `LowerLimbSubdivision`*.*

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 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 Figure 3.9), 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`. 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 Figure 3.8). In the case of the upper and lower limbs, the same modeling principle applies but the property `proximallyConnectedTo` and its inverse `distallyConnectedTo` are used to establish the connections (see Figure 3.10 and Figure 3.11). 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` along 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) [85], one of the most complete knowledge source for bioinformatics which represents the phenotypic structure of the human body, and the Uber anatomy ontology (Uberon) [86], 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.

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 Figure 3.12). 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 `Wearable-SurvivalProperty` 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) [84]. 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

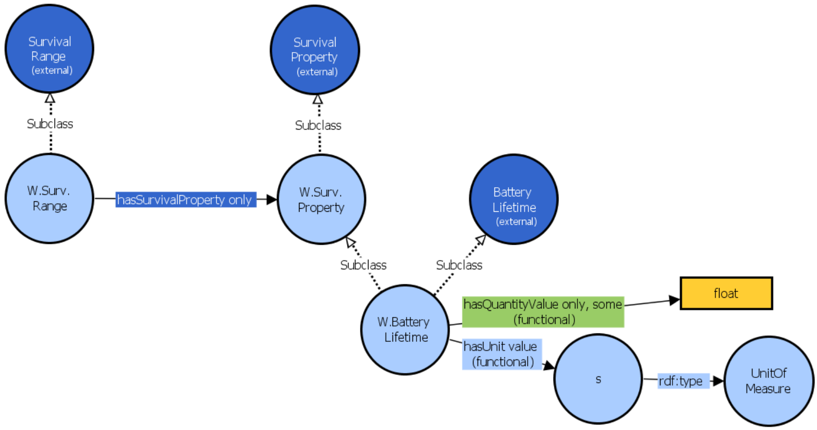


Figure 3.12: *Wearables Survival Range Ontology: overview of the class `WearableSurvivalRange` and the class `WearableSurvivalProperty`.*

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.

3.2.3. 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 3.4.2.

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 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 [87]. 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 sensor platform should be updated every time there is a change to reflect the real time situation.

3.3. 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 behavior recognition systems whenever a sensor is detected to have suffered some anomaly. The goal of the sensor selection method is to ensure continuity of behavior recognition. Due to the failure or the fault of one of the MIMU sensors, the performance of the wearable behavior recognition system might decrease. Using a replacement sensor which can provide the human behavior recognition system with a similar sensing functionality, the performance of this behavior recognition system 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 behavior recognition system (see Subsection 3.3.1). 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 behavior recognition system (see Subsection 3.3.2).

3.3.1. 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 behavior recognition system. In the proposed sensor selection method, SWRL rules are utilized to define the candidate replacement sensors. SWRL [8] 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 human behavior recognition system. Therefore, depending on the characteristics of the behavior 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 3.1 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 behavior recognition system 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*.

In case there is no other MIMU able to measure the same type of magnitude, transfer learning could be applied [88]. In this case, the requirement would be finding a replacement sensor of another modality capable of measuring a different type of magnitude. Rule#2 states that the magnitude observed by two different MIMUs has to be different in order for *?s2* to be a candidate replacement for *?s1*.

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 as specified in Rule#3.

Table 3.1: *Rules for identification of candidate sensor replacements based on the MIMU types.*

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

In the rules presented in Table 3.1, 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 3.2 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 3.1 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. Therefore, if the MIMU *?s2* has a sensitivity (*?p2*) that takes a value *?v2* which is greater than or equal to *?v1* which is the sensitivity (*?p1*) of the sensor *?s1*, then *?s2* is a candidate replacement for *?s1*.

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 (*swrlb:greaterThanOrEqual(?v2, ?v1)*, where *?v1* and *?v2* are the values of the resolution of the MIMUs *?s1* and *?s2*).

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 (*swrlb:lessThanOrEqual(?v2, ?v1)*, where *?v1* and *?v2* are the values of the drift or the noise of the sensors *?s1* and *?s2*).

Finally, Rule#9 identifies candidate replacements which belong to a different MIMU type but execute the measurements at the same rate or frequency. Therefore, if the sensor *?s2* has a frequency (*?p2*) that takes a value *?v2* equal to *?v1* which is the frequency (*?p1*) of the sensor *?s1*, i.e., *swrlb:equal(?v2, ?v1)*, then *?s2* is a candidate replacement for *?s1*.

Rule#4, Rule#5, Rule#6, Rule#7, Rule#8 and Rule#9 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.

Table 3.2: *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)

Continued on next page

Table 3.2 continued from previous page

ID	Description	Rule
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)

Continued on next page

Table 3.2 continued from previous page

ID	Description	Rule
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)

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 [4]. 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 behavior recognition system. Therefore, some rules incorporating the location of the wearable sensor platform hosting the MIMU need to be defined. Table 3.3

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 specifies that two MIMUs (*?s1* and *?s2*) are part of the same wearable sensor platform, represented as *?w1* and *?w2*, and therefore *?s2* is a candidate replacement for *?s1*.

Another option would be identifying as a candidate replacement a MIMU sensor hosted on a wearable sensor platform located on the same body part where the anomalous sensor is. If the two sensors are worn on the same body part, one could expect that the measurements they provide would be very similar. Rule#11 states that if two MIMUs (*?s1* and *?s2*) are part of two wearable sensor platforms which are located on the same body part (*?l1* and *?l2*), *?s2* is a candidate replacement for *?s1*.

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. 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 part. According to Rule#13, 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. 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. Rule#14 presents this idea: if two MIMUs (*?s1* and *?s2*) are part of two wearable sensor platforms which are located on two body parts (*?l1* and *?l2*) which are part of the same body division, represented in the rule as *partOf(?l1,?d1)*, *partOf(?l2,?d2)* and *sameAs(?d1, ?d2)*, then *?s2* is a candidate replacement for *?s1*.

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. In 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. Rule#15 defines this situation: if two MIMUs (*?s1* and *?s2*) are part of two wearable sensor platforms which are located on symmetric body parts, represented in the rule as *symmetricTo(?l1, ?l2)*, then *?s2* is a candidate replacement for *?s1*.

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. According to Rule#16, any sensor on the left upper limb would be a candidate to replace an anomalous sensor on the right forearm.

Table 3.3: *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)

Continued on next page

Table 3.3 continued from previous page

ID	Description	Rule
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)

Continued on next page

Table 3.3 continued from previous page

ID	Description	Rule
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)

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 3.4 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 seconds (*swrlb:greaterThan(?v2, "3600"^^float)*).

Table 3.4: *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)

All the presented rules can be combined in order to obtain more meaningful descriptions for the identification of candidate sensor replacements. Table 3.5 shows some examples of 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 seconds.

Rule#20 combines Rule#4 with Rule#15 and identifies that a candidate replacement is a MIMU able of measuring the same type of magnitude with an equal or greater measurement range, and part of a wearable sensor platform located on the symmetric body part.

Table 3.5: *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)

Continued on next page

Table 3.5 continued from previous page

ID	Description	Rule
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)

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.

3.3.2. 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 human behavior recognition system.

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 [9], a query language for RDF, is utilized in the sensor selection method because of its fully potential to query OWL 2 data. Listing 3.1 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.

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

Listing 3.1: *SPARQL query to retrieve all the candidate sensors to replace a MIMU with identifier* `<sensor-id>`.

The query presented in Listing 3.1 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 behavior recognition system. 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 3.2 should be executed.

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

Listing 3.2: *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 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. 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`.

3.4. Evaluation of MIMU-Wear and the Sensor Selection Method

The sensor selection method, based on MIMU-Wear and designed to ensure the continuity of behavior recognition, is evaluated for an exemplary application scenario in the domain of 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 recognition system 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 [89, 90] in order to identify an appropriate candidate replacement sensor. 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.

3.4.1. Deployment Scenario

Nine MIMU-based wearable platforms are considered in this exemplary scenario (see Figure 3.13). 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 [91]. 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.

3.4.2. 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 usage of MIMU-Wear is presented in Section 3.2.3, but here the process is described through an example. First, the generic descriptions of the three

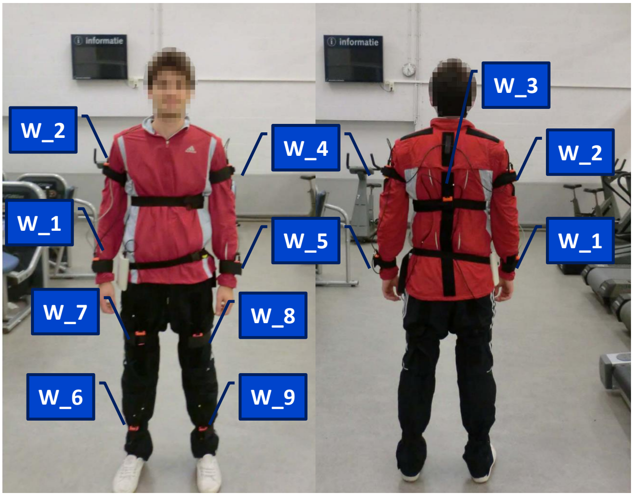


Figure 3.13: *Sensor deployment for the experimental scenario. Nine MIMU-based wearable platforms (W_ 1,...,W_ 9) are placed on different parts of the user body.*

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.

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` (see Figure 3.14). The class `MTxAcc-MeasurementCapability` is defined here to represent the actual mea-

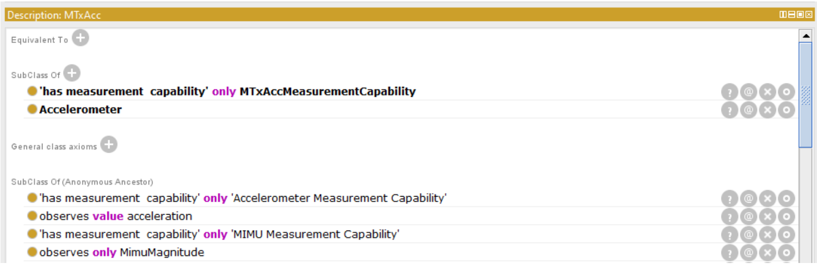


Figure 3.14: *`MTxAcc`: class modeling the accelerometer embedded into the MTx.*

surement 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, and which can be obtained from the MTx specification sheet [91]. 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` (see Figure 3.15).

The measurement range for the standard configuration of the MTx accelerometer is ± 50 m/s^2. In order to represent this range, the individual `standard_MTxAcc_mrange` of the class `Accelerometer-MeasurementRange` has asserted for the property `hasMaxValue` the value `“50”^^float` and for the property `hasMinValue` the value `“-50”^^float` (see Figure 3.16).

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 deg 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`.

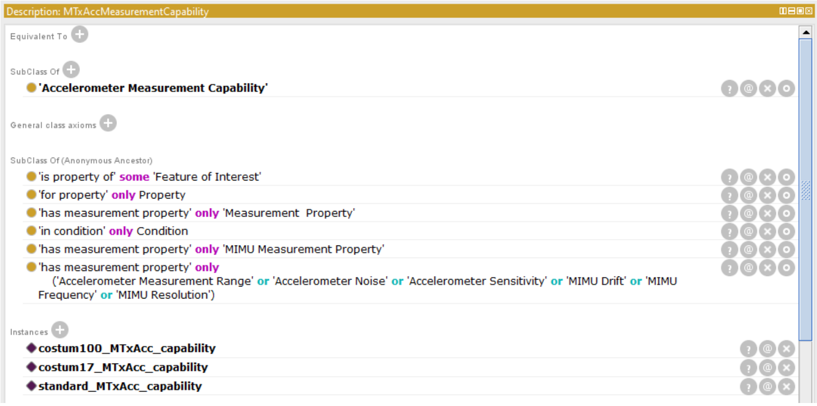


Figure 3.15: *`MTxAccMeasurementCapability`: class modeling the measurement capabilities of the MTx accelerometer.*

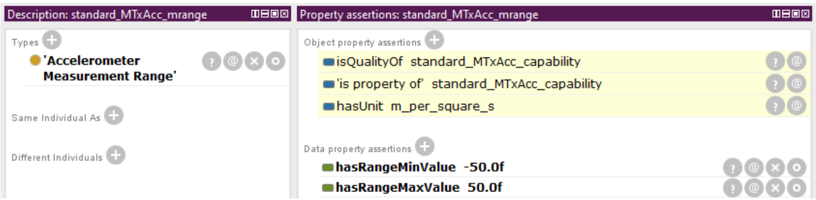


Figure 3.16: *`standard_MTxAcc_mrange`: individual of the class `AccelerometerMeasurementRange` which models the ± 50 m/s^2 measurement range for the standard configuration of the MTx accelerometer. The axioms marked in yellow represent the inferred knowledge, whereas the others are asserted axioms.*

Finally, the noise of the MTx accelerometer is 0.002 m/s$^2/\sqrt{\text{Hz}}$ 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` (see Figure 3.17).

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`,

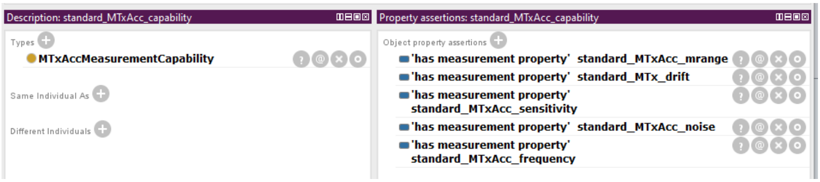


Figure 3.17: *`standard_MTxAcc_capability`: individual of the class `AccelerometerMeasurementCapability` representing the capabilities of the standard configuration of the MTx accelerometer.*

`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.

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 creation of the MTx descriptions is explained.

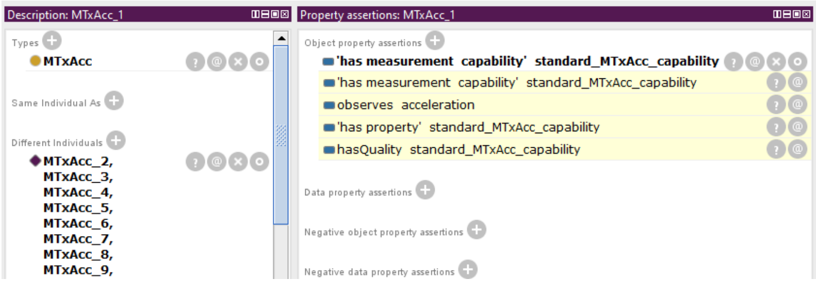


Figure 3.18: *`MTxAcc_1`: instance of the the class `MTxAcc` which describes the accelerometer embedded into the MTx. The axioms marked in yellow represent the inferred knowledge, whereas the others are asserted axioms.*

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 Figure 3.18) 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 Figure 3.19) represents the wearable sensor platform worn on the right forearm (W_1 in the scenario). The corresponding individual `MTxSys_1` (see Figure 3.20) 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

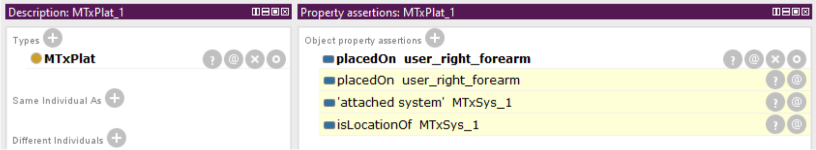


Figure 3.19: *`MTxPlat_1`: instance of the the class `MTxPlat` which describes the MTx. The axioms marked in yellow represent the inferred knowledge, whereas the others are asserted axioms.*

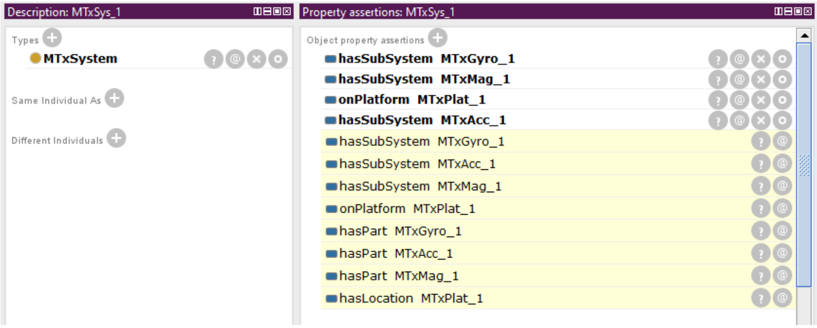


Figure 3.20: *`MTxSys_1`: instance of the the class `MTxSystem` which describes the MTx. The axioms marked in yellow represent the inferred knowledge, whereas the others are asserted axioms.*

the values of the property `ssn:hasSubsystem`. For example, in the case of the individual `MTxSys_1`, the property `ssn:hasSubsystem` 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`) should also be created. Table 3.6 summarizes the names of the individuals modeling the scenario.

Table 3.6: *Summary of the ontological description of the MIMU-based wearable sensor platforms for the scenario presented in Figure 3.13.*

ID (Fig 3.13)	**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`

3.4.3. 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 [4, 3]. On the other hand, the more sensors are utilized, the more computationally and energy expensive the activity recognizer turns to be [37], 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 Figure 3.13) are configured to capture only acceleration since this magnitude proves to work well-enough for the recognition of a variety of activities [92]. 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 [93, 94, 95, 96].

The generic rules presented in Section 3.3.1 have to be particularized for the actual application scenario and the queries resented in Section 3.3.2 prioritized accordingly. This is required 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. Four rules are envisioned for this application scenario (see Table 3.7). 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 3.1 and Rule#11 in Table 3.3.

A second option consists 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 3.1 and Rule#15 in Table 3.3.

The third rule is a combination of Rule#1 in Table 3.1 and Rule#12 in Table 3.3, 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 3.1 and Rule#13 in Table 3.3.

Table 3.7: *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)

Continued on next page

Table 3.7 continued from previous page

Priority	Rule
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)

In order to request the results provided by the rule with priority P1, the associated SPARQL query which is executed is shown in Listing 3.3. In the same way, to retrieve the results of rule with priority P2 the associated query is shown in Listing 3.4, for the rule with priority P3 the query is shown in Listing 3.5, and for the rule with priority P4 the query is shown in Listing 3.6. 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`.

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

Listing 3.3: *SPARQL query for retrieving the results of rule with priority P1*

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

Listing 3.4: *SPARQL query for retrieving the results of rule with priority P2*

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

Listing 3.5: *SPARQL query for retrieving the results of rule with priority P3*

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

Listing 3.6: *SPARQL query for retrieving the results of rule with priority P4*

Summarizing, the iterative query method for sensor selection works as follows. First the query in Listing 3.3 is executed. If it provides a result, the search is stopped since a candidate replacement has been identified. Otherwise, the query in Listing 3.4 is executed and so on. If no results are obtained while executing the last query, i.e., Listing 3.6, 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 3.3 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 3.4, where the string `<sensor-id>` is replaced with `MTxAcc_1`. This query returns the individual `MTxAcc_5`, which is the accelerometer in W5, 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` 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 along the property `symmetricTo`. Therefore, the rule with priority P2 in Table 3.7 is satisfied and the following axiom is inferred `MTxAcc_1 hasReplacementSameTypeConnAdjPart MTxAcc_5` (see Figure 3.21). 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 3.3) does not produce any result. Then, the second query (Listing 3.4) 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

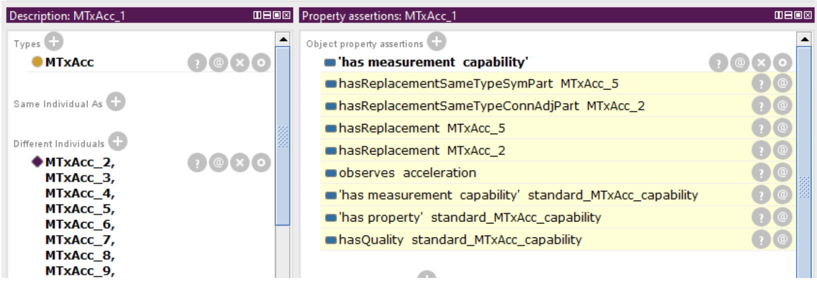


Figure 3.21: *Instance `MTxAcc_1` of the class `MTxAcc` which shows the candidate replacement sensors via the inferred property `hasReplacement` and its subproperties.*

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 3.3. 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 3.4. 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 3.5 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 3.6 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 W4, 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 platform located on the user back.

3.4.4. Reliability of the Sensor Selection Method

The different replacement scenarios described before are here evaluated by using the REALDISP dataset [97]. 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 Figure 3.13.

