

## Context-Aware Multi-layered Ontology for Composite Situation Model in Pervasive Computing



Abderrahim Lakehal<sup>1\*</sup>, Adel Alti<sup>1,2</sup>, Philippe Roose<sup>3</sup>

<sup>1</sup> LRSD Laboratory, Computer Science Department, Sciences Faculty, Ferhat Abbas University SETIF-1, Sétif 19000, Algeria

<sup>2</sup> Department of Management Information Systems, College of Business & Economics Qassim University, P.O. Box 6633, Buraidah 51452, KSA

<sup>3</sup> LIUPPA, University of PAU, Anglet 64000, France

Corresponding Author Email: [a.alti@qu.edu.sa](mailto:a.alti@qu.edu.sa)

<https://doi.org/10.18280/isi.250501>

### ABSTRACT

**Received:** 16 July 2020

**Accepted:** 25 September 2020

#### Keywords:

*ontology, heterogeneous connected objects, smart domains, situations, multi-OCSM*

With the rapid advancement of technologies and analysis tools in the smart systems, enabling real-time context monitoring of user's living conditions and quality services delivery is increasing. Current studies in this area are focused on developing mobile applications with specific services, based on toolkit that allow developers to obtain context information from sensors. However, there exists a notable lack of ontology able to represent all the necessary context information starting from distributed users, and constantly changing environment. The modeling of user's domains to represent diverse mobile and IoT devices, and finalizing with the description of user's composite situations in smart\*(health, home, cities, car, office, etc.) domains. Considering interoperability, reusability, and flexibility, a new context composite situation ontology for smart systems is proposed with better representation of heterogeneous context. The ontology enables to sense, reason, and infer composite situations in various smart domains, prioritizes critical situations and facilitates the delivery of smart mobile service. Proposed ontology is formalized and validated on different smart environments with different user's situations. Several experiments were carried out with a real-life motivating scenario. Experimental results showed that the proposed approach has reduced queries times and improved flexibility.

## 1. INTRODUCTION

In this pervasive era as ubiquitous access to digital content is available every time through IoT (Internet of Things). These IoT smart objects are multiplied and distributed by different providers. They leverage heterogeneous smart objects with various communication protocols (i.e. ZigBee, 3G, Wi-Fi, etc.) and a variety of context data types (i.e. text, audio, image, video, etc.). The busyness of one of these smart objects may cause losses of context data, which can lead to discontinuity capturing the context information in several smart environments. For instance, the healthcare domain that consists of several emergencies may need continuous health monitoring and need to be processed quickly. In critical situations, such as losing the connection from the heart beating sensor implanted in the patient, it may cause the loss of health data and this could become fatal. Therefore, in such cases, urgent situations must be identified efficiently and the continuity of service must be ensured in order to guarantee agreeable fit. Identifying urgent situations and ensuring continuity of service in pervasive systems is a crucial problem [1, 2]. In order to identify the user situations in pervasive computing environments there are various approaches available such as fuzzy logic, machine learning [3, 4]. Ontology-based approach has been widely used for modeling large and heterogeneous physical concepts (i.e. smart object, device) with high abstract concepts (i.e. context, situation, service). To ensure precise representation of various user's

needs under several moving locations in multi smart domains, temporary or permanent situations rules are semantically specified into an ontology model within the context domain. Later situation is identified to trigger the appropriate services associated with this situation [5]. Moreover, semantic is essential in order to identify situations of users within a given smart domain regardless of situation's categories or context sources.

Several ontologies models have been proposed in pervasive computing. However, context-aware ontologies have been used current user's needs and their situations in flexible way [2]. Some studies focus on developing context-aware ontology within a specific smart domain such as health domain and smart home [6]. Others IoT-based smart domain provides generic ontology and a set of design tools to help designer to represent and handle contextual information of users and relatives entities to monitor and identify user's situation. Existing ontologies offer appropriate capabilities, including the semantic description of context and user's needs. However, some ontologies are mostly dedicated to a specific smart context domain. For instance, health-specific ontology represents patient's body parameters, health diseases, and smart objects related to health domain. So, it is difficult to cover all context information in multiple smart domains (home, office, transport, health, etc.). Also, existing ontologies do not deal with composite situations (parallel, sequential, recurrence, and alternative), nor do they handle multiple devices and smart objects at the run-time. In real-life,

situations at semantic level have different priorities ordering from low priority, normal priority, and high priority where high priority situations are handled before the situations having low priority. Existing ontologies neglect the situation priority concept and service durability under several user's locations (moving through different smart domains) during the run time. Self-specification of user's needs in the ontology is based on user-defined situations rules. Therefore, we need to design an ontology model to manage user-centric situations and provides new concepts such as composite situations in pervasive computing systems including composition operators (parallel, sequence, recurrence and alternative).