An analysis of the classification reliability 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 the platforms W_1 (hereafter, ACC_1), W_3 (hereafter, ACC_3) and W_9 (hereafter, ACC_9). 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-seconds duration [98]. 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 [99]. These features are used as input to the classifier, which is here defined through a decision tree model [100] 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 [36, 88, 4, 96].

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 trustful average estimate of the reliability of the recognizer in each case. Moreover, the $Fscore$ or F_1-*score* [101], a combination of precision and recall measures, is used as reliability 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 Figure 3.22, Figure 3.23 and Figure 3.24. The baseline is driven by the ideal case, for which a $Fscore$ average value of 0.85 is attained. Now, in the first case (Figure 3.22), once ACC_1 behaves abnormally a $Fscore$ drop of around 0.25 is observed, thus leading to an overall $Fscore$ 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. This sensor is actually the one that would be found for replacement through MIMU-Wear as shown in Section 3.4.3. In the second case (Figure 3.23) the benefit of using an accurate sensor replacement strategy is even clearer. Once ACC_9 gets affected the $Fscore$ 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 $Fscore$ 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 (Figure 3.24) 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.

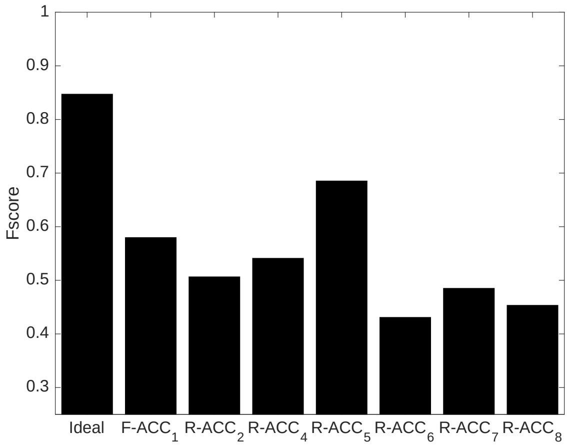


Figure 3.22: *Fscore for the different sensor replacement scenarios when there is a failure of the accelerometer in W_1 (ACC_1). Legend: "Ideal" = configuration ACC_1, ACC_3, ACC_9; "F-ACC_1" = same as the ideal configuration but with the ACC_1 not working properly; "R-ACC_k" = same as ideal configuration but with ACC_1 replaced with ACC_k.*

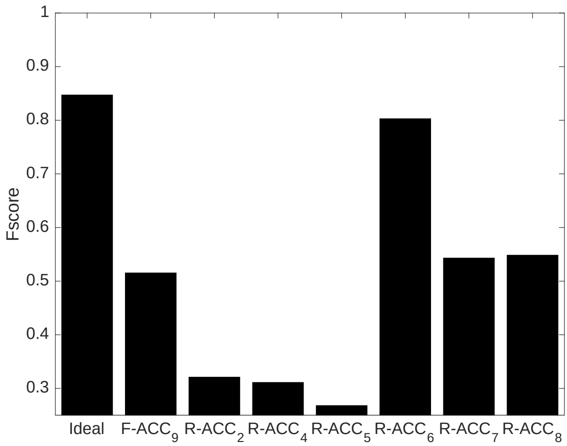


Figure 3.23: *Fscore for the different sensor replacement scenarios when there is a failure of the accelerometer in W_9 (ACC_9). Legend: "Ideal" = configuration ACC_1, ACC_3, ACC_9; "F-ACC_9" = same as the ideal configuration but with the ACC_9 not working properly; "R-ACC_k" = same as ideal configuration but with ACC_9 replaced with ACC_k.*

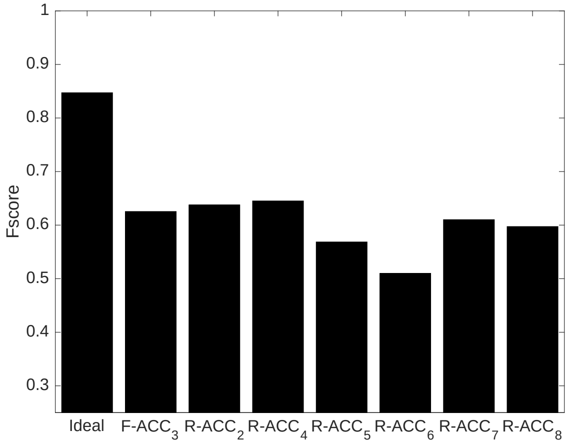


Figure 3.24: *Fscore for the different sensor replacement scenarios when there is a failure of the accelerometer in W_3 (ACC_3). Legend: "Ideal" = configuration ACC_1, ACC_3, ACC_9; "F-ACC_3" = same as the ideal configuration but with the ACC_3 not working properly; "R-ACC_k" = same as ideal configuration but with ACC_3 replaced with ACC_k.*

4

Ontology-based Context Inference for Human Behavior Analysis

4.1. Overview

Recent years have witnessed a huge progress in the automatic identification of individual primitives of human behavior such as activities or locations. However, the complex nature of human behavior demands more abstract contextual information for its analysis. This work presents an ontology-based method that combines low-level primitives of behavior, namely activity, locations and emotions, unprecedented to date, to intelligently derive more meaningful high-level context information. Section 4.2 introduces a new open ontology describing both low-level and high-level context information as well as their relationships. Furthermore, the context inference method building on the developed ontology and on ontological reasoning is presented in Section 4.3 and evaluated in Section 4.4. The proposed inference method proves to be robust while identifying high-level contexts even in the event of erroneously-detected low-level contexts.

4.2. An Ontology for the Description of Human Context

The Mining Minds Context Ontology models context for human behavior identification in order to enable the provision of personalized health and wellness services [102, 10]. Since Dey proposed the first widely-accepted definition of context [103], many different interpretations of context have arisen. Human context is here defined as any information characterizing the physical, mental and social situation of a person that enables the identification of their behavior. Furthermore, human context is here categorized into two different levels of abstraction: low-level context and high-level context. Low-level context is defined as primitive context, i.e., contexts that can be directly identified from user data and do not require any other type of context information to be derived. Specifically, activities, locations and emotions are here considered as the three categories of low-level context. Activities can be normally identified from the body movement; locations can be directly derived from the user position; and emotions can be obtained from the user sentiments or physiological responses. High-level context is the context which requires several contexts of diverse nature in order to be identified. This means that a high-level context builds on a combination of low-level contexts. Therefore, high-level contexts are more complex and abstract contexts.

The Mining Minds Context Ontology aims at comprehensively modeling the most commonplace and widely-used contexts for health and wellness services. These contexts are typically observed for both sedentary and active lifestyles. Specifically, the high-level contexts include daily contexts like *office work*, *sleeping*, *house work*, *commuting*, *amusement*, *gardening*, *exercising*, *having meal*, and *inactivity*. The low-level contexts required to compose the description of the high-level context have to be automatically recognizable. Thus, very simple low-level contexts in the domains of activities, locations and emotions are defined. Low-level contexts describing activities include sedentary activities associated to unhealthy habits, mild activities of the daily living and some vigorous ones related to sport and fitness practices. Namely, the modeled activities are *lying down*, *standing*, *sitting*, *riding escalator*, *riding elevator*, *walking*, *running*, *jumping*, *hiking*, *climbing stairs*, *descending stairs*, *cycling*, *stretching*, *dancing*, *sweeping*, and *eating*. Similarly, the low-level contexts describing the locations comprise the places where the user spends their daily life, i.e., *home*, *office*, *yard*, *gym*, *mall*, *restaurant*, *outdoors*, and *transport*. The low-level contexts describing the emotions embrace the most prominent moods or states of mind, which are *anger*, *happiness*, *neutral*, *sadness*, *fear*, *disgust*, *surprise*, and *boredom*. The specific combinations of low-level contexts that compose each high-level context are derived from the experience of behavioral scientists. Figure 4.1 graphically represents these definitions of high-level context, which are modeled in the Mining Minds Context Ontology. The considered contexts are intended to represent a wide spectrum of situations and actions in a person's life; however, it must be noted that this list can certainly be extended in view of potential future applications while considering other less recurrent contexts.

In broad strokes, the main novelties of the Mining Minds Context Ontology are a more comprehensive description of context using a two-level model and the incorporation of emotion information to detect some high-level contexts. First, a layered approach is followed in which high-level contexts build on a combination of low-level contexts. Current approaches model context in different dimensions, for example the user is performing an activity, has a location, and has a mood. However, in these models there is no clear link between the different dimensions of context, neither are they used to derive other contexts. Thus, some valuable information for the identification of human behavior is lost when using a one-level model. Second, the emotions enable the definition of new high-level contexts which can only be identified

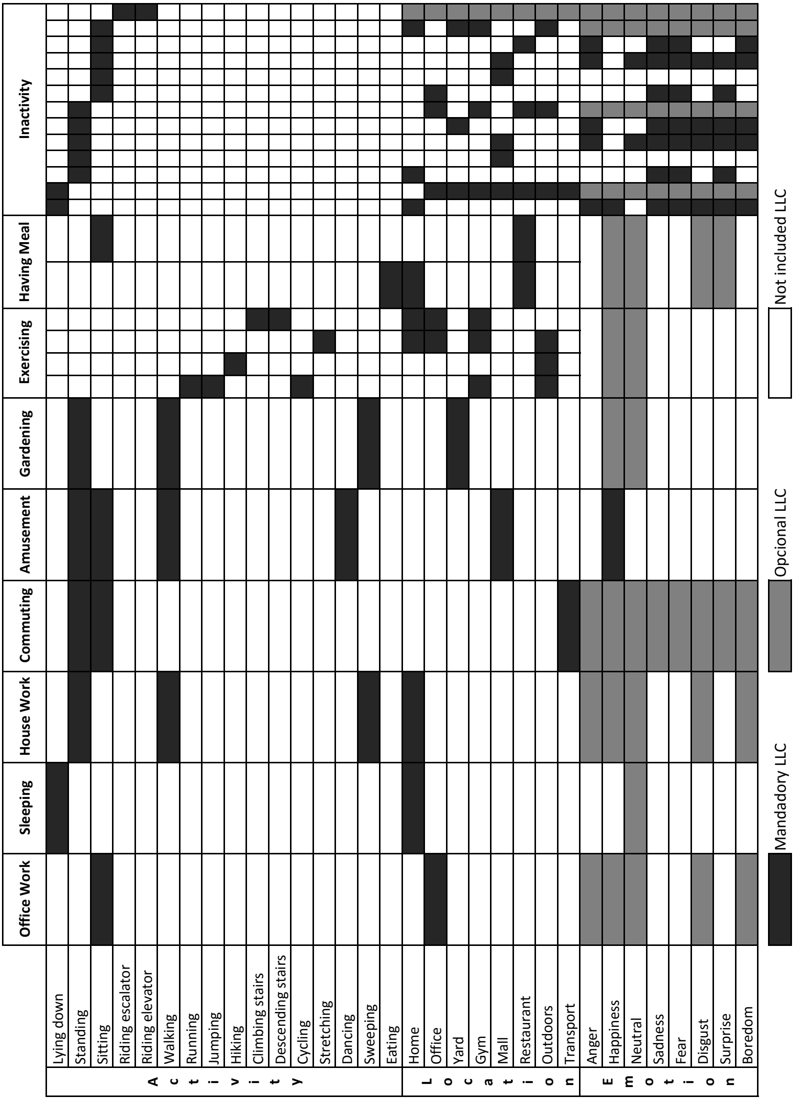


Figure 4.1: *Graphical representation of the combination of low-level contexts which compose the high-level contexts modeled in the Mining Minds Context Ontology.*

whenever a specific emotion takes place. This is the case of the high-level context *amusement* which must imply that the person is happy and having fun. For this context, it is not enough to know that the person is *sitting* in the *mall*, but also that their emotion is *happiness* in order to infer that the context refers to *amusement*. Therefore, in some cases the activity and the location might not be enough to detect the high-level context, and the emotion enables the identification of more diverse high-level contexts. The Mining Minds Context Ontology is an OWL 2 ontology [6] and is publicly available at `http://www.miningminds.re.kr/icl/context/context-v2.owl`.

4.2.1. Terminology for the Definition of Context

The Mining Minds Context Ontology defines the concept of user context. The context is associated to a given user and has a start and an end. While a context has necessarily a start referring to the time in which the context initiates, the finalization of the context is not strictly necessary. This is motivated by the fact that the context may prolong over time and be still valid at the present time. A given context can refer to either low or high-level context. Low-level contexts represent either activities, locations or emotions, which can further compose a high-level context. In some cases only one category of the low-level context is enough to determine the high-level context. This is the case of *inactivity*, where a sole sedentary activity like *sitting* defines this context. In some other cases, a specific category of low-level context is essential in order to identify the high-level context. For example, *amusement* can only be detected if the emotion is of type *happiness*. Accordingly, the ontology has been designed to support any combination of low-level contexts to define a specific high-level context. Given the seldom availability of emotion data, the ontology has been designed to procure the identification of some high-level contexts, even in the absence of emotion information. The modeling of this concept of context using the formal ontological description is presented in the following.

The main concept of the Mining Minds Context Ontology is the class `Context`, which represents the context of a user in an interval of time. Several necessary conditions are described for this class to model the concept of context (Figure 4.2). The existential and universal restrictions on the object property `isContextOf` ensure that any individual of the class `Context` is linked to an individual of the class `User` representing the user to which the context belongs. The existential and universal

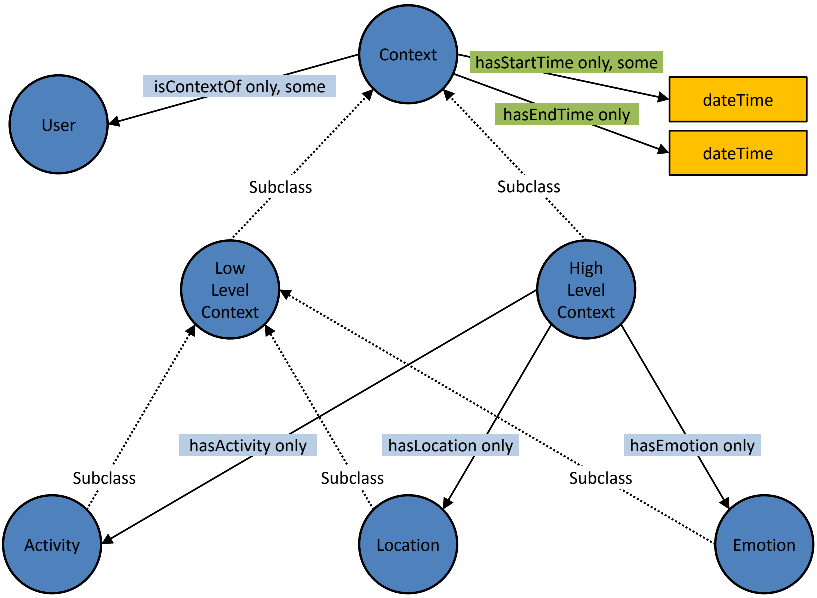


Figure 4.2: *Mining Minds Context Ontology: the class* `Context`*, its subclasses and the relations among them.*

restrictions on the functional data property `hasStartTime` state that all the individuals of the class `Context` must be related along this property to a unique `dateTime` datatype of the W3C XML Schema Definition Language (XSD) [84] representing the instant of time in which the context starts. The universal restriction on the functional data property `hasEndTime` indicates that if there is a relationship of an individual of the class `Context` along the property `hasEndTime`, it has to be to a member of the XSD `dateTime` datatype representing the end time of the interval in which the context is valid.

The class `LowLevelContext` represents the basic categories of low-level contexts via the classes `Activity`, `Location` and `Emotion` (Figure 4.3). The different recognized activities are modeled as 17 disjoint subclasses of the class `Activity`: `LyingDown`, `Sitting`, `Standing`, `Walking`, `Running`, `Cycling`, `Hiking`, `Stretching`, `Jumping`, `Dancing`, `Eating`, `Sweeping`, `ClimbingStairs`, `DescendingStairs`, `RidingElevator`, `RidingEscalator`, and `UnidentifiedActivity`. The names of the classes make reference to the activity they represent and

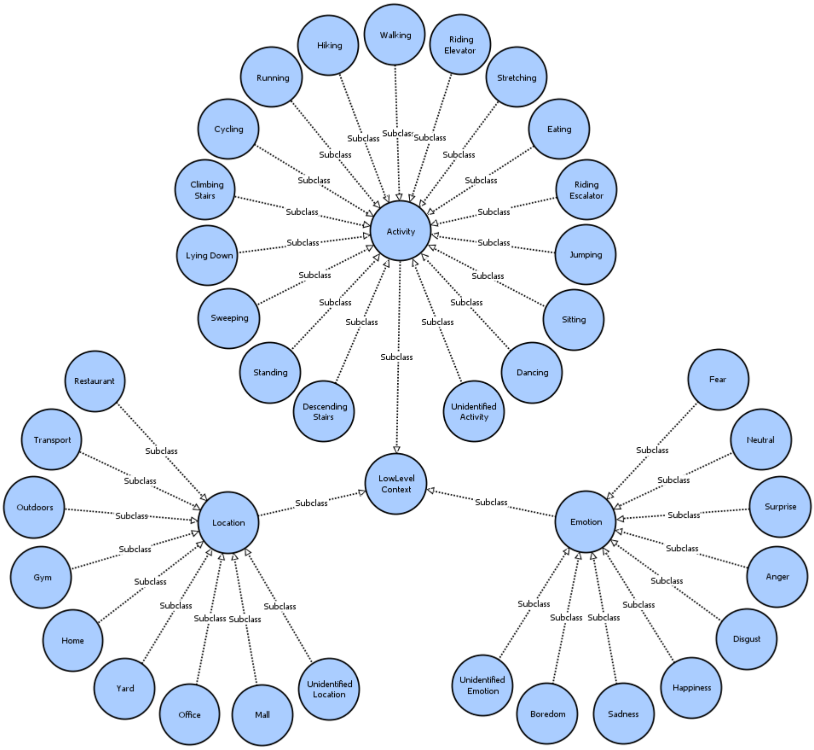


Figure 4.3: *Mining Minds Context Ontology: the class* `LowLevel-Context` *and its subclasses.*

the class `UnidentifiedActivity` represents any activity that does not belong to the other subclasses of `Activity`. The class `Location` has nine disjoint subclasses used to model the detected locations: `Home`, `Office`, `Restaurant`, `Gym`, `Mall`, `Transport`, `Yard`, `Outdoors`, and `UnidentifiedLocation` which represents any other location. The recognized emotions are modeled through the nine disjoint subclasses of the class `Emotion`: `Happiness`, `Sadness`, `Anger`, `Disgust`, `Fear`, `Boredom`, `Surprise`, `Neutral`, and `UnidentifiedEmotion` which represents any other emotion.

The class `HighLevelContext` models the concept of high-level context (Figure 4.2). The universal restrictions on the object properties `hasActivity`, `hasLocation` and `hasEmotion` model the relationship be-

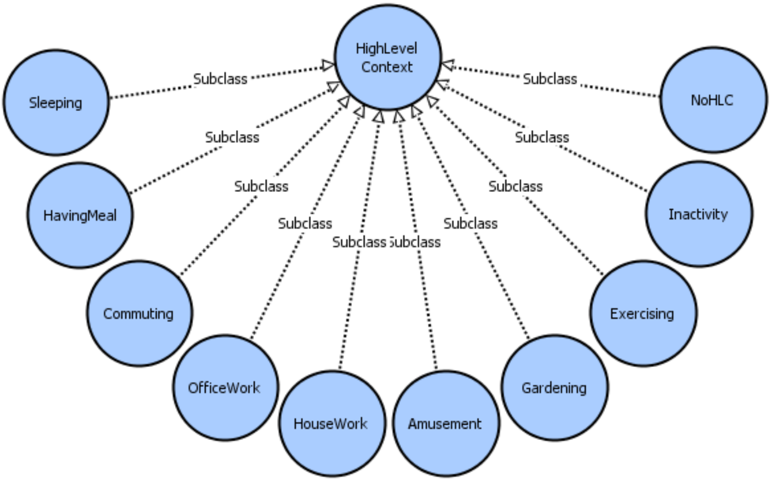


Figure 4.4: *Mining Minds Context Ontology: the class* `HighLevelContext` *and its subclasses.*

tween the individuals of the class `HighLevelContext` and the individuals of the different subclasses of `LowLevelContext`, which compose the high level context. The different types of high-level contexts are modeled via ten subclasses of the class `HighLevelContext` (Figure 4.4). Their equivalent anonymous classes are defined in Protégé, the open-source ontology editor [83]: `OfficeWork` (Figure 4.5), `Sleeping` (Figure 4.6), `HouseWork` (Figure 4.7), `Commuting` (Figure 4.8), `Amusement` (Figure 4.9), `Gardening` (Figure 4.10), `Exercising` (Figure 4.11), `HavingMeal` (Figure 4.12), `Inactivity` (Figure 4.13), and `NoHLC` (Figure 4.14).

In order to be a member of the defined class `OfficeWork` (Figure 4.5), an individual of the class `HighLevelContext` must have a property of type `hasActivity` which relates to an individual of the class `Sitting`, and this property can only take as value an individual of the class `Sitting`. Moreover the individual of the class `HighLevelContext` must also have a property of type `hasLocation`, which relates to an individual of the class `Office` and only to an individual of the class `Office`. Finally, in case the individual of the class `HighLevelContext` has a property of type `hasEmotion`, this property must relate to an

```
HighLevelContext
  and (hasActivity some Sitting)
  and (hasLocation some Office)
  and (hasActivity only Sitting)
  and (hasEmotion only (Anger or Boredom or Disgust or Happiness or Neutral))
  and (hasLocation only Office)
```

Figure 4.5: *Mining Minds Context Ontology: Definition of the class `OfficeWork`.*

```
HighLevelContext
  and (hasActivity some LyingDown)
  and (hasLocation some Home)
  and (hasActivity only LyingDown)
  and (hasEmotion only Neutral)
  and (hasLocation only Home)
```

Figure 4.6: *Mining Minds Context Ontology: Definition of the class `Sleeping`.*

```
HighLevelContext
  and (hasActivity some (Standing or Sweeping or Walking))
  and (hasLocation some Home)
  and (hasActivity only (Standing or Sweeping or Walking))
  and (hasEmotion only (Anger or Boredom or Disgust or Happiness or Neutral))
  and (hasLocation only Home)
```

Figure 4.7: *Mining Minds Context Ontology: Definition of the class `HouseWork`.*

individual of the class `Anger`, the class `Boredom`, the class `Disgust`, the class `Happiness` or the class `Neutral`. This universal restriction does not specify that the relationship along the property `hasEmotion` must exist, but if it exists, it must link to the specified class members.

To be a member of the defined class `Amusement` (Figure 4.9), an individual of the class `HighLevelContext` must have a property of type `hasActivity` which relates to an individual of the class `Dancing`, the class `Sitting`, the class `Standing`, or the class `Walking`, and this property can only take as value an individual of one of these four classes: `Dancing`, `Sitting`, `Standing` or `Walking`. Moreover the individual of the class `HighLevelContext` must also have a property of type `hasLocation`

```
HighLevelContext
  and (hasActivity some (Sitting or Standing))
  and (hasLocation some Transport)
  and (hasActivity only (Sitting or Standing))
  and (hasLocation only Transport)
```

Figure 4.8: *Mining Minds Context Ontology: Definition of the class* `Commuting`*.*

```
HighLevelContext
  and (hasActivity some (Dancing or Sitting or Standing or Walking))
  and (hasEmotion some Happiness)
  and (hasLocation some Mall)
  and (hasActivity only (Dancing or Sitting or Standing or Walking))
  and (hasEmotion only Happiness)
  and (hasLocation only Mall)
```

Figure 4.9: *Mining Minds Context Ontology: Definition of the class* `Amusement`*.*

```
HighLevelContext
  and (hasActivity some (Standing or Sweeping or Walking))
  and (hasLocation some Yard)
  and (hasActivity only (Standing or Sweeping or Walking))
  and (hasEmotion only (Happiness or Neutral))
  and (hasLocation only Yard)
```

Figure 4.10: *Mining Minds Context Ontology: Definition of the class* `Gardening`*.*

which relates to an individual of the class `Mall` and only to an individual of the class `Mall`. Finally, the individual of the class `HighLevelContext` must also have a property of type `hasEmotion` which relates to an individual of the class `Happiness` and only to an individual of the class `Happiness`. Summarizing, an individual of the class `HighLevelContext` has to fulfill the described existential and universal restrictions on the properties `hasActivity`, `hasLocation` and `hasEmotion` in order to be inferred as a member of the class `Amusement`. Hence, the assertion of an individual of the class `Happiness` for the property `hasEmotion` is mandatory to infer the class `Amusement`. The type of the restrictions on the property `hasEmotion` is the main modeling difference between the class `Amusement` and the previously presented class `OfficeWork`.

```
HighLevelContext
 and
  (((hasActivity some Hiking)
    and (hasLocation some Outdoors)
    and (hasActivity only Hiking)
    and (hasLocation only Outdoors))
  or
  ((hasActivity some Stretching)
    and (hasLocation some (Gym or Home or Office or Outdoors))
    and (hasActivity only Stretching)
    and (hasLocation only (Gym or Home or Office or Outdoors)))
  or
  ((hasActivity some (ClimbingStairs or DescendingStairs))
    and (hasLocation some (Gym or Home or Office))
    and (hasActivity only (ClimbingStairs or DescendingStairs))
    and (hasLocation only (Gym or Home or Office)))
  or
  ((hasActivity some (Cycling or Jumping or Running))
    and (hasLocation some (Gym or Outdoors))
    and (hasActivity only (Cycling or Jumping or Running))
    and (hasLocation only (Gym or Outdoors))))
 and (hasEmotion only (Happiness or Neutral))
```

Figure 4.11: *Mining Minds Context Ontology: Definition of the class* `Exercising`.

In the definition of the class `Amusement` the property `hasEmotion` is mandatory due to existential and universal restrictions on this property, whereas in the definition of the class `OfficeWork` the property `hasEmotion` is optional since the restriction on this property is universal but not existential.

To be a member of the defined class `Inactivity` (Figure 4.13), an individual of the class `HighLevelContext` must not be member of the class `Amusement`, the class `Commuting`, the class `Exercising`, the class `Gardening`, the class `HavingMeal`, the class `HouseWork`, the class `OfficeWork`, or the class `Sleeping`, i.e., it must not be a member of other subclasses of `HighLevelContext`. Moreover the individual of the class `HighLevelContext` must also have a property of type `hasActivity` which relates to an individual of the class `LyingDown`, the class `RidingElevator`, the class `RidingEscalator`, the class `Sitting`, or the class `Standing`, and this property can only take as value an individual of one of these five classes: `LyingDown`, `RidingElevator`, `RidingEscalator`, `Sitting`, or `Standing`. In the modeling of the class `Inactivity`, not only existential and universal restrictions are used, but also the concept of complement class.

```
HighLevelContext
  and
    (((hasActivity some Eating)
      and (hasLocation some (Home or Restaurant))
      and (hasActivity only Eating)
      and (hasLocation only (Home or Restaurant)))
    or
    ((hasActivity some Sitting)
      and (hasLocation some Restaurant)
      and (hasActivity only Sitting)
      and (hasLocation only Restaurant)))
  and (hasEmotion only (Disgust or Happiness or Neutral or Surprise))
```

Figure 4.12: *Mining Minds Context Ontology: Definition of the class* `HavingMeal`*.*

```
HighLevelContext
  and (not (Amusement or Commuting or Exercising or Gardening or HavingMeal
            or HouseWork or OfficeWork or Sleeping))
  and (hasActivity some
        (LyingDown or RidingElevator or RidingEscalator or Sitting or Standing))
  and (hasActivity only
        (LyingDown or RidingElevator or RidingEscalator or Sitting or Standing))
```

Figure 4.13: *Mining Minds Context Ontology: Definition of the class* `Inactivity`*.*

```
HighLevelContext
  and (not (hasActivity some Activity))
  and (not (hasEmotion some Emotion))
  and (not (hasLocation some Location))
```

Figure 4.14: *Mining Minds Context Ontology: Definition of the class* `NoHLC`*.*

The class `NoHLC` represents the absence of high-level context in an interval of time. This means that none of the low-level contexts are available in the time interval and therefore, it is not possible to identify a high-level context in that period of time. In order to model this concept, the complement class and existential restrictions are used in the definition of the class `NoHLC` (Figure 4.14). Thus, to be a member of the definded class `NoHLC`, an individual of the class `HighLevelContext` must not be a member of the anonymous class `hasActivity some Activity`, neither of the anonymous class `hasLocation some Location`, nor of the anonymous class `hasEmotion some Emotion`.

4.2.2. Instances of Context

An illustrative scenario is presented here to showcase the representation of instances of low-level contexts and high-level contexts in the Mining Minds Context Ontology (Figure 4.15). Let us imagine that it is 10 November 2015, and the user with identifier 9876 enters at 11:03:55 the office building of her or his working place. This event is detected by a location detector, a positioning system that interprets the coordinates of the user as the location of her or his office. Therefore, the low-level context of category location is identified as being of type *office* at 11:03:55. She or he starts talking on the phone, and a system capable of recognizing emotions detects from the tone of her or his voice that the user is bored. Thus, the low-level context of category emotion is identified as being of type *boredom* at 11:05:05. The phone call finalizes at 11:06:40, and then, no emotion is detected anymore. Meanwhile, at 11:05:25, the user sits down at her or his workplace. This event is detected by an activity recognizer that continuously measures her or his body motion. The low-level context of category activity is identified as being of type *sitting* at 11:05:25. It should be noted that every change in any of the low-level contexts may potentially lead to a new high-level context. For example, at 11:05:05, the combination of the activity *sitting*, the location *office* and the emotion *boredom* creates a high-level context that is classified as *office work*. At 11:06:40, when the emotion is no longer available, but the activity remains as *sitting* and the location as *office*, the high-level context for this user continues being identified as *office work*. Some combinations of low-level contexts do not constitute a known class of high-level context, based on the defined ontology. This is the case of the two high-level contexts at the beginning of this scenario. Namely, only location or the combination of the location *office* and the emotion *boredom* turn out to be not enough to identify a more abstract high-level context. Each context has associated a name, which serves as unique identifier. These names are automatically created by the system whenever a new context is detected and are composed of the prefix *"llc_"* or *"hlc_"* and a sequential unique number. For the sake of simplicity, in this example up to three digits are considered; however, large numbers are normally used by the system to procure unique identifiers. Furthermore, in order to make the example more understandable, for the low-level contexts it has been appended the membership of the instance to its name. For example, the context representing the activity *sitting* is named `llc_360_sitting`.

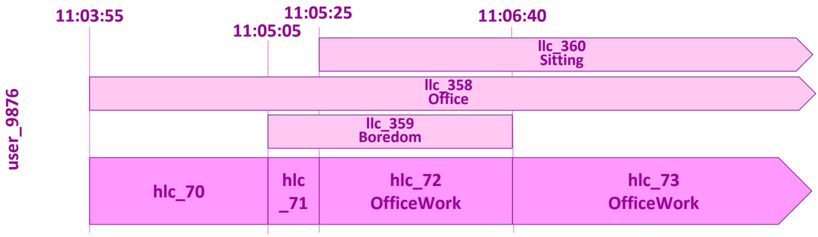
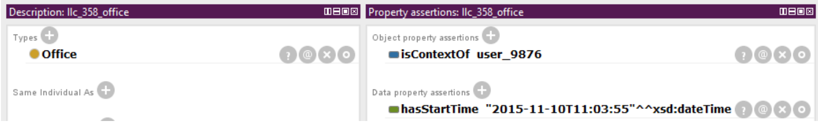


Figure 4.15: *Exemplary scenario representing low-level contexts and high-level contexts.*

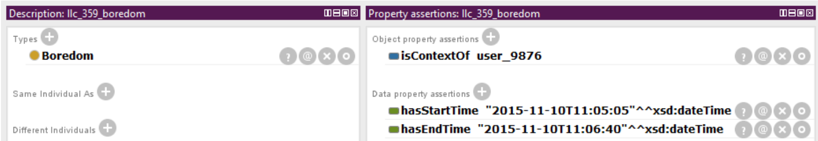
The terminology described in Section 4.2.1 is utilized at this point to generate the instances of context resulting from this scenario. The instances of low-level context are directly created from the information provided by the activity recognizer, location detector or emotion recognizer. In Section 4.2.2, the generation of the low-level contexts is presented. High-level contexts can be created from the information of the low-level contexts which are part of it and which triggered its occurrence. In Section 4.2.2, the generation of the high-level contexts is introduced. High-level contexts can also be classified, i.e., the membership of the high-level context or the class to which a high-level context belongs can be determined. In Section 4.2.2, the inference of the membership of the high-level contexts is described. Since the process of inferring the membership of a high-level context is also called classification, the high-level contexts for which their membership has been inferred are hereafter called *classified* high-level contexts. Conversely, the high-level contexts for which their membership has not been inferred are hereafter called *unclassified* high-level contexts. Finally, it is possible that the classification of an unclassified high-level context does not result in any inferred statement. In other words, the high-level context does not belong to any of the classes of high-level context defined in the terminology. In this case, the high-level context, which has been intended to be classified, but does not belong to any known class, is called *unidentified* high-level context.

Instances of Low-Level Context

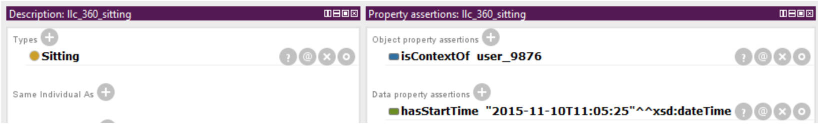
The low-level contexts are modeled as members of the subclasses of `LowLevelContext`: `Activity`, `Location`, and `Emotion`. Figure 4.16



(a) llc_358_office



(b) llc_359_boredom



(c) llc_360_sitting

Figure 4.16: *Representation of the instances of low-level context for the exemplary scenario by using the Mining Minds Context Ontology on Protégé.* `llc_358_office` *is a member of the class* `Office`*;* `llc_359_boredom` *is a member of the class* `Boredom`*; and* `llc_360_sitting` *is a member of the class* `Sitting`*.*

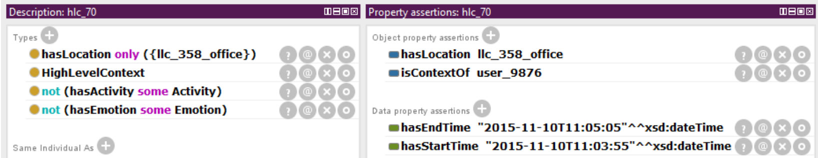
shows how the low-level contexts for the presented scenario are described in Protégé. `llc_358_office`, `llc_359_boredom`, and `llc_360_sitting` are members of the classes `Office`, `Boredom`, and `Sitting`, respectively. These instances model the low-level context of the user with identifier 9876. Thus, `llc_358_office`, `llc_359_boredom` and `llc_360_sitting` are related along the property `isContextOf` to the individual `user_9876` which is a member of the class `User`. All the individuals representing the low-level contexts have a relationship along the property `hasStartTime` to a value in the form of XSD `dateTime` which represents the start time of the interval in which the low-level context is valid. For example, for the individual `llc_359_boredom`, the property `hasStartTime` links to the value `“2015-11-10T11:05:05”^^dateTime`, which indicates that this context started at 11:05:05 on 10 November 2015. Moreover, for this very individual, the property `hasEndTime` relates to the value

"2015-11-10T11:06:40"^^dateTime, which means that this low-level context only occurred until 11:06:40 on 10 November 2015. Therefore, the individual llc_359_boredom models a low-level context of the type boredom for the user with identifier 9876 and which was valid in the period of time comprising from 11:05:05 to 11:06:40 on 10 November 2015.

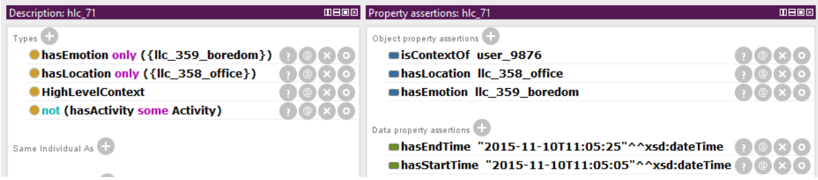
Instances of Unclassified High-Level Context

The unclassified high-level contexts are modeled as members of the class HighLevelContext for which their properties and types are stated. Property assertions are used to define the low-level contexts which compose the unclassified high-level context. The properties hasActivity, hasLocation and hasEmotion relate to the individuals of the subclasses of the classes Activity, Location and Emotion, respectively. Reasoning in OWL is based on the Open World Assumption (OWA), which means that it cannot be assumed that something does not exist unless it is explicitly stated that it does not exist. Therefore, type assertions are used as closure axioms to indicate that an unclassified high-level context is composed of a unique and finite set of low-level contexts. Specifically, for each of the low-level contexts components of the high-level context, it is stated the type equivalent to the anonymous class represented by the universal restriction on the property hasActivity, hasLocation or hasEmotion where the value of the filler is the collection comprising only the low-level context. Furthermore, type assertions are also used as closure axioms to indicate that there is no low-level context of a specific category being part of the unclassified high-level context. In this case, for each of the categories of low-level context absent on the unclassified high-level context, it is stated the type equivalent to the anonymous class which is the negation class of the existential restriction on the property hasActivity, hasLocation or hasEmotion where the filler is the class representing the category of low-level context, Activity, Location or Emotion, respectively.

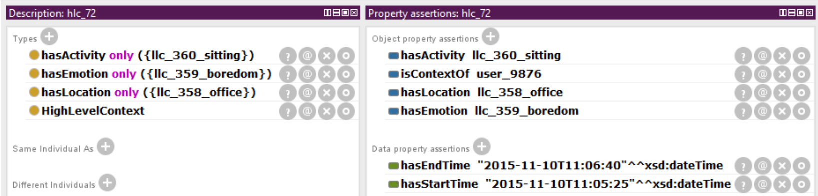
Figure 4.17 shows how the unclassified high-level contexts for the presented scenario are described in Protégé. hlc_70, hlc_71, hlc_72, and hlc_73 are members of the class HighLevelContext. Similarly as for the low-level contexts, the individuals representing the unclassified high-level contexts have relationships along the properties isContextOf, hasStartTime, and hasEndTime. For the individual hlc_72, the property hasActivity relates to the



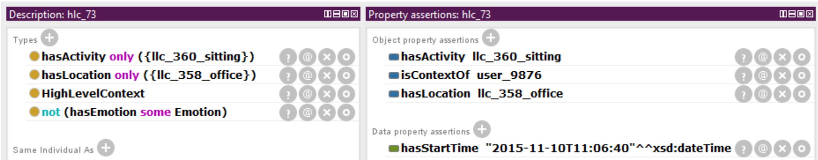
(a) hlc_70



(b) hlc_71



(c) hlc_72



(d) hlc_73

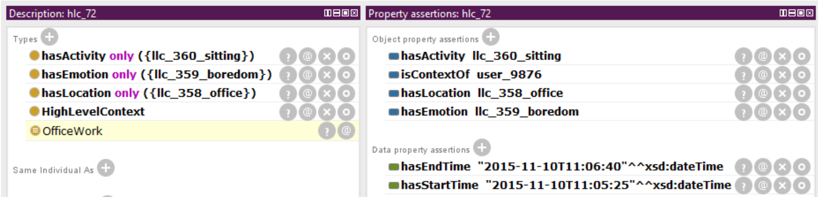
Figure 4.17: *Representation of the instances of unclassified high-level context for the exemplary scenario by using the Mining Minds Context Ontology on Protégé. The unclassified high-level contexts `hlc_70`, `hlc_71`, `hlc_72`, and `hlc_73` are composed of some of the low-level contexts `llc_358_office` (member of the class `Office`), `llc_359_boredom` (member of the class `Boredom`) and `llc_360_sitting` (member of the class `Sitting`).*

individual llc_360_sitting, the property hasLocation to the individual llc_358_office, and the property hasEmotion to the individual llc_359_boredom. Due to the OWA, hlc_72 has been asserted the type hasActivity only ({llc_360_sitting}), the type hasLocation only ({llc_358_office}), and the type hasEmotion only ({llc_359_boredom}). These statements indicate that the individual hlc_72 only has a hasActivity relationship to llc_360_sitting, a hasLocation relationship to llc_358_office and a hasEmotion relationship to llc_359_boredom. The individual hlc_73 is composed of the same activity and location as hlc_72; however, no emotion is part of this unclassified high-level context. Therefore, hlc_73 has been asserted the type not (hasEmotion some Emotion). This statement indicates that the individual hlc_73 does not have any property of type hasEmotion linking to an individual of the class Emotion, i.e., this unclassified high-level context does not contain any emotion.