In this research work, we introduce a new ontology model entitled MULTI-OCSM (MULTI-layered Ontology-based Composite Situation Model) different from the ontologies proposed in the literature. This ontology aims to model smart objects, smart domains, context, simple and composite situations, and application services. In fact, real-life situations are more complex: parallel, sequence, interlacing, etc. This ontology has new modeling of composite situations pertaining to smart environments using specific combination operators: parallel, sequence, recurrence and alternative. The combination of these operators allows creating similar real-life complex situations. Although there are plenty of researches in the topic of pervasive computing, our research, unlike most ontologies, added some key contributions in the field that are:

- The proposed ontology relies on multi-layered context-aware semantic services, which provides event interpretation and more flexibility under several moving locations in several smart environments
- The proposed unified ontology merges common concepts about smart domains and supports composite situations using combination operators that ensures a high-level of abstraction in pervasive computing.
- The proposed ontology provides priority-based action services related to the user in case of emergency.

This research work is organized as follows. The existing works are discussed in Section 2. The multi-smart domains context properties and user situations analysis is discussed in Section 3. The proposed context-aware ontology is detailed in Section 4. Section 5 validates the context composite situation ontology using a smart home case study. Finally, section 6 concludes this paper and presents future researches.

## 2. RELATED WORKS

Many initiatives have been taken in ontologies specification for semantic situation modeling and reasoning in pervasive environments. Despite the benefits of these works towards situation identification concerns, there are research works that do not take into account context assumptions such as unhandled composite situations and situations' priority. In this research, we represented the composite situations using various operators of composition and the priority of situations are specified by predefined rules.

Roberto et al. [6] propose SHERLOCK system that provides semantic techniques to exchange contextual knowledge between devices, and to assist users in different context changes by selecting appropriate services for his current needs. SHERLOCK relies on OWL (Ontology Web Language) for knowledge modeling and reasoning using local ontology where several smart objects and user situations are

specified. However, there is an apparent lack of a new way to manage composite situations and users' needs at one time in a dynamic and intelligent way.

Da et al. [5] propose an ontology-based context-driven management middleware entitled Kalimucho-A, which established semantic-based distributed context control and management. This research study facilitates the management of smart service related to the context data. However, it lacks of concurrent events management technique and suffers from dynamic parallel management of parallel incoming events and identification of composite situations.

To address this issue, Angsuchotmetee et al. [7] presented an ontology model, known as Multimedia Semantic Sensor Network Ontology (MMSN-Onto), to represent and process in real-time complex events in Multimedia Sensor Network. The approach uses pipeline-based technique for complex events detection by observing and combining several sensors. It provides a distributed logic-based detection and continuous monitoring technique to identify events that occur on multimedia sensors networks. The proposed system can achieve high accuracy detection with short processing time. However, it still has limitations, as it does not identify dynamically composite situations and critical anomalies (e.g., fire situations, heart attacks and other urgent situations). Therefore, this work lacks the reconfiguration aspect and priority integration among situations on distributed applications.

Other ontologies emerged based on Semantic Sensor Network (SSN) [8] such as HSSAN Ontology (Hybrid Semantic Sensor Networks) [9], HealthCare Ontology [10], and Smart-Home Ontology [11]. They provide a direct mapping between event-based sensors and OWL context-oriented patterns. However, these ontologies lack a high semantic-based description with the growth of heterogeneous smart objects. Our proposed work ensures the continuity of service reconfiguration at the semantic level and enhances the response time through explicit semantic dependency between situations and their relevant context sources.

Lu et al. [12] present a lightweight ontology for IoT (IoT-Lite) expanded from the standard SSN ontology [8]. IoT-Lite ontology models the key concepts of IoT environment (i.e. coverage, actuators, services, entity, and virtual entity, attribute and measure unit). This ontology enables sensors data to be interoperable and explored on heterogeneous systems with less effort and fast interrogation time. However, IoT-Lite does not describe many essential concepts such as user profiles, its preferences, explicit descriptions of user's composite situations and its context usage. This poor description may lead to a lean user experience resulting in weak service customization. In our work, similarly to that IoT-Lite ontology, we describe sensors/actuators, events, situations as semantic services to obtain the same genericity. Furthermore, we incorporate the concept of microservices (i.e., small unit of code) in these semantic services that allow different functionalities such as observing sensors and identifying parallel situations at a semantic level in a specific smart domain. In comparison to the IoT-Lite ontology, the objective of our research is to conduct a user-centric approach, where user needs and its composite situations in multi-domain smart environments are well described.

In an attempt to recognize complex daily living activities (ADLs) [2] in a smart environment, Okeyo et al. [3] propose an ontological solution for composite activity (e.g., concurrent and interleaved tasks) recognition in smart homes. The

knowledge bases are modeled as Daily Living Activity (ADL) ontology for both simple and composite activities. This approach provides a multi-agent framework to enable multiple activities to be simultaneously monitored and recognized. This framework uses SWRL with JESS engine to infer logical composite activities. However, this work lacks the concept of activity's priority that can play a significant role in human activities scheduling to determine urgent activities.

Ramirez et al. [13] introduce a graphical framework based on DSL (Domain Specific Language) for specification and evaluation of elderly/handicapped Daily Living Activities (ADL) according to the AGGIR (Autonomy Gerontology ISO-Resources Groups) grid variables. This DSL has been used for integrating spatiotemporal operators to manage time and location of composite activities and complex events within a home environment. Therefore, they define a situation as a set of occurred events. Three AGGIR variables (i.e., washing, toileting, and transfer) have been picked for evaluation purposes. Results show the accuracy of the proposed system to track and control the occurred events and tasks of the elderly. However, this DSL does not handle the composition of multi-domains situation in which situations from different smart domains (e.g., health with transport) can be combined through combination operators such as parallel, sequential, recurrence, and alternative. The multi-domain situation combination can offer a new user's experience by incorporating different smart domain situations where users can manipulate and take benefit from all the existed smart-\* domains. In addition, this work lacks the concept of activity's priority that can play a significant role in human activities scheduling to determine urgent activities for elderly.

**Table 1.** Comparison of related ontology approaches

Related works	Composite situation	Multi-domains	Scalability	Flexibility
[6]	×	City	Medium	Medium
[5]	×	Generic	Low	Low
[7]	Partially	Home	Medium	Low
[9]	×	Generic	Low	Low
[10]	×	Health	Low	Low
[11]	×	Home	Low	Low
[12]	×	Generic	Low	Low
[3]	×	Home	Low	Low
[4]	Partially	Home	Low	Low

Table 1 is a summary of the limitations and strengths of the situation modeling approaches cited bellows. The works are compared in terms of composite situation modeling, multi-smart domains situation modeling, scalability and flexibility. These works have many limitations regarding several reasons. The first reason, major solutions lack of the high scalability (*handling big volumes of context data, continuous sensor's events and user's mobility*), which directly influences the whole reasoning process. The second reason, major solutions has been carried out in supporting specific smart domain that affect flexibility under several moving locations in several smart environments. The third reason, the majority of works lack the modeling and identification of composite situations using combination operators such as parallel, sequence, recurrence, and alternative. Taking into account the limitations of the state-of-the-art work, thus a unified semantic model for smart-\* domains is the main contribution of this paper.

### 3. CONTEXT AND SITUATION PROPERTIES ANALYSIS IN SMART-\* DOMAINS

To analyze the context properties of heterogeneous smart environments and make the context management easier, the properties-oriented domain analysis techniques is used to extract common and specific concepts of various smart domains. Our aim is to build a semantic model based on unifying concepts of multiple smart domains in order to deliver a well-enriched and more accurate context ontology, user's constraints violations and composite situation checking.

#### 3.1 Concept extraction based on case studies

This section revealed four categories of smart case studies that are extremely important for extracting the common and specific concepts which represent context and composite situations properties in assisting users in various environments (smart assisted living, smart workspace, guided and control car and healthcare). These smart environments can react to users according to the daily-life events that occur in different domains (e.g., health, social and professional). The context properties can fall under any of the corresponding subclasses of the five key classes (Context, Situation, Smart objects, Domain, and Service). This will be followed by a comparison between the four categories of smart-\* domains (home, health, car and office).

##### 3.1.1 Smart-home case study: Smart assisted living for people using smart devices

Looking first case study is about smart home that use heterogeneous sensors (e.g., temperature sensor, humidity sensor, luminosity sensor, movement sensor, camera, noisy sensor, kitchen sensor, etc.), assistive devices (e.g., smart-TV, laptop, temperature dashboard, alarm system dashboard, etc.) to capture home context information and activities of user in his smart home with daily situations such as open the curtain, adjust the lighting, turn on the air conditioner, open the garage, activate the alarm system, turn on the kitchen, etc. Home context information are analyzed for successful detection and generation of commands. These commands can be either triggered manually based on user decisions, or automatically based on home context information and user preferences. Moreover, the availability of smart devices has involved in many other fields including security (e.g., providing assistance when a potentially dangerous situation occurs such as fire detection, where the system can trigger automatically the fire extinguishing system), security (e.g., automated temperature adjustment), and energy-saving (e.g., lights control). All the components of the smart home are summarized in Table 1.

##### 3.1.2 Smart-health case study: Assisted living and successful care system for people

The next case study is about a proactive system that enable greatest fear for elderly people and afford healthy and safe lifestyle for patients or loved people. It is necessary to detect cancer, heart diseases, and diabetes as such urgent situations that speed up the care services for people. For this kind of monitoring there needs to be dedicated human body sensors (e.g., glucose sensor, body temperature sensor, etc.) or embedded sensors in pervasive environment (e.g., camera, movement detector, etc.). The system analysis this context information and identifies abnormal situations (e.g., high glucose level) to intervene automatically (e.g., inject insulin).