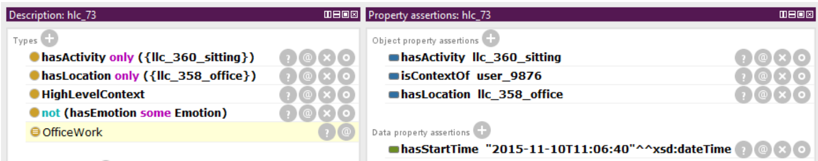
Instances of Classified High-Level Context

The classified high-level contexts are obtained using a reasoner which infers the membership of the unclassified high-level contexts. Thus, a classified high-level context is an individual of the class HighLevelContext, which is determined to be also a member of one of the ten subclasses of HighLevelContext: OfficeWork, Sleeping, HouseWork, Commuting, Amusement, Gardening, Exercising, HavingMeal, Inactivity or NoHLC. Figure 4.18 shows the classified high-level contexts for the working scenario and which have been inferred in Protégé using the Pellet reasoner [104]. The individuals hlc_70 and hlc_71 are not presented in the figure since they do not belong to any known class of high-level context, i.e, they are unidentified high-level contexts.

The individual hlc_72 is inferred by the reasoner to belong to the class OfficeWork (Figure 4.18(a)). Since this individual of the class HighLevelContext complies with the definition of the class OfficeWork, it is classified as being a member of this class. hlc_72 fulfills the existential and universal restrictions on the property hasActivity, which state that a member of the class OfficeWork must have some hasActivity relationship to an individual of the class Sitting and only to a member of this class. These restrictions are met since the property hasActivity only links the individual hlc_72 to the individual llc_360_sitting, which is a member of the class Sitting.



(a) hlc_72



(b) hlc_73

Figure 4.18: *Representation of the instances of classified high-level context for the exemplary scenario by using the Mining Minds Context Ontology on Protégé. The classified high-level contexts* `hlc_72` *and* `hlc_73`*, which are both inferred to be members of the class* `OfficeWork`*, are composed of some of the low-level contexts* `llc_358_office` *(member of the class* `Office`*),* `llc_359_boredom` *(member of the class* `Boredom`*) and* `llc_360_sitting` *(member of the class* `Sitting`*).*

Similarly, `hlc_72` also fulfills the existential and universal restrictions on the property `hasLocation`. Furthermore, `hlc_72` fulfills the universal restriction on the property `hasEmotion`, which states that in the case a member of the class `OfficeWork` has a `hasEmotion` relationship, it has to link to only an individual of the class `Boredom`. In fact, `hlc_72` is only related along the property `hasActivity` to the individual `llc_359_boredom`, which is a member of the class `Boredom`.

The individual `hlc_73` is also classified by the reasoner as being a member of the class `OfficeWork` (Figure 4.18(b)). Similar to the classified high-level context `hlc_72`, the individual `hlc_73` also complies with the existential and universal restrictions on the properties `hasActivity` and `hasLocation`. However, the property `hasEmotion` about the individual `hlc_73` is not asserted. The universal restriction on the property `hasEmotion` does not state that the relationship must exist. In fact, it may not exist at all and the restriction still be fulfilled, as it is the case for `hlc_73`. Thus, the individual `hlc_73` can be inferred as being a member of the class `OfficeWork`. The classification as members of the class `OfficeWork` of the two individuals of the class `HighLevelContext`, `hlc_72` and `hlc_73`, one with a `hasEmotion` relationship and another without it, proves the flexibility of the Mining Minds Context Ontology, which enables the identification of high-level contexts, even if one of the pieces of low-level information is missing. This is considered to be helpful in real-life scenarios where emotion recognition systems are not always available or may generate detection events in a less regular basis than activity recognizers or location detectors.

4.3. A Method for the Inference of High-Level Context

This section presents a method based on ontology reasoning to automatically infer rich and meaningful human context to enhance the operation of behavior-aware systems. This method infers abstract context representations based on categories, such as physical activities, emotional states and locations. These categories, which are derived from the wide-spectrum of multimodal data obtained from the user interaction with the real- and cyber-world, are intelligently combined and processed in order to determine and track the user context. The inferred user context can be utilized to provide personalized health and wellness services. The inference method relies on the Mining Minds Context Ontology (Section 4.2) and applies OWL 2 DL reasoning to identify abstract user context.

The novel context inference method involves several steps which are graphically represented in Figure 4.19. Conversely to most similar approaches, this method supports the instance-based identification of context. Every time unstructured low-level information, namely activities, emotions and locations, are identified by a low-level context recognizer, the method to infer more abstract high-level context begins. In the first phase of the method, the ontological concepts representing the user context are generated. Thus, the received low-level information is interpreted and transformed into the ontological concept representing this type of low-level context. This new instance of low-level context is stored for persistence. Then, other low-level contexts valid at the same moment in time are identified. After that, a new instance of an unclassified high-level context linking to the low-level contexts that compose it is generated. In the second phase of the method, the unclassified high-level context is verified and classified. Therefore, the semantic and

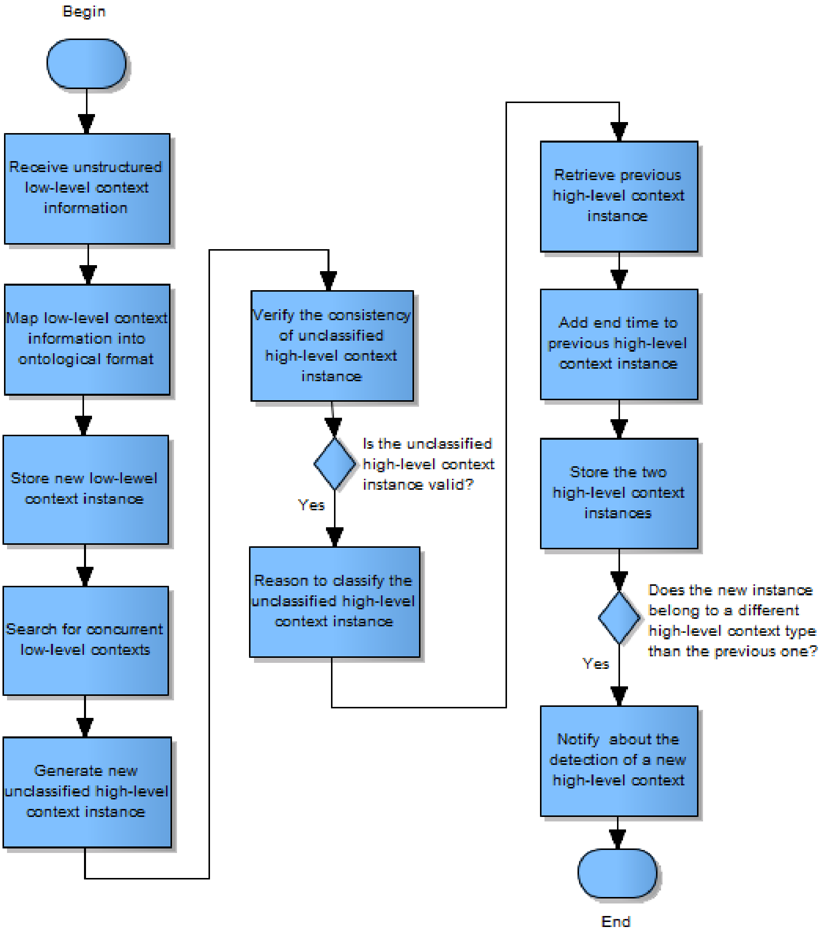


Figure 4.19: *Diagram depicting the steps involved in the context inference method.*

syntactic consistency of the unclassified high-level context are checked. If the unclassified high-level context is valid, it is classified, i.e., the membership of the unclassified high-level is identified by applying ontological reasoning. In the third and last phase of the method, the newly classified high-level context is made available to any third party application that registered for this type of information. The identification

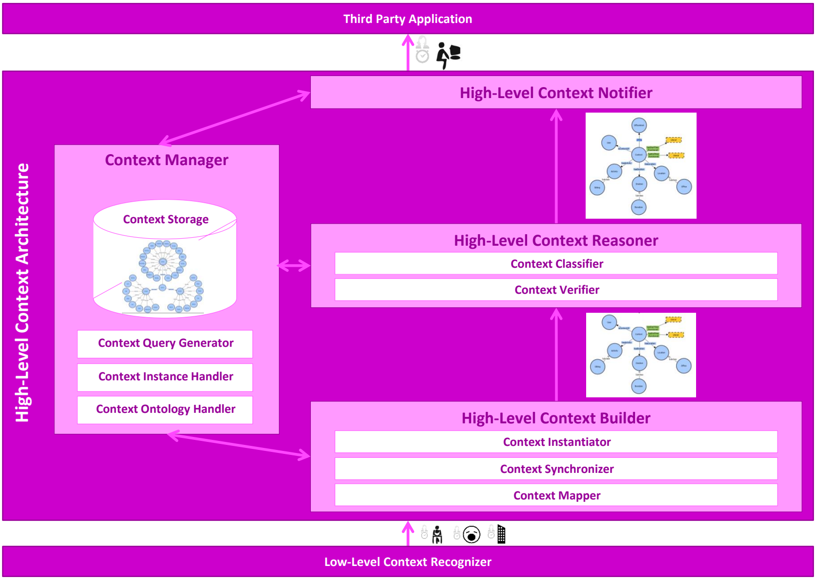


Figure 4.20: *High-Level Context Architecture (HLCA) which implements the method to infer high-level context.*

of a new high-level context implies that the previous one is not valid anymore. Thus, the previous high-level context is retrieved and its end time is set. Then, the two instances are stored for their persistence. Finally, if the new instance belongs to a different high-level context type than the previous one, the change in the high-level context is notified.

In order to implement the proposed ontology-based method for the inference of meaningful context, a system architecture, named the High-Level Context Architecture (HLCA), is designed and specified (Figure 4.20). The HLCA consists of four main components: High-Level Context Builder (Section 4.3.1), High-Level Context Reasoner (Section 4.3.2), High-Level Context Notifier (Section 4.3.3), and Context Manager (Section 4.3.4). In the following the different components of the HLCA are described in detail. For the sake of understanding, an example from the scenario presented in Section 4.2.2 is here considered to illustrate the operation of each component of the HLCA. Namely, the inference of a new high-level context at 11:05:25 on 10 November 2015 is considered. At that moment, a new low-level context of

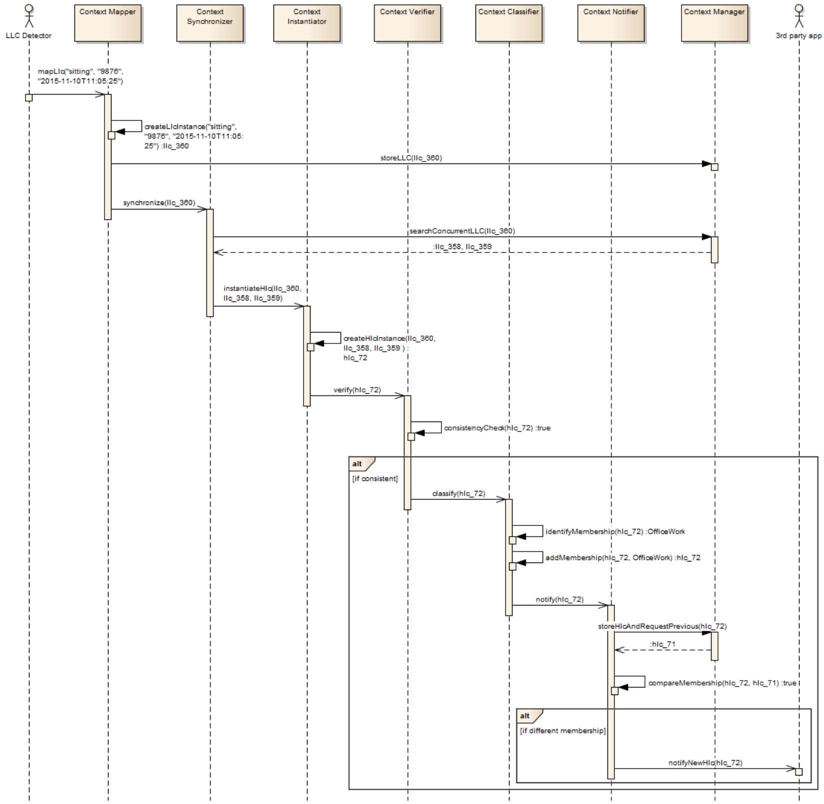


Figure 4.21: *Sequence diagram representing the interaction among component of the High-Level Context Architecture (HLCA).*

the category *sitting* for the user with identifier 9876 is detected by an external low-level context recognizer. This event triggers the novel context inference method implemented by the HLCA. After the method has been applied, a new high-level context of type *office work* is identified and served to the registered third party applications. Figure 4.21 presents the sequence diagram of the interactions among components of the HLCA in the application of the context inference method for this particular example.

4.3.1. High-Level Context Builder

The High-Level Context Builder receives the low-level information, i.e., activities, emotions, and locations, and generates the ontological concepts representing an unclassified high-level context associated with that information. The High-Level Context Builder has three subcomponents: the Context Mapper, the Context Synchronizer, and the Context Instantiator.

Context Mapper

The Context Mapper interprets the received low-level information and transforms it into the corresponding ontological concepts. Specifically, it maps the labels plus metadata into ontological instances of low-level context. Whenever the Context Mapper gets a new label, it creates an instance of the subclass of the class `LowLevelContext` which represents the corresponding activity, location or emotion (as described in Section 4.2.2). The property `hasStartTime` is stated to relate this instance to the time in which the low-level context started and which is part of the received metadata. Furthermore, the user to which the context belongs is related along the property `isContextOf`. Once the low-level context instance has been created, it is stored in the Context Manager for its persistence (see Section 4.3.4) and it is notified to the Context Synchronizer.

For the working example, the Context Mapper receives at run-time the activity label "sitting" and several metadata, i.e., the identifier of the user "9876" and the time in which the context starts "2015-11-10T11:05:25". The Context Mapper generates an instance of low-level context and then asserts the properties about it. The instance `llc_360_sitting` of the class `Sitting` presented in Figure 4.16(c) is created. This instance has a `isContextOf` relationship to the individual `user_9876` and a `hasStartTime` relationship to the value `"2015-11-10T11:05:25"^^dateTime`.

Context Synchronizer

The Context Synchronizer searches for concurrent low-level contexts, whenever the Context Mapper has notified a newly detected low-level context instance. A change in the low-level context implies a new high-level context, comprising the new low-level context and the other low-level contexts still valid at the start of the new low-level context. The

Context Synchronizer needs to determine the other low-level contexts of a given user which are valid a the start time of the new low-level context instance created by the Context Mapper. Therefore, one of the most important roles of the Context Synchronizer is to align concurrent low-level contexts of the same user which might have been received in an unordered manner due to the diverse delays introduced by different low-level context recognizers. In order to search for the concurrent low-level contexts, the Context Synchronizer requests information stored in the Context Manager and accesses it through the Context Instance Handler (see Section 4.3.4). Once the Context Synchronizer has determined the low-level contexts concurrent to the one that triggered the process, the Context Instantiator is invoked.

In the considered example, when the Context Synchronizer is notified by the Context Mapper about the identification of the new low-level context represented by the instance `llc_360_sitting`, it searches for concurrent low-level contexts by querying the information stored in the Context Manager. The instances `llc_358_office` and `llc_359_boredom`, presented in Figure 4.16(a) and in Figure 4.16(b), are found to be concurrent to the low-level context `llc_360_sitting`. These two low-level contexts belong to the same user, i.e., user with identifier 9876, and they are still valid at 11:05:25 on 10 November 2015, when the new low-level context `sitting` starts.

Context Instantiator

The Context Instantiator creates a new instance of an unclassified high-level context linking to the constituent low-level contexts. Whenever the Context Synchronizer detects a set of low-level contexts which are concurrent to a newly detected one, the Context Instantiator creates a new instance of an unclassified high-level context containing these low-level contexts (as described in Section 4.2.2). Therefore, an instance of the class `HighLevelContext` is created and the different low-level contexts which compose the high-level context are related to it along the properties `hasActivity`, `hasLocation`, and `hasEmotion`. Moreover, the closure axioms are established via type assertions on these properties. In case there is a low-level context of a particular type, the Context Instantiator generates the axiom stating that the property can only link to that given low-level context. Otherwise, if no low-level context has been determined for one of the categories -activities, locations or emotions-, the Context Instantiator creates the axiom stating that there is no low-level context of that category. Furthermore, the Context Instantiator establishes a `hasStartTime` relationship to the time in which the high-level context change happened, i.e., the time in which the newly detected low-level context started and which triggered the creation of the new unclassified high-level context. Moreover, the user to which the high-level context belongs is related along the property `isContextOf`. The identifier of the user to which the high-level context belongs is the same than the one associated to the low-level contexts which compose the high-level context. Once the Context Instantiator has created the instance of an unclassified high-level context, this is served to the High-Level Context Reasoner (see Section 4.3.2) for its verification and classification.

For the working example, the Context Instantiator receives from the Context Synchronizer the newly detected low-level context represented by the instance `llc_360_sitting` and the concurrent low-level contexts `llc_358_office` and `llc_359_boredom`. The Context Instantiator creates the instance `hlc_72` of the class `HighLevelContext` (see Figure 4.17(c)) and links it to the low-level contexts which compose it. Therefore, the properties `hasActivity`, `hasLocation` and `hasEmotion` relate, respectively, to the instances `llc_360_sitting`, `llc_358_office` and `llc_359_boredom`. The closure axiom `hasActivity only ({llc_360_sitting})` indicates that the individual `hlc_72` only has a `hasActivity` relationship to the individual `llc_360_sitting`. Similarly, the other two closure axioms, `hasLocation only ({llc_358_office})` and `hasEmotion only ({llc_359_boredom})`, state the uniqueness of the relationships. The Context Instantiator also specifies that the instance `hlc_72` has a `isContextOf` relationship to the individual `user_9876` which is the owner of the different low-level contexts composing the high-level context. Finally, the Context Instantiator creates a relationship along the property `hasStartTime` to the moment in which the change in the low-level context triggered the identification of the new high-level context. The start time of the high-level context `hlc_72` is the start time of the low-level context `llc_360_sitting`. Thus, for the instance `hlc_72` the property `hasStartTime` links to the value `"2015-11-10T11:05:25"^^dateTime`.

4.3.2. High-Level Context Reasoner

The High-Level Context Reasoner performs a consistency check on the unclassified high-level context instance created by the High-Level Context Builder (see Section 4.3.1). In case the instance is valid, the High-Level Context Reasoner identifies the context type to which the high-level context belongs, i.e., it classifies the high-level context instance. In order to perform these tasks, the High-Level Context Reasoner applies ontological inference supported by the formal description of context in the Mining Minds Context Ontology (see Section 4.2.1). The High-Level Context Reasoner comprises two subcomponents: the Context Verifier and the Context Classifier.

Context Verifier

The Context Verifier checks the semantic and syntactic consistency of the unclassified high-level context provided by the High-Level Context Builder. Therefore, the instance of unclassified high-level context is validated and verified versus the Mining Minds Context Ontology, which is stored in the Context Manager and can be accessed through the Context Ontology Handler (see Section 4.3.4). During the consistency check, non-logical or malformed high-level contexts can be detected. For example, the high-level contexts which do not contain the necessary property `hasStartTime` or the ones composed from multiple different instances of low-level contexts of the same type. Once the Context Verifier has ensured that the unclassified high-level context is valid, this instance is provided to the Context Classifier for further processing.

In the described example, the Context Verifier receives from the Context Instantiator the newly created high-level context `hlc_72`. This instance is checked for its semantic and syntactic consistency, it is considered to be valid, and it is then served to the Context Classifier.

Context Classifier

The Context Classifier identifies the type of high-level context to which the unclassified high-level context belongs; thus, converting the unclassified instance into a classified high-level context. The classification of the unclassified high-level context instance into one of the defined high-level context classes is based on the inference functionalities provided by the Mining Minds Context Ontology.