For instance, in case of critical anomalies (e.g., like heart-attacks), the system triggers an immediate response to the closest urban gateway and calls for immediate urgent medical attention where the victim's coordinates are found and submitted. After that, the system searches for the closest available ambulance and requests the nearest hospital for immediate preparation for hosting the patient. Therefore, pervasive environment capabilities can be used in healthcare applications to enhance intervention services while reducing the heavy load of nurses in hospitals. All the components of the smart healthcare monitoring system for diabetic people are summarized in Table 1.

### 3.1.3 Smart-car case study: Automated guided and traffic control smart car

This case study about a pervasive monitoring system that spans several components including vehicle tracking, communications between vehicles (e.g., interchanging context information such as speed, destination, distance between them, etc.), coordination between vehicles and traffic control systems to avoid congestions. Therefore, an intelligent traffic control system is deployed to avoid crowded streets based on potential traffic congestion, and offer more efficient navigation systems for users. Looking outside of the

immediate home environment, it is necessary that for any unexpected traffic congestion situation on the current route, the system must be notified automatically and must react dynamically to select a new better road towards the user's destination. The summary of system components based on the proposed model is shown in Table 1.

### 3.1.4 Smart office case study: Building a successful and efficient workspace service

The final case study, smart office application can offer a great potential for the better goals and benefits. The application facilitates the management and distribution of tasks to workers and enhance employee safety. This is critical because, even if employee is totally stressed out, the system provides information on whether the employee is in good location of workspace, or whether tasks have performed, and whether the relative needs to take action. For instance, the system depends on RFID (Radio Frequency Identification) and GPS technologies to track and identify workplace employees. As workers are equipped with RFID readers, the system can track automatically the development of activities and indicates the remaining daily tasks of the employee. These are examples of the basic context information that we are summarizing in Table 2.

**Table 2.** Comparative study of four smart domains

	Main Concepts	Domain Concepts	Smart-Home	Smart-Health	Smart-Office	Smart-Car
<b>Context</b>	<b>User</b>					
	Profile	User	√	√	√	√
		Patient	×	√	×	×
		Physics	×	√	×	×
		Worker	×	×	√	×
		Driver	×	×	×	√
	Preference	Implicit	√	√	√	√
		<b>Host</b>				
		Explicit	√	√	√	√
		Laptop	√	√	√	×
		Smart TV	√	×	√	×
		Tablet	√	√	√	√
		Mach. Coffee	√	√	√	√
		Smartphone	√	√	√	√
		<b>QoS</b>				
		Continuity	√	√	√	√
		Efficiency	√	√	√	√
		Safety	√	√	√	√
		Interoperability	√	√	√	√
		Scalability	√	√	√	√
	<b>Environment</b>					
	Location	Absolute	√	√	√	√
		Relative	√	√	√	√
	Time	Absolute	√	√	√	√
		Interval	√	√	√	√
	Condition	Noise	√	√	√	√
		Humidity	√	√	√	×
	Document	Video. Image	√	√	√	Text
Modality		Click... Touch	√	√	√	Voice
Social	Accounts	√	√	√	×	
	Friends	√	√	√	V2V	
<b>Simple Situation</b>						
<b>User's Situation</b>	Normal	Wake Up	√	×	×	×
		Shower	√	×	√	×
		Drink Coffee	√	×	√	×
		Working	√	√	√	×

Main Concepts	Domain Concepts	Smart-Home	Smart-Health	Smart-Office	Smart-Car
	Meeting	√	√	√	×
	Driving	×	×	×	√
Urgent	Low Glucose	√	√	√	√
	Fire	√	×	√	√
	Intrusion	√	×	√	√
	Car Accident	×	×	×	√
<b>Composite Situation</b>					
Normal	Wake up-shower	√	×	×	×
	Shower-Drink	√	×	√	×
	Drink □ Email	√	×	√	×
Urgent	⊃Low Glucose	×	√	×	×
	⊃High Glucose	×	√	×	×
	Glucose -Heart	×	√	×	×
<b>Service Listener</b>					
	User Monitor.	√	√	√	√
	Health Monitor.	×	√	×	×
	Activity Track.	×	√	×	×
	Cardiac Implant	×	√	×	×
	Location	√	√	√	√
	Time Service	√	√	√	√
	Care Message	×	√	×	×
	Remote Ctrl	√	√	√	√
	Road Ctrl	×	×	×	√
Service	<b>Action Service</b>				
	Wake Up	√	×	×	×
	Shower	√	×	×	×
	Coffee Mach.	√	×	√	×
	Message	√	√	√	√
	Intrusion	√	×	√	×
	Fire Fighting	√	×	√	×
	Inject Insulin	×	√	×	×
	Risk of Death	×	√	×	×
	Remote Diag.	×	√	×	×
	Car Ctrl	×	×	×	√
<b>Sensor</b>					
	Temperature	√	×	√	×
	Light	√	×	√	×
	Motion	√	×	√	×
	IP Camera	√	√	√	√
	State Of Door	√	√	√	√
	Coffee Mach.	√	×	×	×
	Humidity	√	×	×	×
	Water Temp.	√	×	×	×
	Garage	√	×	×	×
	Body Temp.	√	√	√	√
	Heart Rate	√	√	√	√
	Insulin Level	√	√	√	√
Smart Objects	Location	√	√	√	√
	Time Sensor	√	√	√	√
	Traffic	×	×	×	√
	Video Recorder	√	×	√	×
	RFID Sensor	×	×	×	√
<b>Actuator</b>					
	Light Switch	√	×	√	×
	Door Locking	√	×	√	×
	Smart Mirror	√	×	√	×
	Inject Insulin	×	√	×	×
	Fire Alarm	√	×	√	√
	Intrusion Alarm	√	×	√	√
	Alert	√	√	√	√
	Light Switch	√	×	√	×
<b>User's Domain</b>					
Domain	Security	√	√	√	√
	Home	√	×	×	×
	Health	√	√	√	√
	Office	×	×	√	×

### 3.2 Common and specific-concepts of smart domains

A comparison between smart home, smart health, smart car and smart office is carried out to find the common context properties as well as variable context properties based on the concepts identification framework discussed in section 3.1. The mapping of the fundamental metadata, the domain concepts as well as the commonality and variability of studied smart environments shown in Table 2. There exist a similarity and difference in the studied smart environments as shown in Table 2. With the identified similarity and difference, generic context properties and situations for smart-\* domains can be recognized automatically and the context-aware composite ontology can be developed with flexibility, interoperability and reusability. We break down each smart domain into a set of main classes that cover the common information of all smart-\* domains. These key concepts can be extended to domain specific-concepts hierarchically.

The recognition accuracy during the situation's identification process is based on the situation rules using the number of detected events and checked conditions. Further, the semantic expression of the person is accurately identified according to user's current location and time.

### 4. PROPOSED ONTOLOGY

The main objective of our proposed Multi-OCSM (Multi-layered Ontology-based Composite Situation Model) ontology is to: (1) - describe various smart devices communicating through various protocols and media types (text, audio, image, video, etc.), (2) – classify simple and composite situations according to the context of user for efficient and effective situations managing, (3) – provide formal semantics for user's domains structure as well as relations between them,(4) – represent semantic service information: service role, service category, semantic of inputs/outputs, (5) – unify the properties and constraints of user for easier managing and sharing, and (6) – automatic deduction of service events and actions corresponding to semantic specifications of user's constraints. Moreover, this ontology accelerates the process of identifying relevant situations by matching user-profiles and context information on a semantic level.

A semantic classification of situations rules is useful for intelligent decision-making process depends on the current user's requirements and its context. The purpose of situations classification is to improve the efficiency of the identification process by prioritizing them according to one or more criteria such as situation category, the activity of user, user's location, and situation coverage. Our ontology aims to capture the user's context at the semantic level. Figure 1 depicts Multi-OCSM ontology model for mobile applications.

#### 4.1 Conceptualization of our ontology Multi-OCSM

Multi-OCSM ontology is a novel generic ontology for Smart-(health, home, cities, office, etc.). This ontology includes concepts that model all the functionalities of the intelligent environment (i.e. event services, situation services and action services). Multi-OCSM is an ontology-based composite situation that can play crucial role in promoting inference through the formal representation of domain knowledge. This ontology extends standard ontologies including the Semantic Sensor Network (SSN) ontology [14], the DOLCE Ultra-Light ontology (DUL) [15] and the OWL Service (OWL-S) ontology [16] which are already reusable. Multi-OCSM ontology can be used for several ends. Members of a given user's domain group can use Multi-OCSM to communicate and share information between them. It can be used to interpret and detect simultaneous events and various users' situations. The Multi-OCSM consists of four layers as follows:

- Layer 1 (smart objects and devices layer): this layer contains several physical/logical sensors and devices according to their type and characteristics. A wide range of sensors help to collect context data from the physical world.
- Layer 2 (context layer): this layer is used to classify contextual information provided by the smart objects and devices layer using multi-dimensional aspects such as outdoors context (e.g., place, time, network) and user context (e.g., user's role such as a student, employer or teacher, user's current location, time and user's agenda). Our ontology uses location, time and role information to classify new situations rules.
- Layer 3 (situation layer): a composite situation is described as sub-situations combinations based on the semantics provided by the context layer using composition operators.
- Layer 4 (service and application layer): as previously presented, each situation identified by the situation layer corresponds to a set of appropriate action services. Each service is represented on mobile applications by its distributed microservices. The adaptation of an application may be achieved by reconfiguring its microservices.