Specifically, one of the key features of this ontology is that it can be processed by a reasoner which can automatically perform the classification process. This means that the unclassified high-level context instance is compared versus the definitions of the different high-level context classes to determine whether it complies with the conditions that define the class. In case it complies, the instance is inferred to belong to that class. The classification process is triggered every time the Context Classifier receives a new valid instance of high-level context from the Context Verifier. After the membership of the unclassified high-level context instance has been determined, the Context Classifier adds to the unclassified high-level context instance the axiom stating that this instance belongs to a specific type of high-level context. Therefore, the instance of the class `HighLevelContext` which models the classified high-level context is related along the property `rdf:type` to the subclass of the class `HighLevelContext` representing the high-level context of which the instance is a member. It is possible that the unclassified high-level context does not belong to any of the known classes described in the Mining Minds Context Ontology. This means that no membership is inferred and the unclassified high-level context is considered to belong to an unidentified type of high-level context. In this case, the classified high-level context has the same exact representation than the corresponding unclassified high-level context. Finally, the Context Classifier serves the classified high-level context to the High-Level Context Notifier (see Section 4.3.3).

For the working example, the Context Classifier receives from the Context Verifier the high-level context `hlc_72`. The Context Classifier applies the classification method to this unclassified high-level context in order to determine its membership. The individual `hlc_72` is inferred to belong to the class `OfficeWork` since it complies with the definition of the class `OfficeWork` (as described in Section 4.2.2). Therefore, the Context Classifier creates the axiom `hlc_72 rdf:type OfficeWork` which indicates that the individual `hlc_72` is a member of the class `OfficeWork`. The classified high-level context instance `hlc_72` is provided to the High-Level Context Notifier for its notification.

4.3.3. High-Level Context Notifier

The High-Level Context Notifier makes available to third party applications the newly identified high-level contexts. The High-Level Context Notifier receives from the High-Level Context Reasoner a classified high-level context instance and notifies the subscribed third parties about the detection of a new high-level context. This notification is only conducted if the new instance belongs to a high-level context type different than the previous one. Only changes in the high-level context type are notified, this means that differences in the low-level context composition which do not imply a change on the type of high-level context are not communicated to the third parties. Furthermore, the High-Level Context Notifier stores the new high-level context into the Context Manager for its persistence via the Context Instance Handler (see Section 4.3.4) and gets as an answer from this component the previous valid high-level context.

For the described example, the High-Level Context Notifier receives from the High-Level Context Reasoner the high-level context `hlc_72` which has been classified as `OfficeWork`. The High-Level Context Notifier contacts the Context Instance Handler for the persistence of the instance `hlc_72` into the Context Storage. Moreover, the High-Level Context Notifier receives from the Context Instance Handler the previous valid instance of high-level context `hlc_71`. The High-Level Context Notifier compares the membership of `hlc_72` to the membership of the previous valid high-level context `hlc_71`. The High-Level Context Notifier determines that there has been a change in the type of high-level context, the previous instance `hlc_71` was unidentified and the new instance `hlc_72` is `office work`. Therefore, the third parties are notified about the change in the high-level context modeled as the instance `hlc_72`.

4.3.4. Context Manager

The Context Manager persists the Mining Minds Context Ontology, including the terminology for the definition of context and the instances of context. Furthermore, this component eases the interactions with the persisted context information by facilitating the exchanges with the storage infrastructure. The Context Manager has four subcomponents: the Context Storage, the Context Ontology Handler, the Context Instance Handler and the Context Query Generator.

Context Storage

The Context Storage is a database which provides persistence for the storage of the Mining Minds Context Ontology, including both the context definition terminology and the context instances. Since the context is modeled via an ontology and the context instances are represented as ontological instances, this storage is devised to be a database of the type triple store. Moreover, the Context Storage also provides read and write functionalities for the Mining Minds Context Ontology. However, this storage cannot be directly accessed and all the interactions are handled through the Context Ontology Handler and the Context Instance Handler.

Context Ontology Handler

The Context Ontology Handler provides the management functionalities to interact with the Mining Minds Context Ontology terminology stored in the Context Storage. This component enables loading the context ontology to the Context Storage at the system start time. The Context Ontology Handler also supports the retrieval of the context ontology which is stored in the Context Storage, so that the rest of components of the HLCA have access to the latest version of the ontological terminology. Furthermore, the Context Ontology Handler enables the extension at runtime of the context ontology. The extensibility is required to evolve the context ontology, therefore, including new types of low-level contexts and new definitions for the high-level contexts. Every time the ontology is updated, the rest of components of the HLCA making direct use of the context ontology are notified to obtain an updated version of the terminology.

Context Instance Handler

The Context Instance Handler deals with the retrieval and storage of context information in the Context Storage. The Context Instance Handler offers three different functionalities: storage of a newly mapped low-level context, retrieval of concurrent low-level contexts, and storage of a newly inferred high-level context while retrieving the previous valid high-level context. The Context Instance Handler poses to the Context Storage the SPARQL queries [9] created by the Context Query Generator in order to retrieve the persisted context information. Specifically, the logic of the Context Instance Handler for the storage of a newly inferred high-level context is as follows. The identification of a new high-level context implies that the previous context for the given user is not valid anymore. Therefore, the storage process includes the finalization of the previous valid high-level context instance. This operation entails to set the value of the end time of the previous valid high-level context stored in the Context Storage. In order to find the previous valid high-level context, the Context Instance Handler needs to pose the appropriate SPARQL queries to the Context Storage. The Context Query Generator is invoked to create the queries for the previous valid high-level context based on the newly inferred high-level context instance (see Section 4.3.4). Furthermore, it must be noted that a an earlier new high-level context could be inferred after the classification of a posterior one. This scenario is not very common but could happen due to the different delays in the data-driven recognition process for the low-level contexts. If this situation occurs, the newly inferred high-level context is only valid until the start time of the posterior high-level context already stored in the Context Storage. Therefore, the storage process also includes the finalization of the newly inferred high-level context instance.

In the considered example, the High-Level Context Notifier interacts with the Context Instance Handler to persist the newly classified high-level context instance `hlc_72` and to retrieve the previously valid instance of high-level context. Therefore, the Context Instance Handler stores the instance `hlc_72` into the Context Storage. Moreover, the Context Instance Handler retrieves from the Context Storage the previously valid instance of high-level context. The previous high-level context is here an individual of the class `HighLevelContext` modeling the context of the user represented by the individual `user_9876` and which is valid at at 11:05:25 on 10 November 2015. In order to retrieve the previous high-level context for the instance `hlc_72`, the Context Instance Handler invokes the Context Query Generator which creates the SPARQL queries presented in Listing 4.1 and Listing 4.2. These queries are posed to the Context Storage which returns as the matching result the high-level context `hlc_71`. Then, the Context Instance Handler finalizes the previous high-level context instance `hlc_71`. This means that the individual `hlc_71` is related along the property `hasEndTime` to the value `“2015-11-10T11:05:25”^^dateTime`, which is the value for the property `hasStartTime` of the newly identified high-level context `hlc_72`. In this exemplary scenario, it is assumed that there are no delays in the recognition of the low-level contexts and therefore, there are no high-level contexts posterior to `hlc_72` which had already been detected.

Context Query Generator

The Context Query Generator is the component which generates the SPARQL queries [9] required by the Context Instance Handler in order to find the matching context instances stored in the Context Storage. The SPARQL queries are automatically created based on some information derived from the context instance that the Context Instance Handler provides to the Context Query Generator. The Context Query Generator is capable of generating several different SPARQL queries depending on the expected outcome required for each specific use case scenario.The Context Query Generator creates SPARQL queries for the identification of a low-level context still valid at the start time of a newly recognized low-level context, which belongs to the very user and which is of the same context category. The Context Query Generator also creates SPARQL queries for the identification of the start time of the next posterior low-level context which belongs to the actual user and which is of the same context category. The Context Query Generator can also create SPARQL queries for the identification of low-level contexts of a given user which are concurrent at the start time of a newly recognized low-level context instance. In addition, the Context Query Generator creates SPARQL queries for the identification of a high-level context which is still valid at the start time of a new high-level context and which belongs to the same user. Finally, the Context Query Generator creates SPARQL queries for the identification of the start time of the next posterior high-level context belonging to the same user.

The logic for the creation of SPARQL queries for the identification of a high-level context which is still valid at the start time of a new high-level context and which belongs to the same user is the following. There are two cases in which the previous high-level context is still valid , either it does not have an end time or its end time is posterior to the start time of the new high-level context. In the first case, the SPARQL needs to match a high-level context for the same user which has a start time previous to the start time of the new high-level context but does not have an end time. In the second case, the SPARQL needs to match a high-level context for the same user which has a start time previous to the start time of the new high-level context and an end time posterior to the start of the new high-level context.

The specific SPARQL queries to request the previous high-level context for the instance `hlc_72` are presented in Listing 4.1 and Listing 4.2. In the considered example, the previous high-level context for `hlc_72` is an individual of the class `HighLevelContext` which belongs to the user represented by the individual `user_9876` and which is valid at 11:05:25 on 10 November 2015. Therefore, the matching individual has to be a member of the class `HighLevelContext`, must have a `isContextOf` relationship to the individual `user_9876`, must have a `hasStartTime` relationship to a value less than or equal to

`“2015-11-10T11:05:25”^^dateTime`. Furthermore, it must not have any `hasEndTime` relationship (as in the query presented in Listing 4.1), or if it has such a relationship it must be to a value greater than `“2015-11-10T11:05:25”^^dateTime` (as in the query presented in Listing 4.2).

```
SELECT  ?hlc
WHERE {
    ?hlc  rdf:type  HighLevelContext ;
          isContextOf  user_9876 ;
          hasStartTime  ?starttime .
    FILTER NOT EXISTS ?hlc  hasEndTime  ?endtime .
    FILTER ( ?starttime <= "2015-11-10T11:05:25"^^xsd:dateTime )
}
```

Listing 4.1: *SPARQL query to request the previous high-level context for the instance `hlc_72` in a scenario without recognition delays.*

```
SELECT  ?hlc
WHERE {
    ?hlc  rdf:type  HighLevelContext ;
          isContextOf  user_9876 ;
          hasStartTime  ?starttime ;
          hasEndTime  ?endtime .
    FILTER ( ?starttime <= "2015-11-10T11:05:25"^^xsd:dateTime )
    FILTER ( ?endtime > "2015-11-10T11:05:25"^^xsd:dateTime )
}
```

Listing 4.2: *SPARQL query to request the previous high-level context for the instance `hlc_72` in a scenario with recognition delays.*

4.3.5. HLCA Implementation

The HLCA has been implemented in Java using available open source libraries. All the components of the HLCA build on Apache Jena (v2.11.2) [105], a semantic web framework which includes some APIs for handling RDF [106], OWL [6], and SPARQL [9]. In the implementation of the High-Level Context Reasoner, an off-the-shelf open source reasoner, namely Pellet (v2.3.2) [104], has been utilized in combination with Jena to enable the ontological inference functionalities. Furthermore, in the Context Manager, the Jena Triple Store (TDB) has been used as the Context Storage for the persistence of the Mining Minds

Context Ontology. The communication between the HLCA and the low-level context recognizers, which identify activities, locations and emotions, has been implemented by means of RESTful web services [107] and establishing service contracts among them. The same mechanism applies to the communication of the HLCA with the third party applications registered to get access to the newly identified high-level contexts.

4.4. Evaluation of the Context Ontology and Inference Method

This section analyzes the robustness of the proposed context ontology and the performance and reliability of the novel context inference method. Section 4.4.1 explores the tolerance offered by the Mining Minds Context Ontology for the inference of high-level contexts under the presence of low-level context errors. Section 4.4.2 studies the performance of the High-Level Context Architecture with respect to processing time and management of context instances. Section 4.4.3 evaluates the reliability of the context inference method implemented by the High-Level Context Architecture during realistic executions.

4.4.1. Robustness of the Context Ontology

The proposed Mining Minds Context Ontology (see Section 4.2) has been evaluated to determine how robust the identification of high-level contexts can be in the event of having erroneously detected low-level contexts. In other words, this evaluation aims at measuring the level of resilience of the high-level context level against errors originated at the low-level context level. Pellet (v2.3.2) [104], an open source OWL 2 DL reasoner for Java has been used in the evaluation test program. First, a set of 1,800 instances representing all the possible combinations of low-level contexts, i.e., activities, locations and emotions, have been generated. Then, the instances have been posed to the reasoner and the corresponding high-level contexts have been inferred. The resulting array of high-level contexts represents the ground-truth for this evaluation. Subsequently, various scenarios with increasing levels of error in the low-level contexts have been defined. Namely, 5, 10, 20 and 50 per cent of errors have been respectively introduced in the 1,800 instances as to emulate potentially erroneous low-level contexts. For example, in the case of having a 10% of affected instances a total of 180 randomly selected instances are deliberately modified. The error has been introduced by replacing the supposedly affected low-level context with a new value randomly selected from the remaining contexts in the affected category (activity, location or emotion). Thus for example, if the original instance of high-level context is composed of the contexts *sitting*, *office* and *boredom*, and the affected context is the activity, the newly generated instance could contain the contexts *running*, *office* and *boredom*. Moreover, in order to evaluate the prominence of each specific context category or combination thereof, the analysis has been formulated for all the combinations of low-level categories, i.e., introducing errors in solely the activity, location, emotion, or combination of activity and location, activity and emotion, location and emotion, and all activity, location and emotion. The instances resulting from all these scenarios have been posed to the reasoner and the resulting high-level contexts have been compared against the ground truth to determine the accuracy of the model. Each of the experiments has been repeated one hundred times in order to ensure the statistical robustness. The average and standard deviation accuracy is presented in Table 4.1 for each corresponding study.

Table 4.1: *Mean and standard deviation of the accuracy of the high-level context recognition under different levels of errors in the detected low-level contexts.*

Low-level Context	**5% Error**	**10% Error**	**20% Error**	**50% Error**
Activity (A)	97.60±0.05	95.13±0.05	90.39±0.04	75.32±0.20
Location (L)	99.45±0.02	98.82±0.05	97.61±0.15	93.93±0.02
Emotion (E)	99.63±0.02	99.18±0.05	98.32±0.05	96.04±0.07
A & L	97.08±0.10	94.27±0.16	88.48±0.11	72.63±0.10
A& E	97.16±0.12	94.22±0.06	89.60±0.10	73.53±0.30
L & E	99.00±0.05	98.02±0.09	96.24±0.05	91.25±0.09
A & L & E	96.56±0.06	93.10±0.30	87.52±0.11	71.60±0.13

From an overall analysis of the obtained results it can be concluded that the impact of the error introduced in the low-level context is generally lower at the high-level context. For example, in the case of introducing a 5% error, the accuracy drops approximately no more than 0.4% at best and 3.5% in the worst case scenario. Similarly, for the 10%, 20% and 50% error cases the minimum and maximum accuracy drops are below the corresponding level of error. Experiencing a lesser impact is generally due to the fact that not all the misrecognitions at the low-level context lead to an inference error at the high-level context. For example, if the activity *running* is recognized instead as *climbing stairs*, and provided the rest of low-level contexts to be *gym* for the location and *neutral* for the emotion, the inferred high-level context remains to be *exercising*. Similar examples in which the error in the low-level context is not propagated to the high-level context can be found for the case of erroneous locations and emotions. It can also be observed that the activity is the most prevalent category in terms of error impact, which is certainly as a consequence of the importance given to the activity context in the definition of high-level contexts. Conversely, the location and especially the emotion tend to show a lower effect on the high-level context. In fact the definition of some high-level contexts allows for a good level of resilience against errors in the locations and the emotions. This is the case of the high-level context *inactivity*, which is determined from a sole sedentary activity, like *lying down*, and nearly any location and emotional state. Therefore, even if an the location is erroneously detected, the inferred high-level context would result in *inactivity*. The only exception to this case would happen if the location is misreconized as *home*, since *lying down* at *home* and with a *neutral* emotional state is identified as the high-level context *sleeping*. Moreover, errors simultaneously present in various low-level contexts generally increase the chance of misidentification of the actual high-level context. Therefore, the combinations of errors in several low-level categories report a lower accuracy in the high-level context recognition than in the case of having only errors in a single category. As it was expected, the highest impact is observed when all three low-level contexts are subject to error. Either way, the error in the recognition of the high-level context remains below the level of error introduced in the considered low-level contexts. Finally, it must be noted that owing to the descriptive logic characteristic of ontologies, and conversely to probabilistic classification models, combinations of correct low-level contexts will always lead to a correctly inferred high-level context.

4.4.2. Performance of the Context Inference Method

In order to assess the performance of the proposed context inference method, the current implementation the HLCA (see Section 4.3.5) has been executed on a laptop operating Windows 10 with a 1.80 GHz

Intel Core i7 CPU, 8GB RAM, and a HDD with 5400-RPM spindle speed, I/O data-transfer rate up to 6 Gb/s and 16 MB buffer. Using a test Java application the low-level context recognizers have been emulated. The evaluation has consisted in the generation of 250,000 random low-level contexts belonging to 100 different users and which represented their context information for a time span of 16 days. First the category of the low-level context (activity, location or emotion) has been randomly selected and then one of the types for that category has also been randomly chosen. After that, the metadata associated to the low-level context label has been generated. The low-level context has been randomly assigned to one of the 100 users. The start time of each low-level context has also been randomly selected between 1 and 10 seconds after the start time of the previous low-level context. The generated low-level contexts, including the labels and the metadata, have been input one at a time to the HLCA for their mapping, synchronization, instantiation, verification, classification and notification. It is important to notice that the low-level contexts are served to the HLCA sequentially and at their simulated occurrence time. Thus, the HLCA works at real-time and processes each single instance on-the-fly right after receiving it. Concurrency is procured through user-based multithreading, thus supporting simultaneous processing of low-level contexts from different users taking place at the same time. Some resources such as the Context Storage are shared among threads (users). During the evaluation the time required for the context identification has been calculated and the volume of information generated and stored on the Context Storage has further been determined.

Figure 4.22 shows the time invested by each of the HLCA components and the system as a whole in the context identification process. The number of instances indicates the number of high-level contexts which have already been processed by the HLCA when the recognition process is triggered due to a change in the low-level context. Even if the context recognition process is performed instance-wise, the number of previously processed instances is important because of the volume of information generated by the system during the recognition process and persisted in the Context Storage. The processing times are further averaged to have an overall figure summarizing the time taken by the each component of the HLCA. Table 4.2 presents the mean and standard deviation of these times as well as the percentage of these timesdevoted to the interaction of the component with the Context Manager. This interaction is particularly relevant because the Context

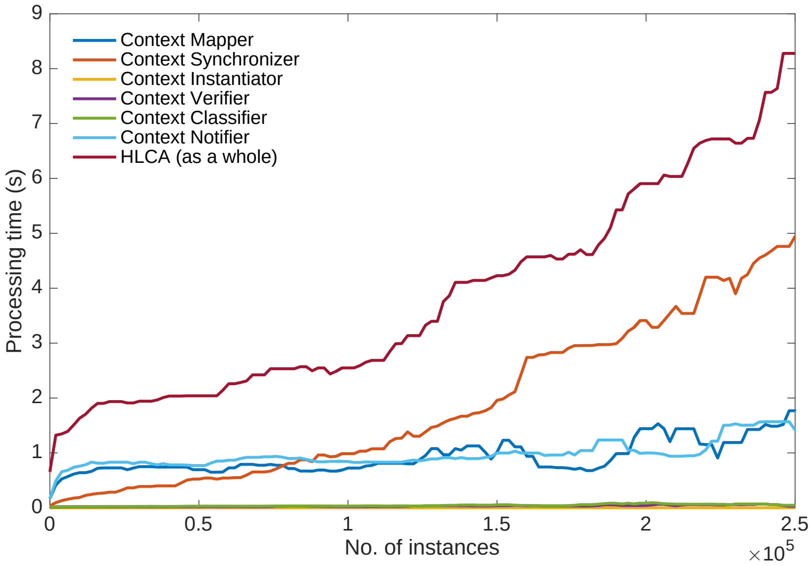


Figure 4.22: *Processing time invested by each of the HLCA components in the context identification. The number of instances indicates the amount of previously processed high-level contexts when the recognition process is triggered.*

Manager hosts the Context Storage, the shared resource which persists and loads the context information.

One can observe the differences of scale in the processing times for each of the components of the HLCA and the disparate tendencies of these times when the number of recognized context instances increases. The processes in which the HLCA component does not have any interaction with the Context Storage take much less time than the ones involving it. Furthermore, in the cases where the Context Storage is not involved, the processing time does not increase with the number of identified context instances.The Context Classifier and the Context Verifier take only some milliseconds to verify and classify the high-level context instance. This time is quite small due to the architectural design principle for which each single instance of high-level context is reasoned separately on-the fly at run-time. The Context Instantiator does not access either the Context Storage, since the required interactions to find the concurrent low-level contexts are performed by the

Table 4.2: *Mean and standard deviation of the processing time invested by each of the HLCA components in the context identification, as well as the percentage of these times devoted to the interaction with the Context Manager.*

Component	Processing Time (s)	Context Manager (%)
Context Mapper	0.986 ± 0.348	99.53
Context Synchronizer	2.188 ± 1.670	99.97
Context Instantiator	0.001 ± 0.000	0.00
Context Verifier	0.032 ± 0.014	0.00
Context Classifier	0.046 ± 0.019	0.00
Context Notifier	1.012 ± 0.268	99.99

Context Synchronizer. Therefore, the Context Instantiator takes only one millisecond to create a new instance of high-level context and this time does not increase with the number of instances because of the independence of the process from any other high-level context.

In case the components of the HLCA invoke the Context Storage, the processing times rise and the interactions with the Context Storage tend to represent most of the computational time, specifically more than 99%. This means that the actual component is relatively quick to perform its job but the context read and write processes which involve the Context Manager delay the complete process. The processing time for the Context Mapper and the High-Level Context Notifier follow a similar pattern. These processing times increase with the number of instances, at the beginning and with very few instances the times rocket, but then they stabilize and reach values around one second. The similarity in the evolution of the processing times for these two components is normal because their interactions with the Context Manager are of the same type. In the first case, the Context Mapper stores the new low-level context instance, retrieves the previous low-level context and after updating it, stores it again into the Context Manager. In the second case, the High-Level Context Notifier stores the new high-level context instance, retrieves the previous high-level context, compares them and after updating the previous instance, stores it again into the Context Manager. Therefore, the evolution of the processing time for operations that involve read and write to the Context Manager can be observed in the times for the Context Mapper and the High-Level Context Notifier. The processing performed by the Context Synchronizer in order to request concurrent low-level context instances is the most time demanding process of the HLCA. In this case, most of the time is devoted to the execution of the SPARQL queries and the retrieval of the matching solutions from the Context Manager. The processing time for the Context Synchronizer increases almost lineally with the number of instances. In fact, for few instances this time is below the processing time for the Context Mapper and High-Level Context Notifier, but then it becomes much higher. Therefore, the Context Synchronizer is the bottle neck of the HLCA, with a clear impact on the evolution of the time required for the context identification.

The relevance of the time invested by the HCLA to recognize a high-level context fairly depends on the application domain. Thus for example, if an alert has to be sent right away or a prompt action be taken based on the detected high-level context, then this time might be arguably long. However, if the identified information is rather used for analyzing the trajectories of behavior over time, then this time turns to be hardly relevant. Under these considerations, the processing time for the recognition of high-level contexts could be the main limitation of the actual implementation of the HLCA and should be improved in future work. A potential solution could consist in introducing a cache system into the High-Level Context Builder to save temporarily only the latest instances of low-level context and periodically persist them into the Context Manager. With such a solution the Context Synchronizer would not need to interact with the Context Manager and could pose the SPARQL queries directly to the cache; thus, retrieving the low-level context instances from a much smaller store. The Context Mapper, also part of the High-Level Context Builder, could share this very cache with the Context Synchronizer and increase its performance as well. If the cache would prove to be a good solution, such a system could also be introduced in the Context Notifier. This component has a similar behavior than the Context Mapper and its processing time could be reduced as well. Alternate solutions for accelerating the processing time for the identification of high-level contexts could include parallelizing tasks, defining different levels of cache-memory or simply scaling the infrastructure through cloud-based services.

Finally, Figure 4.23 depicts the size of the Context Storage in the Context Manager increasing lineally with the number of stored high-level context instances. The initialization of the Context Storage, i.e., storing the terminology defining the Mining Minds Context Ontology,

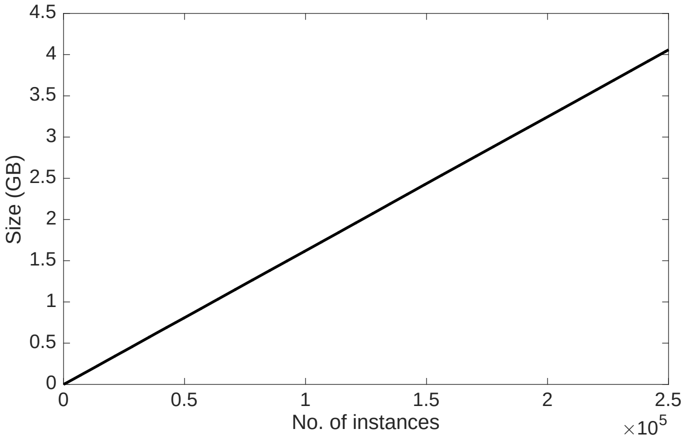


Figure 4.23: *Size of the Context Storage depending on the number of persisted instances of high-level context. It must be noted that the storage of each high-level context instance has associated the storage of the low-level context instance which triggered its creation. Thus, for example, 250,000 instances in the X-axis represent 250,000 high-level contexts plus 250,000 low-level contexts stored on disc.*

requires only 408.5 KB on disc. The storage of each new high-level context instance, which has associated the storage of the low-level context instance which triggered its creation, increases the size of the Context Storage in 17 KB, in average. Thus, for the previous simulation of 250,000 changes in the context, which leads to a total of 500,000 context instances on disc (i.e., 250,000 high-level context instances and 250,000 low-level contexts instances), the Context Storage reached a size of 4.06 GB. Despite the Context Manager proves to fairly handle this volume of data, the increasing time observed for I/O operations in long-term scenarios with several users demands for some of the aforementioned solutions.

4.4.3. Reliability of the Context Inference Method

An online evaluation of the HLCA is conducted to estimate the context recognition capabilities during realistic executions. Hence, this evaluation provides insights on the operation of the proposed context inference method during the regular use of the system. The HLCA is here tested

Figure 4.24: *Experimental setup. The smartwatch was generally placed by users on the right wrist, while the smartphone was kept in different locations based on the user's choice. The Kinect video device was only used for monitoring in the home scenario.*

altogether with four low-level context recognizers, which provide unstructured low-level context information, namely activities, locations and emotions, at runtime.

The current implementation of the HLCA (described in Section 4.3.5) has been deployed on the Microsoft Azure public cloud environment [108]. Conversely, to the study presented in Section 4.4.2, here, the cloud environment has been considered in order to support scalability and limitless computational power.

An inertial activity recognizer, a video activity recognizer, an audio emotion recognizer and a geopositioning location recognizer identify the low-level contexts. These low-level context recognizers process the data obtained by three sensing devices: a Samsung Galaxy S5 smartphone, an LG G Watch R smartwatch and a Kinect v2 video device (Figure 4.24). The inertial activity recognizer builds on the acceleration and rate of turn data collected by the smartphone and the smartwatch to identify the user's physical activity. The video activity recognizer operates on Kinect depth video data to recognize the user's body motion in a home scenario. Fusion is implemented for the unification of the activity recognition results in case the inertial activity recognizer and the video activity recognizer work simultaneously. The audio emotion recognizer utilizes the audio data recorded through the smartphone's microphone during phone call conversations in order to capture emotional states. Finally, the geopositioning location recognizer builds on the smartphone's GPS sensor information, longitude and latitude coordinates, to determine the person location. The low-level context recognizers are capable of identifying only a subset of the low-level context represented in the Mining Minds Context Ontology. Specifically, a set of quotidian actions involving some physical activities (*Eating*, *Running*, *Sitting*, *Standing*, *Walking*, *Stretching*, *Sweeping* and *Lying Down*), locations (*Home*, *Office*, *Restaurant*, *Gym* and *Mall*) and various emotions (*Anger*, *Happiness*, *Neutral* and *Sadness*). Details on the characteristics of the low-level context recognizers are found in [109].

During the evaluation, a total of five independent volunteers (i.e. S1-S5, see Table 4.3) were asked to perform a run-through of actions involving most of the low and high-level contexts of interest (Figure 4.25). Each action was carried out during approximately one minute, and some of the contexts were executed various times during the run-through. The labelling of the data was carried out by using a remote application, which was handled by the expert or observer. All of the data collection sessions were also video taped to check anomalous or unexpected patterns in the data and to correct labelling mistakes during the posterior curation of the data.

Table 4.3: *Characteristics of the participants involved in evaluation. The height is given in cm, while the weight is measured in kg.*

Subject	**Age**	**Gender**	**Height**	**Weight**
S1	31	Male	174	85
S2	25	Male	183	59
S3	29	Male	161	57
S4	27	Male	170	75
S5	30	Male	178	91

The contexts recognized during each run by the HLCA and the different low-level context recognizers are contrasted against the registered ground truth, here depicted in Figure 4.26. Actual and predicted contexts are aligned in time taking into account the delay associated with the processing of the data (on average, 78 ms for the activity recognition, 77 ms for the emotion recognition, 0.46 ms for the location recognition and approximately 2 s for the high-level context inference). Transitions among the contexts of interest are left out of the study since a null-class rejection schema has not been explicitly implemented [110].

Figure 4.25: *Examples of some of the actions which determine the contexts during the evaluation.*

The impact that misrecognitions at the low-level have on the high-level inference process can be observed in the evaluation results. It is clearly seen that a perfect identification of the high-level context is reached for those cases in which the recognized low-level contexts match the real ones. Examples of these cases are observed during the first hundred of seconds. Even in the event of low-level context misrecognitions, the recognition capabilities of the HLCA remain nearly unaltered. This is for example observed during the executions taking place from the second 900 onwards. Around that time, various mistakes are encountered at the emotion level, which nevertheless do not drive to incorrect conclusions at the high level. This robustness is attained thanks to the way the high-level contexts are defined, for example by giving more importance to the performed activity or location than the emotional state. More prominent errors are observed around the second 450 and in the range 700-820 approximately. These erroneous inferences are as a result of the incorrect activities recognized at a lower level. Then, it can be concluded that there is in general a relevant dependency in terms of the recognition capabilities for the high-level context on the low-level context.

Anyway, as already observed in the evaluation of the context ontology (Section 4.4.1), the impact of the errors in the low-level contexts is generally lower at the high-level context. This effect can be observed

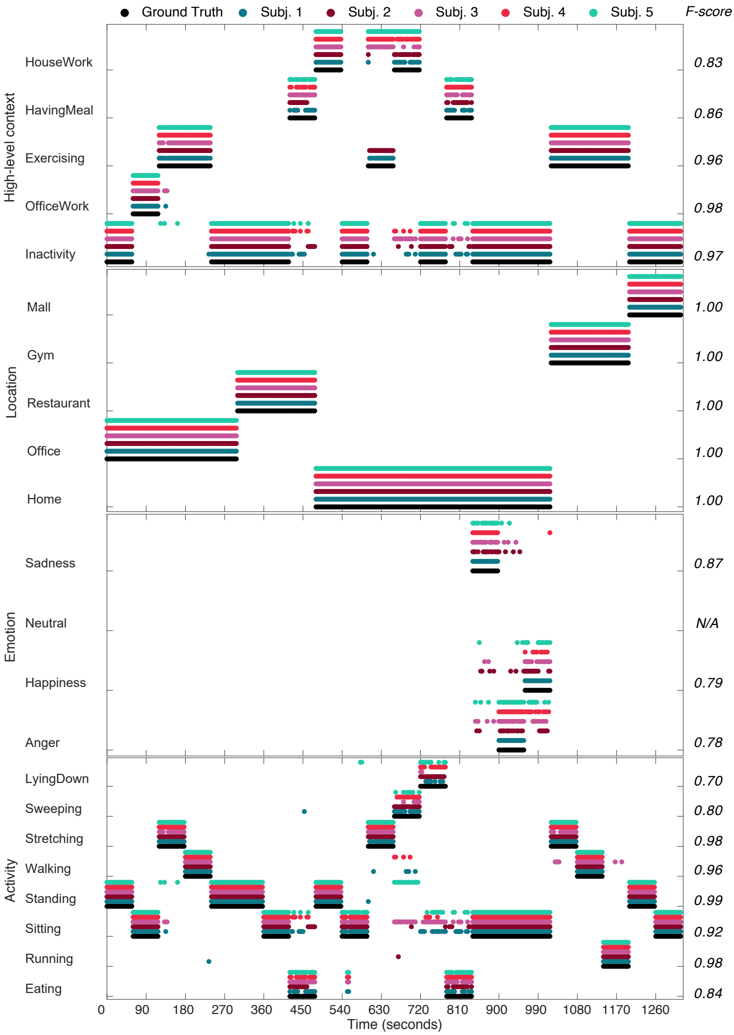


Figure 4.26: *Low- and high-level contexts recognized during online evaluation for the subjects S1-S5. Actual contexts are given by the ground-truth labels. Overall reliability for each context and across all subjects is given by the corresponding F-score.*

in the reliability of the recognition, which is measured via the F-score, a metric that 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 reliability in the inferred high-level contexts takes a value of F-score equal to 0.92, which is higher than the individual reliability in the recognition of the activities (F-score = 0.91) and the emotions (F-score = 0.61). The increase of reliability is due to the fact that not all the misrecognitions at low-level context lead to an inference error at high-level context.

5

Conclusion

5.1. Achievements

The goal of this thesis was to investigate on the application of ontology engineering and reasoning in order to solve some of the most prominent limitations of human behavior recognition systems working in realistic conditions. Therefore, ontologies, ontological reasoning and querying mechanisms were applied to comprehensively describe heterogeneous sensing resources and to dynamically select them in order to support the continuous operation of behavior recognition systems. Moreover, ontologies and ontological reasoning were also used to comprehensively describe human context information and to automatically infer meaningful and rich expressions of human context which could enhance the real-world operation of behavior-aware systems. In the following, the achievement of the four objectives defined to support the thesis goal are described.

Objective 1: Design and development of an ontology for the comprehensive and interoperable description of sensing technologies used in behavior recognition systems.

This work has presented MIMU-Wear, an OWL 2 ontology which provides syntactic interoperability and semantic compatibility in behavior recognition systems. The MIMU-Wear Ontology comprehensively describes wearable sensor platforms consisting of magnetic and inertial measurement units, including the MIMUs capabilities, such as their measurement properties, and the characteristics of wearable sensor platforms, including their on-body location and their survival properties. The MIMU-Wear Ontology provides implicit semantics enabling the automatic interpretation of the resource descriptions, their abstraction from the underlying technology, and the abstraction of the sensor selection method from the actual sensing infrastructure.

The MIMU-Wear Ontology builds on the standard W3C SSN Ontology and is designed in a modular manner with several plugable domain ontologies: the MIMU Ontology describing the characteristics of MIMUs, the MIMU Capabilities Ontology modeling the MIMUs sensing capabilities, the MIMU Magnitudes Ontology representing the different magnitudes observed by MIMUs, the MIMU Units Ontology representing the measurement units required to describe the MIMUs capabilities, the Wearable Sensor Platform Ontology modeling the characteristics of wearable sensor platforms, the Human Body Ontology modeling the human body parts representing the locations where wearable sensor platforms are worn, and the Wearable Survival Range Ontology modeling the survival conditions of wearable systems.

The modularity of MIMU-Wear 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 sensor 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 MIMU, 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, building on the standard W3C SSN Ontology facilitates the widespread adoption of MIMU-Wear 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, thus opening up a broad spectrum of ontologies in which MIMU-Wear could be integrated.

Objective 2: Definition and validation of a method based on ontology reasoning and querying to dynamically select sensing technologies to support continuity of behavior recognition.

This work has presented a novel method based on ontology reasoning and querying to dynamically select some of the available MIMUs, embedded into wearable platforms, whenever a MIMU part of the human behavior recognition system suffers some abnormality and needs a replacement. The proposed sensor selection method builds on the MIMU-Wear Ontology and applies ontological reasoning to infer the knowledge about the candidate sensor replacements from a set of heuristic rules. Therefore, several SWRL rules have been created to define the characteristics of candidate replacement MIMUs which could be used in the human behavior recognition system. The presented SWRL rules are exemplary and need to be particularized and prioritized for each specific application scenario. The method establishes that queries are iteratively posed on the ontological descriptions of the MIMUs in order to select the most appropriate MIMU for the replacement of the defective one. The iterative query 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 the result a candidate replacement sensor.

SPARQL queries have been presented to match the ontological descriptions of the available MIMUs with the required sensor characteristics.

The proposed ontology-based sensor selection method has been evaluated to prove that the replacement of an anomalous MIMU ensures the continuity of recognition, i.e., the reliability of the human behavior recognition system recovers with respect to the failure situation after the replacement takes place. Therefore, a realistic scenario in the area of wearable activity recognition has been modeled using the MIMU-Wear Ontology. The ontological descriptions of nine commercial wearable sensor platforms have been created and used to describe the real-world deployment scenario. For the validation of the sensor selection method, four SWRL rules have been defined and prioritized and four SPARQL queries have been created. The iterative query method has been applied for three different scenarios in which one of the three MIMUs active in the activity recognition process is simulated to fail. Using the sensor selection method, the best replacement sensor has been identified for each one of the scenarios. Then, the reliability of the activity recognition system has been calculated for the original configuration, for the configuration with the anomalous sensor, and for the different configurations in which the anomalous sensor is replaced. The reliability of the original system drops more than a third when one of the sensors behaves abnormally. Replacing the affected sensor with the sensor selected through the proposed ontology-based method has shown an improvement in all three cases. In the worst case, when the sensor to be replaced is noisy and little informative by nature, the improvement is subtle yet better in general than what is obtained when choosing a replacement in an arbitrary way. More significant improvements with respect to the failure situation are observed for different sensor combinations. In the best case, the recognition capabilities of the system are practically restored after replacing the anomalous sensor with the best option. Therefore, it has been proven that replacing potential anomalous sensors with the ones suggested by the ontology-based sensor selection method improves the reliability of the human behavior recognition system. Consequently, the proposed sensor selection method helps to support the continuity of operation required in real-world human behavior recognition.

This work has presented a method for selecting some of the available MIMUs, embedded into wearable platforms, whenever a MIMU part of the human behavior recognition system suffers some abnormality and needs a replacement. However, this is only one of the potential application scenarios for the MIMU-Wear 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 behavior recognition system 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. In all these scenarios, methods based on ontology reasoning and querying and similar to the one proposed in this work could be easily applied.

Objective 3: Design and development of an ontology for the exhaustive modeling of rich and meaningful expressions of context for human behavior analysis.

This work has presented the Mining Minds Context Ontology, an OWL 2 ontology for exhaustively modeling rich and meaningful expressions of context. This ontology is particularly devised to model the most commonplace contexts for health and wellness scenarios which involve sedentary and active lifestyles. Thus, the proposed ontology models multiple primitive components of context, such as the physical activity, the location and the emotion, as well as more abstract daily contexts which can be derived from the combination of these primitives, such as inactivity, exercising, office work or having meal.

The Mining Minds Context Ontology has been designed to support any combination of cross-domain behavior primitives (low-level contexts), in order to infer more abstract human context representations (high-level contexts). The unprecedented incorporation of emotions in the context definition enables the representation of new high-level contexts which can only be identified whenever a specific emotion takes place. The Mining Minds Context Ontology has also been designed to procure the identification of some high-level contexts even in the absence of emotion information. Therefore, the Mining Minds Context Ontology extends beyond the state-of-the-art while uniting emotion information as a novel behavioral component together with activity and location data to model more meaningful contextual information.

The Mining Minds Context Ontology provides the implicit semantics required for the derivation of new richer context information from basic existing context. Thus, this ontology enables the inference of meaningful human context information better describing human behavior for its analysis.

Objective 4: Definition and validation of a method based on ontology reasoning to automatically infer rich and meaningful human context to enhance the operation of behavior-aware systems.

This work has presented an ontology-based method for deriving high-level context information out of the combination of cross-domain low-level context primitives, namely activities, locations and emotions. This novel method builds on the Mining Minds Context Ontology and applies OWL 2 DL reasoning to infer high-level context from basic low-level context primitives. The proposed context inference method has been implemented by the High-Level Context Architecture, a system architecture devised to automatically infer rich and meaningful context in real-time. The High-Level Context Architecture consists of four main components: the High-Level Context Builder which generates ontological concepts representing the user context, the High-Level Context Reasoner which verifies and classifies the high-level context, the High-Level Context Notifier which makes the new high-level context available to third party applications, and the Context Manager which persists the context information.