The proposed layers are applicable in different smart domains (health, home, cities, car, office, etc.) through Kali-smart middleware [5]. We use Kali-smart for dynamic reconfiguration that enables adding/removing/updating services on decentralized multimedia sensors and devices. The middleware is run simultaneously on all available sensors and mobile devices. As shown in Figure 2, the ontology is defined in five key classes: Context (a), Situation (b), Smart Object and Device (c), Domain (d), and Service (e). These classes' define basic concepts found in any intelligent environments

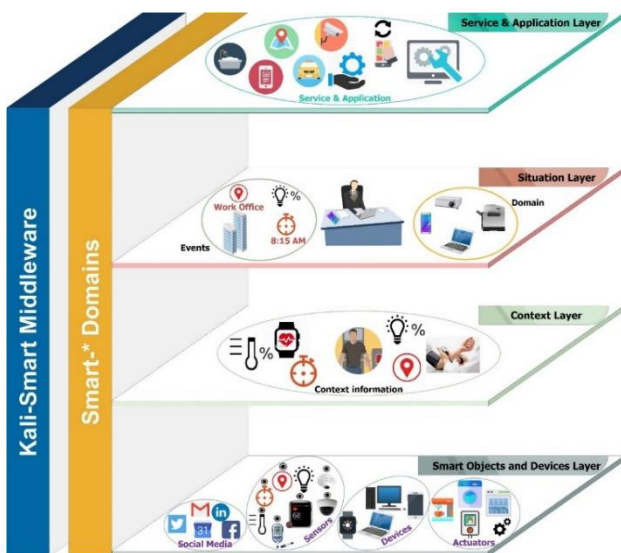


Figure 1. Multi-OCSM ontology layers



that seeks to provide suitable services to users in compliance with their current needs and contextual situations.

#### 4.2 Smart objects class

The first-class consists of several heterogeneous sensors and actuators. Each one has a unique identifier, a local name, a location and other attributes describing its properties (Figure 3). It contains several sensors that monitors contextual data, which are analyzed using different reasoning techniques based on various functionalities. We have two primary sensors types: physical and logical sensors. Logic sensors include calendar sensors, time sensors, social media sensors, hardware sensors, and so on. Physical sensors may contain bio-sensors, weather

sensor, and environment sensor.

#### 4.3 User's domain class

Smart environments can react to users according to daily-life events that occur in different domains (e.g., health, social and professional) using several devices. Each device is responsible for running multiple services. Each service can include one or more Context-aware Intelligent Micro-Services. Any service is deployed on a given smart device. It serves a particular need and has specification for hardware and software requirements (using the description profile) to run in an optimized manner.