The ontological reasoning-based method for the inference of rich context has been validated to prove that it can enhance the operation of behavior-aware systems in realistic conditions. Therefore, the robustness of the proposed context ontology and the performance and reliability of the novel context inference method have been analyzed. The evaluation of the Mining Minds Context Ontology proves its reasonably good robustness properties against potentially erroneous low-level contexts. In fact, the results have shown that the impact of the error introduced in the low-level context is always lower at the high-level. Moreover, it has been observed that the activity is the most prevalent category in terms of error impact, while the location and especially the emotion tend to show a lesser effect on the high-level context. The current prototype implementation of the High-Level Context Architecture has been proven to perform well with respect to processing time and management of context instances. However, in order to ensure scalability, the database transactions management needs to be improved. Finally, the evaluation of the reliability of the context inference method implemented by the High-Level Context Architecture during realistic executions has corroborated the results obtained in the offline evaluation. Even in the event of low-level contexts misrecognitions, the reliability of the high-level context inference remains nearly unaltered.

The presented High-Level Context Architecture constitutes the core engine for the inference of high-level behavioral information in the Mining Minds platform [102, 10]. Mining Minds is a novel digital health and wellness platform designed to seamlessly investigate and support people's lifestyles by intelligently mining human's daily living data generated through heterogeneous resources. Despite the proposed context inference method was originally devised to serve this platform, the High-Level Context Architecture has been defined in a way so it can be used independently for determining any high-level context information from diverse sources of low-level context data. In fact, in case the context inference method has to be applied to a new domain and some new contexts have to be identified, the High-Level Context Architecture would remain the same and only the context ontology would have to be updated. This is thanks to one of the main properties of the ontologies, they enable the decoupling of the knowledge from the code; thus, only requiring the adaptation of the ontology itself.

5.2. Contributions

In Section 5.1 it has been proved that the objectives of this thesis have been thoroughly fulfilled. Now, the main contributions of this thesis are listed:

1. MIMU-Wear: A modular OWL 2 ontology for the comprehensive and interoperable description of MIMUs in wearable platforms.

2. MIMU Ontology: An OWL 2 ontology describing the characteristics of MIMUs, such as their sensing capabilities.

3. Wearable Sensor Platform Ontology: An OWL 2 ontology modeling the characteristics of wearable sensor platforms, including their on-body location and their survival conditions.

4. Human Body Ontology: An OWL 2 ontology modeling the human body parts.

5. A method based on the MIMU-Wear ontology, SWRL rules and SPARQL queries to dynamically select sensing technologies to support continuity of behavior recognition.

6. Mining Minds Context Ontology: An OWL 2 ontology describing rich and meaningful expressions of human context. This on-

tology is publicly available at http://www.miningminds.re.kr/icl/context/context-v2.owl.

7. A method based on the Mining Minds Context Ontology and OWL 2 DL reasoning to automatically infer rich and meaningful human context to enhance the operation of behavior-aware systems.

8. HLCA: A system architecture providing a realization of the ontology-based context inference method, as well as its Java implementation.

5.3. Outlook

This thesis has proposed solutions to overcome some of the most prominent limitations of human behavior recognition systems working in realistic conditions; however, there is still much room for investigation on this topic. Some possible future directions to continue and extend this work are described next.

5.3.1. Smart-Clothing: Enabling Selection in Massive On-Body Sensor Networks

Smart-clothing is a new paradigm in sensing technologies where a vast amount of sensors are integrated into textiles to make them intelligent. Nowadays sensors are sewed to the cloth; however, in the near future they will be directly woven into the textile. The astonishing breakthroughs made so far in smart-clothes and the new advances to be seen in the upcoming years, point towards a scenario in which futuristic smart garments technology may become reality.

The potential of the presented MIMU-Wear Sensor Selection method could be especially leveraged in smart clothing scenarios with tons of sensors. Here, where smart garments are available for health and sports and where the priority is to obtain the required data and the choice of the particular sensors is not that important, the MIMU-Wear Sensor Selection method can provide and effective and rapid solution for the replacement of dying sensors as a result of the degradation of cloths. Thus, this work can have even a higher impact in the way vendors and manufacturers design and define these technologies.

In order to tackle the smart-clothing problem, MIMU-Wear should be extended, adding a new Clothes Ontology describing a variety of commonplace garments, which would be imported into the Wearable Sensor Platform Ontology. In this new scenario, the wearable sensor platform would be embedded into a piece of clothing, which would be the one worn by the user. This would involve an additional link in the ontology, yet the proposed sensor selection method would be perfectly applicable.

5.3.2. Uncertainty in Human Behavior Information

A mainstream topic in the artificial intelligence domain refers to the uncertainty or vagueness associated to most sources of information. This also applies to human behavior information, where primitives and contexts can be subject to different interpretations. In this work the activities, locations and emotions received from the lower levels of context inference were assumed to be 100% certain and accurate. However, in reality, uncertainty in context data tends to be the norm rather than the exception. For instance, there might exist vagueness in definitions of different emotions. A person could be characterized as happy to a 80% extent and neutral to a 20%. Similarly, sensory data could be uncertain and inaccurate with some probability. Accordingly, future work may explore the incorporation of uncertainty in the ontological definitions used for the inference of rich contextual information. Fuzzy representations seem to be the natural way to move into this direction. The intersection of uncertainty and ontologies has been little explored to date, which makes of this a particularly interesting and challenging problem for the future.

5.3.3. Interoperating Human Behavior Recognition Systems

The technological evolution experienced during the last years calls for more openness and shareability. In this thesis different mechanisms have been explored to open and share sensing resources through the use of universal descriptions and methods based on ontologies. The same principle could likewise apply to the outputs or information generated by human behavior recognition systems. In fact, behavior recognition models are normally devised to work for a particular application, thus limiting the use of the recognized behavioral information for other applications or purposes. However, realistic scenarios rather demand behavior recognition models that could be shared among applications.

Mechanisms for the selection of the most adequate recognizer for a given application could optimize the use of resources. Thus for example, depending on the target set of activities or emotions to be identified by the system, a specific model could be selected for operation. The use of models could also change depending on the sensing infrastructure available to the user at a given time, e.g. ambient sensors at home or wearables while being outside, or the kind of behaviors that can be expected at some place, e.g. sedentary activities at home or dynamic activities in the gym. These reasons make necessary the definition of resource descriptions which model heterogeneous resources and provide the interoperability required to dynamically select the best ones.

Another interesting scenario refers to the use of ontologies to procure fusion of behavior information provided by different systems. While this thesis has explored the combination of behavior information of diverse nature in order to generate more abstract contextual information, similar models and methods could be used to merge or fuse tags or labels of the same modality. For instance, ensemble models typically building on numbers, tags or predefined labels yielded by different emotion recognizers could work in an universal manner if these labels are univocally defined through ontological descriptions.

List of Figures

List of Tables

Bibliography

[1] W. David Pierce and Carl D. Cheney. *Behavior Analysis and Learning: Fifth Edition*. Taylor & Francis, 2013.

[2] Kristin A. Riekert, Judith K. Ockene, and Lori Pbert. *The Handbook of Health Behavior Change, 4th Edition*. Springer Publishing Company, 2013.

[3] Oresti Banos, Miguel Damas, Alberto Guillen, Luis-Javier Herrera, Hector Pomares, Ignacio Rojas, and Claudia Villalonga. Multi-sensor Fusion Based on Asymmetric Decision Weighting for Robust Activity Recognition. *Neural Processing Letters*, 42(1):5–26, 2015.

[4] Oresti Banos, Mate Attila Toth, Miguel Damas, Hector Pomares, and Ignacio Rojas. Dealing with the effects of sensor displacement in wearable activity recognition. *Sensors*, 14(6):9995–10023, 2014.

[5] Steffen Staab and Rudi Studer. *Handbook on Ontologies*. International Handbooks on Information Systems. Springer Publishing Company Inc., 2nd edition, 2009.

[6] 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/.

[7] Michael Compton, Payam Barnaghi, Luis Bermudez, Raul Garcia-Castro, Oscar Corcho, Simon Cox, John Graybeal, Manfred Hauswirth, Cory Henson, Arthur Herzog, Vincent Huang, Krzysztof Janowicz, W. David Kelsey, Danh Le Phuoc, Laurent Lefort, Myriam Leggieri, Holger Neuhaus, Andriy Nikolov, Kevin Page, Alexandre Passant, Amit Sheth, and Kerry Taylor. The SSN Ontology of the W3C Semantic Sensor Network Incubator Group. *Web Semantics: Science, Services and Agents on the World Wide Web*, 17(0), 2012.

[8] Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, and Mike Dean. *SWRL: A Semantic Web*

Rule Language Combining OWL and RuleML. W3C Member Submission, 21 May 2004. Available at http://www.w3.org/Submission/SWRL/.

[9] Steve Harris and Andy Seaborne. *SPARQL 1.1 (SPARQL Query Language for RDF)*. W3C Recommendation, 21 March 2013. Available at http://www.w3.org/TR/sparql11-query/.

[10] Oresti Banos, Muhammad Bilal Amin, Wajahat Ali Khan, Muhammad Afzal, Maqbool Hussain, Byeong Ho Kang, and Sungyong Lee. The Mining Minds digital health and wellness framework. *BioMedical Engineering OnLine*, 15(1):165–186, 2016.

[11] Fitbit Surge. Available online: https://www.fitbit.com/surge (Accessed: 2016-10-01).

[12] Jawbone® UP3™. Available online: https://jawbone.com/fitness-tracker/up3 (Accessed: 2016-10-01).

[13] Garmin vívofit®3. Available online: https://explore.garmin.com/en-US/vivo-fitness/ (Accessed: 2016-10-01).

[14] Empatica Embrace. Available online: https://www.empatica.com/product-embrace (Accessed: 2016-10-01).

[15] J.J. Oresko, Z. Jin, J. Cheng, S. Huang, Y. Sun, H. Duschl, and A. C. Cheng. A wearable smartphone-based platform for real-time cardiovascular disease detection via electrocardiogram processing. *IEEE Transactions on Information Technology in Biomedicine*, 14(3):734–740, May 2010.

[16] O. Banos, C. Villalonga, M. Damas, P. Gloesekoetter, H. Pomares, and I. Rojas. Physiodroid: Combining wearable health sensors and mobile devices for a ubiquitous, continuous, and personal monitoring. *The Scientific World Journal*, 2014(490824):1–11, 2014.

[17] S. Patel, C. Mancinelli, J. Healey, M. Moy, and P. Bonato. Using wearable sensors to monitor physical activities of patients with copd: A comparison of classifier performance. In *Proceedings of 6th International Workshop on Wearable and Implantable Body Sensor Networks*, pages 234–239, Washington, DC, USA, 2009.

[18] F. Aiolli, M. Ciman, M. Donini, and O. Gaggi. Climbtheworld: Real-time stairstep counting to increase physical activity. In *MobiQuitous 2014 - 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pages 218–227, 2014.

[19] Cassim Ladha, Nils Y Hammerla, Patrick Olivier, and Thomas Plötz. Climbax: skill assessment for climbing enthusiasts. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 235–244, 2013.

[20] Shian-Ru Ke, Hoang Le Uyen Thuc, Yong-Jin Lee, Jenq-Neng Hwang, Jang-Hee Yoo, and Kyoung-Ho Choi. A review on video-based human activity recognition. *Computers*, 2(2):88–131, 2013.

[21] Beat Fasel and Juergen Luettin. Automatic facial expression analysis: a survey. *Pattern recognition*, 36(1):259–275, 2003.

[22] Shashidhar G Koolagudi and K Sreenivasa Rao. Emotion recognition from speech: a review. *International journal of speech technology*, 15(2):99–117, 2012.

[23] Navid Fallah, Ilias Apostolopoulos, Kostas Bekris, and Eelke Folmer. Indoor human navigation systems: A survey. *Interacting with Computers*, 2013.

[24] Oscar D Lara and Miguel A Labrador. A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys & Tutorials*, 15(3):1192–1209, 2013.

[25] M. Ermes, J. Parkka, J. Mantyjarvi, and I. Korhonen. Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. *IEEE Trans. on Inform. Tech. Biomed.*, 12(1):20–26, 2008.

[26] A. Mannini, S. S. Intille, M. Rosenberger, A. M. Sabatini, and W. Haskell. Activity recognition using a single accelerometer placed at the wrist or ankle. *Medicine and Science in Sports and Exercise*, 45(11):2193–2203, 2013.

[27] Javier Hernandez, Yin Li, James M. Rehg, and Rosalind W. Picard. Cardiac and Respiratory Parameter Estimation Using Head-mounted Motion-sensitive Sensors. *EAI Endorsed Transactions on Pervasive Health and Technology*, 1(1):1–10, 2015.

[28] Javier Hernandez, Daniel McDuff, and Rosalind W. Picard. BioWatch: Estimation of Heart and Breathing Rates from Wrist Motions. In *Proceedings of the 9th International Conference on Pervasive Computing Technologies for Healthcare*, 2015.

[29] L. Liao, D. Fox, and H. Kautz. Extracting places and activities from gps traces using hierarchical conditional random fields. *The International Journal of Robotics Research*, 26(1):119–134, 2007.

[30] O. Banos, M. Damas, H. Pomares, A. Prieto, and I. Rojas. Daily living activity recognition based on statistical feature quality group selection. *Expert Systems with Applications*, 39(9):8013–8021, 2012.

[31] Yixiong Pan, Peipei Shen, and Liping Shen. Speech emotion recognition using support vector machine. *International Journal of Smart Home*, 6(2):101–108, 2012.

[32] Preeti Khanna and M Sasikumar. Rule based system for recognizing emotions using multimodal approach. *International Journal of Advanced Computer Science and Applications*, 4(7), 2013.

[33] I.-H. Bae. An ontology-based approach to adl recognition in smart homes. *Future Generation Computer Systems*, 33:32–41, 2014.

[34] D. Riboni and C. Bettini. OWL 2 modeling and reasoning with complex human activities. *Pervasive Mobile Computing*, 7(3):379–395, June 2011.

[35] P. Korpipaa, J. Mantyjarvi, J. Kela, H. Keranen, and E. J. Malm. Managing context information in mobile devices. *IEEE Pervasive Computing*, 2(3):42–51, July 2003.

[36] O. Banos, M. Damas, H. Pomares, and I. Rojas. On the use of sensor fusion to reduce the impact of rotational and additive noise in human activity recognition. *Sensors*, 12(6):8039–8054, 2012.

[37] P. Zappi, D. Roggen, E. Farella, G. Tröster, and L. Benini. Network-level power-performance trade-off in wearable activity recognition: A dynamic sensor selection approach. *ACM Transactions on Embedded Computer Systems*, 11(3):68:1–68:30, September 2012.

[38] D.-S. Zois, M. Levorato, and U. Mitra. Energy-efficient, heterogeneous sensor selection for physical activity detection in wireless body area networks. *Signal Processing, IEEE Transactions on*, 61(7):1581–1594, April 2013.

[39] Hoang Viet Nguyen, Ellen Munthe-Kaas, and Thomas Plagemann. Energy saving for activity recognition through sensor selection, scheduling and sampling rate tuning. In *Wireless and Mobile Networking Conference (WMNC), 2014 7th IFIP*, 2014.

[40] M. Janicke, B. Sick, P. Lukowicz, and D. Bannach. Self-adapting multi-sensor systems: A concept for self-improvement and self-healing techniques. In *Self-Adaptive and Self-Organizing Systems Workshops (SASOW), 2014 IEEE Eighth International Conference on*, pages 128–136, Sept 2014.

[41] Google Fit. Available online: https://www.google.com/fit/ (Accessed: 2016-10-01).

[42] Apple® HealthKit. Available online: https://developer.apple.com/healthkit/ (Accessed: 2016-10-01).

[43] H.K. Pung, Tao Gu, Wenwei Xue, P.P. Palmes, J. Zhu, Wen Long Ng, Chee Weng Tang, and Nguyen Hoang Chung. Context-aware middleware for pervasive elderly homecare. *Selected Areas in Communications, IEEE Journal on*, 27(4):510–524, May 2009.

[44] J. Xu, Y. Wang, M. Barrett, B. Dobkin, G. Pottie, and W. Kaiser. Personalized, multi-layer daily life profiling through context enabled activity classification and motion reconstruction: An integrated systems approach. *IEEE Journal of Biomedical and Health Informatics*, PP(99):1–1, 2015.

[45] A. Gaggioli, G. Pioggia, G. Tartarisco, G. Baldus, D. Corda, P. Cipresso, and G. Riva. A mobile data collection platform for mental health research. *Personal Ubiquitous Comput.*, 17(2):241–251, 2013.

[46] Thomas Strang and Claudia Linnhoff-Popien. A context modeling survey. In *Workshop on Advanced Context Modelling, Reasoning and Management - The Sixth International Conference on Ubiquitous Computing*, 2004.

[47] Mike Botts, George Percivall, Carl Reed, and John Davidson. OGC Sensor Web Enablement: Overview And High Level Architecture. Technical report, OGC, 2007.

[48] Mike Botts and Alexandre Robin. OpenGIS Sensor Model Language (SensorML), Implementation Specification (OGC 07-000). OpenGIS Implementation Standard, 2007.

[49] Simon Cox. Observations and Measurements Part 1 - Observation schema (OGC 07-022r1). OpenGIS Implementation Standard, 2007.

[50] Simon Cox. Observations and Measurements Part 2 - Sampling features (OGC 07-002r3). OpenGIS Implementation Standard, 2007.

[51] Amit Sheth, Cory Henson, and Satya S. Sahoo. Semantic Sensor Web. *IEEE Internet Computing*, 12(4):78–83, 2008.

[52] D.J. Russomanno, Kothari C., and Thomas O. Building a sensor ontology: A practical approach leveraging iso and ogc models. In *The 2005 International Conference on Artificial Intelligence*, pages 637–643, Las Vegas, NV, 2005.

[53] I. Niles and A. Pease. Origins of the standard upper merged ontology: A proposal for the ieee standard upper ontology. In *In Working Notes of the IJCAI- 2001 Workshop on the IEEE Standard Upper Ontology*, Seattle, WA, 2001.

[54] Siniša Nikolić, Valentin Penca, Milan Segedinac, and Zora Konjović. Semantic web based architecture for managing hardware heterogeneity in wireless sensor network. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*, WIMS '11, pages 42:1–42:9, New York, NY, USA, 2011.

[55] J. Soldatos, N. Kefalakis, M. Hauswirth, M. Serrano, J.-P. Calbimonte, M. Riahi, K. Aberer, P.P. Jayaraman, A. Zaslavsky, I.P. Å¡arko, L. Skorin-Kapov, and R. Herzog. Openiot: Open source internet-of-things in the cloud. In *FP7 OpenIoT Project Workshop 2014, LNCS 9001*, volume 9001, pages 13–25, 2015. cited By 0.

[56] N. Kefalakis, J. Soldatos, A. Anagnostopoulos, and P. Dimitropoulos. A visual paradigm for iot solutions development. In *FP7 OpenIoT Project Workshop 2014, LNCS 9001*, volume 9001, pages 26–45, 2015.

[57] P. Desai, A. Sheth, and P. Anantharam. Semantic gateway as a service architecture for iot interoperability. In *2015 IEEE 3rd International Conference on Mobile Services, MS 2015*, pages 313–319, 2015.

[58] A. Ibrahim, F. Carrez, and K. Moessner. Geospatial ontology-based mission assignment in wireless sensor networks. In *2015 International Conference on Recent Advances in Internet of Things, RIoT 2015*, 2015.

[59] Rimel Bendadouche, Catherine Roussey, Gil De Sousa, Jean-Pierre Chanet, and Kun Mean 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*, SSN'12, pages 49–64, Aachen, Germany, Germany, 2012.

[60] L. Nachabe, M. Girod-Genet, and B. El Hassan. Unified data model for wireless sensor network. *Sensors Journal, IEEE*, 15(7):3657–3667, 2015.

[61] L. Nachabe, M. Girod-Genet, B. El Hassan, J. Khawaja, and H. Salloum. Semantic smart home system: Ontosmart to monitor and assist habitant. *International journal of computers and communications*, 10:78–86, 2016.

[62] 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.

[63] X. Wang, H. An, Y. Xu, and S. Wang. Sensing network element ontology description model for internet of things. In *International Conference on Information Science and Control Engineering*, 2015.

[64] Vlasios Tsiatsis, Alexander Gluhak, Tim Bauge, Frederic Montagut, Jesus Bernat, Martin Bauer, Claudia Villalonga, Payam M

Barnaghi, and Srdjan Krco. The sensei real world internet architecture. In *Future Internet Assembly*, pages 247–256, 2010.

[65] Claudia Villalonga, Martin Bauer, Fernando López Aguilar, Vincent A Huang, and Martin Strohbach. A resource model for the real world internet. In *5th European Conference on Smart Sensing and Context, EuroSSC 2010*, pages 163–176. Springer, 2010.

[66] Y.-C. Hsu, C.-H. Lin, and W.-T. Chen. Design of a Sensing Service Architecture for Internet of Things with Semantic Sensor Selection. In *2014 IEEE International Conference on Ubiquitous Intelligence and Computing, 2014 IEEE International Conference on Autonomic and Trusted Computing, 2014 IEEE International Conference on Scalable Computing and Communications and Associated Symposia/Workshops, UIC-ATC-ScalCom 2014*, 2014.

[67] C. Perera, A. Zaslavsky, C.H. Liu, M. Compton, P. Christen, and D. Georgakopoulos. Sensor search techniques for sensing as a service architecture for the internet of things. *Sensors Journal, IEEE*, 14(2):406–420, Feb 2014.

[68] M. Baldauf, S. Dustdar, and F. Rosenberg. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2(4):263–277, 2007.

[69] Claudio Bettini, Oliver Brdiczka, Karen Henricksen, Jadwiga Indulska, Daniela Nicklas, Anand Ranganathan, and Daniele Riboni. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, 6(2):161 – 180, 2010.

[70] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos. Context aware computing for the internet of things: A survey. *IEEE Communications Surveys Tutorials*, 16(1):414–454, 2014.

[71] Harry Chen, Tim Finin, and Anupam Joshi. The SOUPA ontology for pervasive computing. In V. Tamma, S. Cranefield, T.W. Finin, and S. Willmott, editors, *Ontologies for Agents: Theory and Experiences*, pages 233–258. BirkHauser, 2005.

[72] X. H. Wang, D. Q. Zhang, T. Gu, and H. K. Pung. Ontology based context modeling and reasoning using owl. In *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, pages 18–22. Ieee, 2004.

[73] H. Chen, T. Finin, and A. Joshi. An ontology for context-aware pervasive computing environments. *The Knowledge Engineering Review*, 18(03):197–207, 2003.

[74] Tao Gu, Hung Keng Pung, and Da Qing Zhang. A middleware for building context-aware mobile services. In *In Proceedings of IEEE Vehicular Technology Conference (VTC*, 2004.

[75] R. Hervás, J. Bravo, and J. Fontecha. A context model based on ontological languages: a proposal for information visualization. *J. UCS*, 16(12):1539–1555, 2010.

[76] Maria Poveda Villalon, Mari Carmen Suárez-Figueroa, Raúl García-Castro, and Asunción Gómez-Pérez. A context ontology for mobile environments. In *Proceedings of Workshop on Context, Information and Ontologies*. CEUR-WS, 2010.

[77] Alessandra Agostini, Claudio Bettini, and Daniele Riboni. Hybrid reasoning in the care middleware for context awareness. *Int. J. Web Eng. Technol.*, 5(1):3–23, 2009.

[78] D. Riboni and C. Bettini. COSAR: hybrid reasoning for context-aware activity recognition. *Personal Ubiquitous Computing*, 15(3):271–289, March 2011.

[79] C. Liming, C. Nugent, and G. Okeyo. An ontology-based hybrid approach to activity modeling for smart homes. *IEEE T. Human-Machine Systems*, 44(1):92–105, 2014.

[80] George Okeyo, Liming Chen, Hui Wang, and Roy Sterritt. Dynamic sensor data segmentation for real-time knowledge-driven activity recognition. *Pervasive and Mobile Computing*, 10:155–172, 2014.

[81] Rim Helaoui, Daniele Riboni, and Heiner Stuckenschmidt. A probabilistic ontological framework for the recognition of multilevel human activities. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '13, pages 345–354, New York, NY, USA, 2013.

[82] Natalia Diaz Rodriguez, Manuel P. Cuellar, Johan Lilius, and Miguel Delgado Calvo-Flores. A fuzzy ontology for semantic modelling and recognition of human behaviour. *Knowledge-Based Systems*, 66:46 – 60, 2014.

[83] protégé. Available online: http://protege.stanford.edu/ (Accessed: 2016-10-01).

[84] David Peterson, Shudi Gao, Ashok Malhotra, C. M. Sperberg-McQueen, and Henry S. Thompson. *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/.

[85] Foundational Model of Anatomy ontology (FMA). Available online: http://sig.biostr.washington.edu/projects/fm/AboutFM.html (Accessed: 2016-10-01). http://sig.biostr.washington.edu/projects/fm/AboutFM.html.

[86] Christopher Mungall, Carlo Torniai, Georgios Gkoutos, Suzanna Lewis, and Melissa Haendel. Uberon, an integrative multi-species anatomy ontology. *Genome Biology*, 13(1):R5, 2012.

[87] Linked data. Available online: http://linkeddata.org/ (Accessed: 2016-10-01).

[88] O. Banos, A. Calatroni, M. Damas, H. Pomares, I. Rojas, H. Sagha, J. del R Millan, G. Troster, R. Chavarriaga, and D. Roggen. Kinect=IMU? Learning MIMO Signal Mappings to Automatically Translate Activity Recognition Systems across Sensor Modalities. In *Wearable Computers (ISWC), 2012 16th International Symposium on*, pages 92–99, 2012.

[89] Hesam Sagha, Hamidreza Bayati, José del R. Millán, and Ricardo Chavarriaga. On-line anomaly detection and resilience in classifier ensembles. *Pattern Recognition Letters*, 34(15):1916–1927, 2013.

[90] Hesam Sagha, Alberto Calatroni, José del R. Millán, Daniel Roggen, Gerhard Tröster, and Ricardo Chavarriaga. Robust Activity Recognition Combining Anomaly Detection and Classifier Retraining. In *Proceedings of the 10th annual conference on Body Sensor Networks (BSN)*, 2013.

[91] Xsens Technologies B.V. *XM-B Technical Documentation*, May 2009. Available at: http://www.xsens.com (Accessed: 2016-10-01).

[92] O. Lara and M. Labrador. A survey on human activity recognition using wearable sensors. *Communications Surveys Tutorials, IEEE*, PP(99):1–18, 2012.

[93] Juha Parkka, Miikka Ermes, Panu Korpipaa, Jani Mantyjarvi, Johannes Peltola, and Ilkka Korhonen. Activity classification using realistic data from wearable sensors. *IEEE Transactions on Information Technology in Biomedicine*, 10(1):119–128, 2006.

[94] Kerem Altun and Billur Barshan. Human activity recognition using inertial/magnetic sensor units. In *Human Behavior Understanding*, pages 38–51, 2010.

[95] Hristijan Gjoreski and Matjaž Gams. Accelerometer data preparation for activity recognition. In *International Multiconference Information Society*, 2011.

[96] O. Banos, C. Villalonga, R. Garcia, A. Saez, M. Damas, J. A. Holgado, S. Lee, H. Pomares, and I. Rojas. Design, implementation and validation of a novel open framework for agile development of mobile health applications. *Biomedical Engineering Online*, 14(S2:S6):1–20, 2015.

[97] Oresti Banos, Miguel Damas, Héctor Pomares, Ignacio Rojas, Máté Attila Toth, and Oliver Amft. A benchmark dataset to evaluate sensor displacement in activity recognition. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, pages 1026–1035, New York, NY, USA, 2012.

[98] Oresti Banos, Juan-Manuel Galvez, Miguel Damas, Hector Pomares, and Ignacio Rojas. Window size impact in human activity recognition. *Sensors*, 14(4):6474–6499, 2014.

[99] J. R. Kwapisz, G. M. Weiss, and S. A. Moore. Activity recognition using cell phone accelerometers. *17th Conference on Knowledge Discovery and Data Mining*, 12(2):74–82, 2011.

[100] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification.* John Wiley & Sons, New York, USA, 2000.

[101] M. Sokolova and G. Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, 45(4):427 – 437, 2009.

[102] O. Banos, M. Bilal-Amin, W. Ali-Khan, M. Afzel, T. Ali, B.-H. Kang, and S. Lee. The mining minds platform: a novel person-centered digital health and wellness framework. In *Proceedings of the 9th International Conference on Pervasive Computing Technologies for Healthcare*, 2015.

[103] Anind K. Dey. Understanding and using context. *Personal and Ubiquitous Computing*, 5(1):4–7, February 2001.

[104] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical owl-dl reasoner. *Journal of Web Semantics*, 5(2), 2007.

[105] Apache Jena. Available online: https://jena.apache.org/ (Accessed: 2016-10-01).

[106] D. Brickley and R. V. Guha. *RDF Schema 1.1.* W3C Recommendation, 25 February 2014. Available at https://www.w3.org/TR/rdf-schema/.

[107] Leonard Richardson and Sam Ruby. *RESTful web services.* O'Reilly Media, Inc., 2008.

[108] Microsoft Azure. Available online: http://azure.microsoft.com (accessed on 30 Nov 2015).

[109] O. Banos, C. Villalonga, J. H. Bang, T. H. Hur, D. Kang, S.-B. Park., T. Hyunh-The, L. B. Vui, M.-B. Amin, M.-A. Razzaq, W. Ali Khan, C.-S. Hong, and S. Lee. Human behavior analysis by means of multimodal context mining. *Sensors*, 16(8):1–19, 2016.

[110] Andreas Bulling, Ulf Blanke, and Bernt Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys (CSUR)*, 46(3):33, 2014.

Curriculum Vitae

Personal Information

Claudia Villalonga
Born July 10, 1982, Maó, Menorca, Spain
Citizen of Spain

Education

2014 - 2016	PhD in Information and Communication Technologies of the University of Granada, Spain
2014 - 2016	Master in Business Management (Administration of Organizations in the Knowledge Economy) at Universitat Oberta de Catalunya, Spain
2000 - 2006	Master's Degree in Telecommunications Engineering at Escola Tècnica Superior de Telecomunicacions de Barcelona, Universitat Politècnica de Catalunya, Spain

Work Experience

2015	Researcher of the Industry Academic Cooperation Foundation at Ubiquitous Computing Lab., Department of Computer Engineering, Kyung Hee University, South Korea
2012 - 2014	R&D Project Manager at Informática Gesfor, CGI Group Inc., Madrid, Spain
2008 - 2011	Research Associate at Wearable Computing Lab., Department of Information Technology and Electrical Engineering, ETH Zürich, Switzerland
2008 - 2011	Research Associate at SAP (Schweiz) AG, SAP Research Center Zürich, Switzerland
2006 - 2008	Research Associate at NEC Europe Ltd., NEC Laboratories Europe, Research Division, Heidelberg, Germany

Publications List

International Journals (SCI-indexed)

Villalonga, C., Pomares, H., Rojas, I., Banos, O. **MIMU-Wear: Ontology-based Sensor Selection for Real-World Wearable Activity Recognition**. *Neurocomputing* (2016) [Accepted].

Villalonga, C., Razzaq, M.-A., Ali Khan, W., Pomares, H., Rojas, I., Lee, S., Banos, O. **Ontology-based High-Level Context Inference for Human Behavior Identification**. *Sensors*, vol. 16, no. 10, pp. 1-26 (2016).

Banos, O., Villalonga, C., Bang, J. H., Hur, T. H., Kang, D., Park., S.-B., Hyunh-The, T., Vui, L. B., Amin, M.-B., Razzaq, M.-A., Ali Khan, W., Hong, C.-S., Lee, S. **Human Behavior Analysis by means of Multimodal Context Mining**. *Sensors*, vol. 16, no. 8, pp. 1-19 (2016).

Banos, O., Villalonga, C., Garcia, R., Saez, A., Damas, M., Holgado, J. A., Lee, S., Pomares, H., Rojas, I. **Design, implementation and validation of a novel open framework for agile development of mobile health applications**. *BioMedical Engineering OnLine*, vol. 14, no. S2:S6, pp. 1-20 (2015).

Banos, O., Moral-Munoz, J.A., Diaz-Reyes, I., Arroyo-Morales, M., Damas, M., Herrera-Viedma, E., Hong, C.S., Lee, S., Pomares, H., Rojas, I., Villalonga, C. **mDurance: a Novel Mobile Health System to Support Trunk Endurance Assessment**. *Sensors*, vol. 15, no. 6, pp. 13159-13183 (2015).

Banos, O., Damas, M., Guillen, A., Herrera, L.J., Pomares, H., Rojas, I., Villalonga C. **Multi-sensor fusion based on asymmetric decision weighting for robust activity recognition**. *Neural Processing Letters*, vol. 42, no. 1, pp. 5-26 (2015).

Banos, O., Villalonga, C., Damas, M., Glösekötter, P., Pomares, H., Rojas, I. **PhysioDroid: combining wearable health sensors and smartphones for a ubiquitous, continuous and personal monitoring**. *The Scientific World Journal*, vol. 2014, no. 490824, pp. 1-11 (2014).

Eurich, M., Villalonga, C., Boutellier, R. **Dynamic Formation of Business Networks: A Framework for 'Quality of Information'-based Discovery of Resources**. *International Journal of Applied Logistics (IJAL)*, vol. 2:4, pp. 44-60 (2011).

Presser, M., Barnaghi, P., Eurich, M., Villalonga, C. **The SENSEI Project: Integrating the Physical World with the Digital World of the Network of the Future**. *IEEE Communications Magazine*, vol. 47:4 pp.1-4 (2009)

Book Chapters

Tsiatsis, V., Gluhak, A., Bauge, T., Montagut, F., Bernat, J., Bauer, M., Villalonga, C., Barnaghi, P., Krco, S. **The SENSEI Real World Internet Architecture**. *Towards the Future Internet*, IOS Press, pp. 247-256 (2010)

Battestini, A., Flanagan, A., Floréen, P., Gessler, S., Koolwaaij, J., Lagerspetz, E., Lau, S., Luther, M., Martin, M., Millerat, J., Mrohs, B., Nurmi, P., Paolucci, M., Robinson, J., Suomela, J., Villalonga, C., Wagner, M. **Context Awareness and Management**. *Enabling Technologies for Mobile Services: The MobiLife Book*, John Wiley and Sons, pp. 99-152 (2007)

International Conferences

Villalonga, C., Pomares, H., Banos, O. **Analysis of the Innovation Outputs in mHealth for Patient Monitoring**. *International Conference on Ubiquitous Computing and Communications (IUCC 2016)*, Granada, Spain, December 14-16, (2016)

Villalonga, C., Banos, O., Pomares, H., Rojas, I., Lee, S. **High-Level Context Architecture for Real Time Identication of Human Behavior**. *International Work-Conference on Bioinformatics and Biomedical Engineering (IWBBIO 2016)*, Granada, Spain, April 20-22, (2016).

Banos, O., Villalonga, C., Bang, J. H., Hur, T. H., Kang, D. W., Park, S. B., Hyunh-The, T., Vui, L. B., Lee, S. **Inferring Human Behavior by means of Multimodal Context Mining**. *International Work-Conference on Bioinformatics and Biomedical Engineering (IWBBIO 2016)*, Granada, Spain, April 20-22, (2016).

Villalonga, C., Banos, O., Ali Khan, W., Ali, T., Razzaq, M. A., Lee, S., Pomares, H., Rojas, I. **High-Level Context Inference for Human Behavior Identification**. *International Work-conference on Ambient Assisted Living an Active Ageing (IWAAL 2015)*, Puerto Varas, Chile, December 1-4, (2015).

Banos, O., Damas, M., Guillen, A., Herrera, L. J., Pomares, H., Rojas, I., Villalonga, C., Lee, S. **On the Development of A Real-Time Multi-Sensor Activity Recognition System**. *International Work-conference on Ambient Assisted Living an Active Ageing (IWAAL 2015)*, Puerto Varas, Chile, December 1-4, (2015).

Banos, O., Moral-Munoz, J.A., Diaz-Reyes, N., Arroyo-Morales, M., Damas, M., Pomares, H., Rojas, I., Villalonga, C., Bang, J.H., Kang, D.U., Hong, C. S., Lee, S. **Facilitating Trunk Endurance Assessment by means of Mobile Health Technologies**. *ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2015)*, Osaka, Japan, September 7-11, (2015).

Banos, O., Bang, J. H., Hur, T. H., Siddiqui, M., Hyunh-The, T., Vui, L. B., Ali Khan, W., Ali, T., Villalonga, C., Lee, S. **Mining Human Behavior for Health Promotion**. *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2015)*, Milano, Italy, August 25-29, (2015).

Villalonga, C., Banos, O., Pomares, H., Rojas, I. **Ontological Sensor Selection for Wearable Activity Recognition**. *International Work Conference on Artificial Neural Networks (IWANN 2015)*, Palma de Mallorca, Spain, June 10-12, (2015).

Banos, O., Galvez, J. M., Damas, M., Guillen, A., Herrera, L. J., Pomares, H., Rojas, I., Villalonga, C., Hong, C. S., Lee, S. **Multiwindow Fusion for Wearable Activity Recognition**. *International Work Conference on Artificial Neural Networks (IWANN 2015)*, Palma de Mallorca, Spain, June 10-12, (2015).

Villalonga, C., Banos, O., Pomares, H., Rojas, I. **An Ontology for Dynamic Sensor Selection in Wearable Activity Recognition**. *International Work-Conference on Bioinformatics and Biomedical Engineering (IWBBIO 2015)*, Granada, Spain, April 15-17, (2015).

Banos, O., Garcia, R., Holgado, J. A., Damas, M., Pomares, H., Rojas, I., Saez, A., Villalonga, C. **mHealthDroid: a novel framework for agile development of mobile health applications**. *Sixth International Work-conference on Ambient Assisted Living an Active Ageing (IWAAL 2014)*, Belfast, Northern Ireland, December 2-5, (2014).

Illera, R., Villalonga, C. **Interactive applications to personalize TV's**. *Tenth European conference on Interactive TV and video (EuroiTV 2012)*, Berlin, Germany, July 4-6, (2012)

Villalonga, C., Roggen, D., Tröster, G. **Shaping Sensor Node Ensembles according to their Recognition Performance within a Planning-based Context Framework**. *Eight International Conference on Networked Sensing Systems (INSS 2011)*, Penghu, Taiwan, June 12-15, (2011)

Neff, S., Villalonga, C., Roggen, D., Tröster, G. **Do I look like my neighbors? A Method to Detect Sensor Data Similarity for open-ended Activity Recognition Systems**. *Eight International Conference on Networked Sensing Systems (INSS 2011)*, Penghu, Taiwan, June 12-15, (2011)

Villalonga, C., Bauer, M., López, F., Huang, V., Strohbach, M. **A Resource Model for the Real World Internet**. *Fifth European conference on Smart sensing and context (EuroSSC 2010)*, Passau, Germany, November 14-16, (2010)

Manzoor, A., Villalonga, C., Calatroni, A., Truong, H.-L., Roggen, D., Dustdar, S., Tröster, G. **Identifying Important Action Primitives for High Level Activity Recognition**. *Fifth European conference on Smart sensing and context (EuroSSC 2010)*, Passau, Germany, November 14-16, (2010)

Villalonga, C., Bauer, M., Huang, V., Bernat, J., Barnaghi, P. **Modeling of Sensor Data and Context for the Real World Internet**. *Eighth IEEE International Conference on Pervasive Computing and Communications (PerCom 2010)*, Mannheim, Germany, March 29 - April 2, (2010)

Calatroni, A., Villalonga, C., Roggen, D., Tröster, G. **Context Cells: Towards Lifelong Learning in Activity Recognition Systems**. *Forth European Conference on Smart Sensing and Context (EuroSSC 2009)*, Guildford, UK, September 16-18, (2009) Villalonga, C., Roggen, D., Lombriser, C., Zappi, P., Tröster, G. **Bringing Quality of Context into Wearable Human Activity Recognition Systems**. *First International Workshop on Quality of Context (QuaCon 2009)*, Stuttgart, Germany, June 25-26, (2009)

Strohbach, M., Bauer, M., Kovacs, E., Villalonga, C., Richter, N. **Context sessions: a novel approach for scalable context management in NGN networks**. *Eighth International Middleware Conference (Middleware '07)*, Port Beach, California, US, November 26 2007

Villalonga, C., Strohbach, M., Snoeck, N., Sutterer, M., Belaunde, M., Kovacs, E., Zhdanova, A.V., Goix, L.W., Droegehorn, O. **Mobile Ontology: Towards a Standardized Semantic Model for the Mobile Domain**. *Fifth International Conference on Service-Oriented Computing (ICSOC 2007)*, Vienna, Austria, September 17-20, (2007)

Kernchen, R., Boussard, M., Hesselman, C., Villalonga, C., Clavier, E., Zhdanova, A.V., Cesar, P. **Managing Personal Communication Environments in Next Generation Service Platforms**. *16th IST Mobile and Wireless Communications Summit*, Budapest, Hungary, July 1-5, (2007)