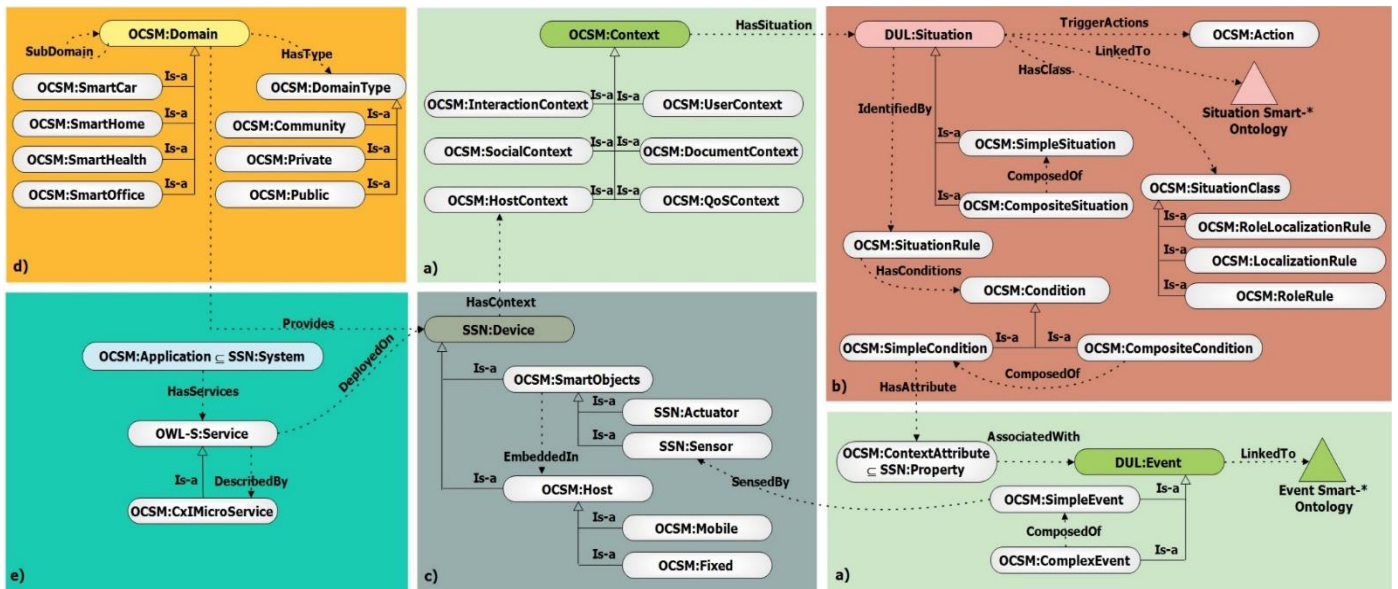


Figure 2. A generic context-aware composite situation ontology

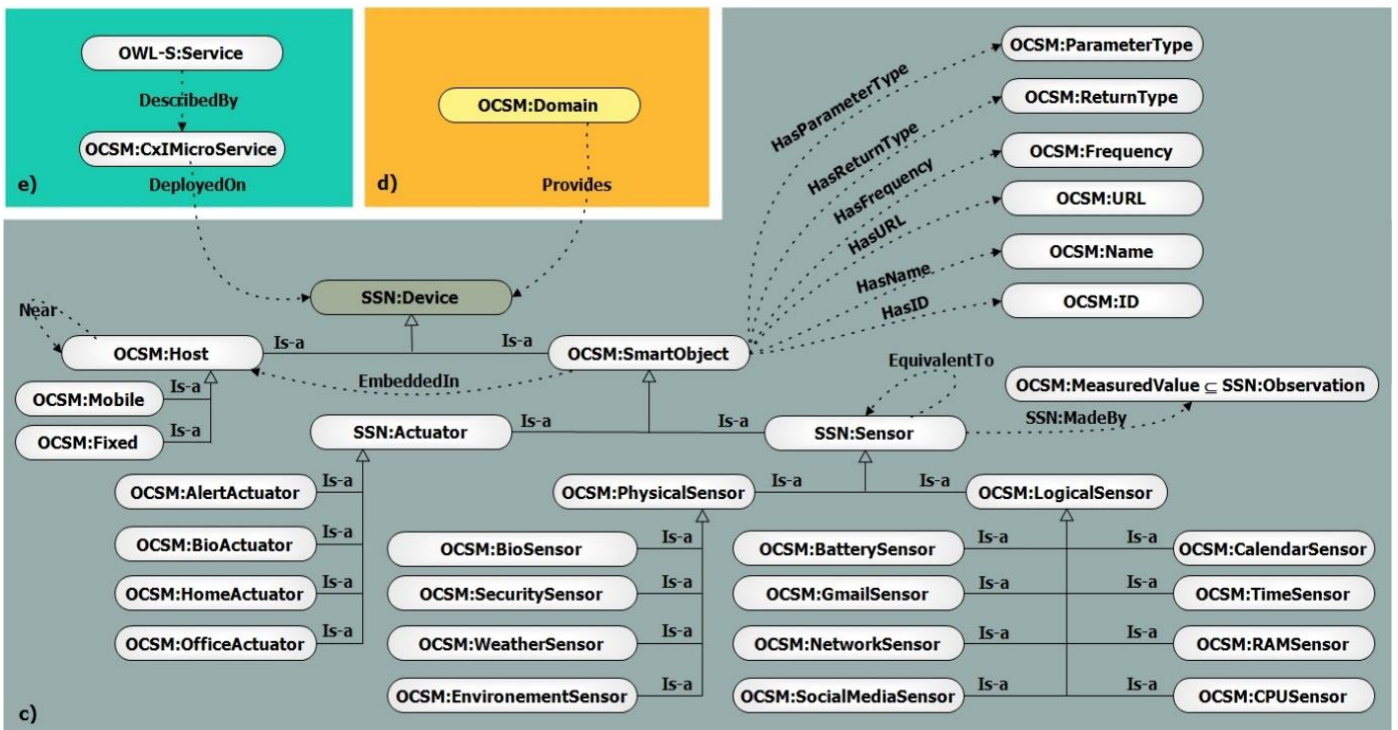


Figure 3. Context smart object Multi-OCSM sub-ontology

#### 4.4 Context class

The third-class Context plays an important role in allowing consumers to provide appropriate facilities based on their local environments (see Figure 4). It's targeted at classifying the contextual information on multiples dimensions:

- The User Context can include a user's location, time and user's agenda detailing all activities undertaken in every daily life (e.g., watching TV or listening to music).
- The Environment Context identifies place and time that are essential for the recognition of activity. Location: defined by GPS coordinates or semantic location terms. Time: defined either by exact time or timestamp or a semantic time term such 'morning'.
- The Host Context represents various fixed and mobile hosts and its related properties. For example, a mobile device is characterized by memory size, CPU speed.
- The Interaction Context defines numerous modalities of interaction (click, gesture, voice, touch). The meaning of the relationship defines a series of properties belonging to particular form of modality.
- The Document Context describes the type and properties of documents (text, image, video and audio) relevant to a specific media type.
- The QoS (Quality of Service) Context defines the service quality of mobile application, described as QoS criteria such as (1) continuity of service; (2) performance of service; and (3) health and safety.
- The Social Context defines the context information created by various social networks and relationships between multiple users at a high level of abstraction.

Each context category has a set of different context attributes. For instance, the device has multiple context attributes (memory capacity, bandwidth). Each context attribute is associated with an event (i.e. the sensed data value by a specific sensor) as presented in Figure 4. We extend the work of Dromzée et al. [17] by using two types of attributes to describe events: area, features vector (picture, video, sound). It enables understand events using semantic meaning independent of their source. Indeed, a semantic value, like "Low" can be applied on multiple contextual parameters, such as external temperature, glucose level, bandwidth, etc. Hence, depending on the context used, the semantic meaning may be entirely different. Moreover, in a particular context, a semantic attribute may have different meanings for many intelligent domains (home, office). For instance, the semantic value "Presence" that belongs to a person corresponds to GPS coordinates when considering smartwatch as the data source, or it belongs various feature values when considering monitoring camera as the data source or belongs to logic value when considering activity tracking sensor as the data source. Actually, this work will potentially be used with a large number of semantic values (see Figure 5). For example for spatial details, one can use semantic values like 'closeTo', 'farFrom', 'intersectWith', etc. However, the system gives more accuracy by inferring the right context information on the right time based on the exact interpretations of constraints. For instance, "Low Temperature" is assigned to a specific value depending on user's location inside or outside home. Therefore, the execution of situation identification process leads to effective decisions that takes as inputs user's current location and time.

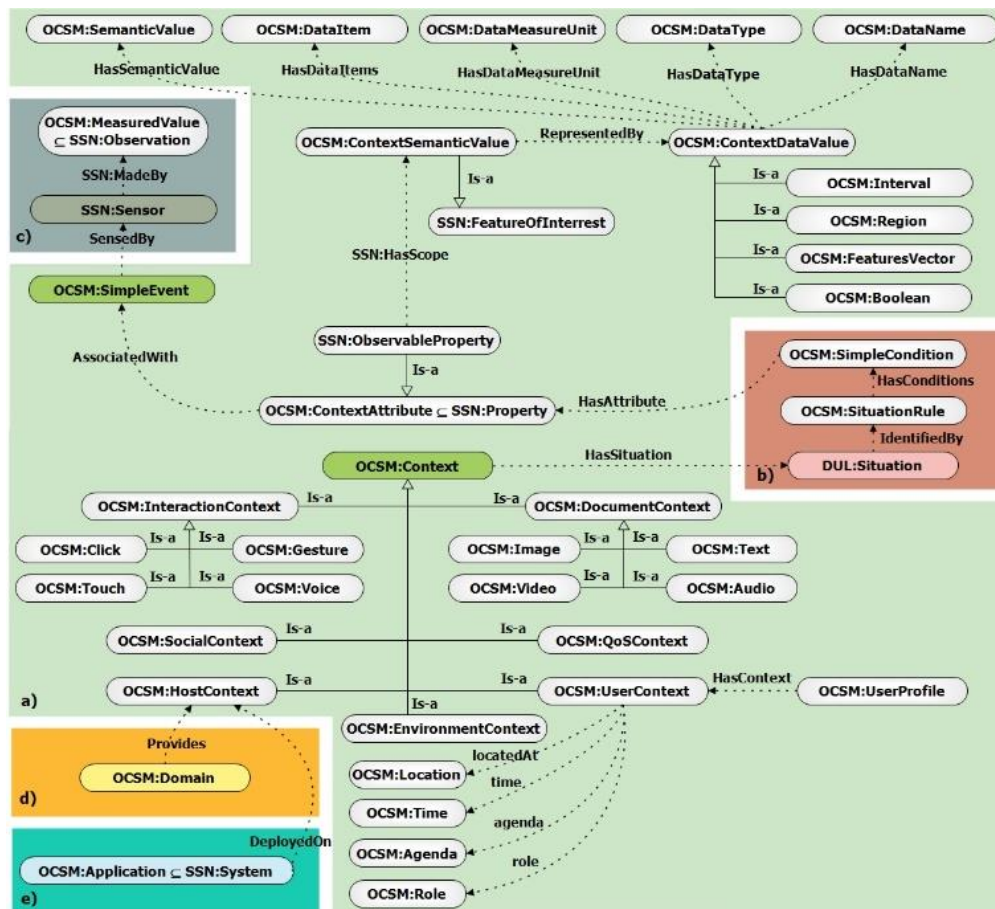


Figure 4. Context Multi-OCSM sub-ontology



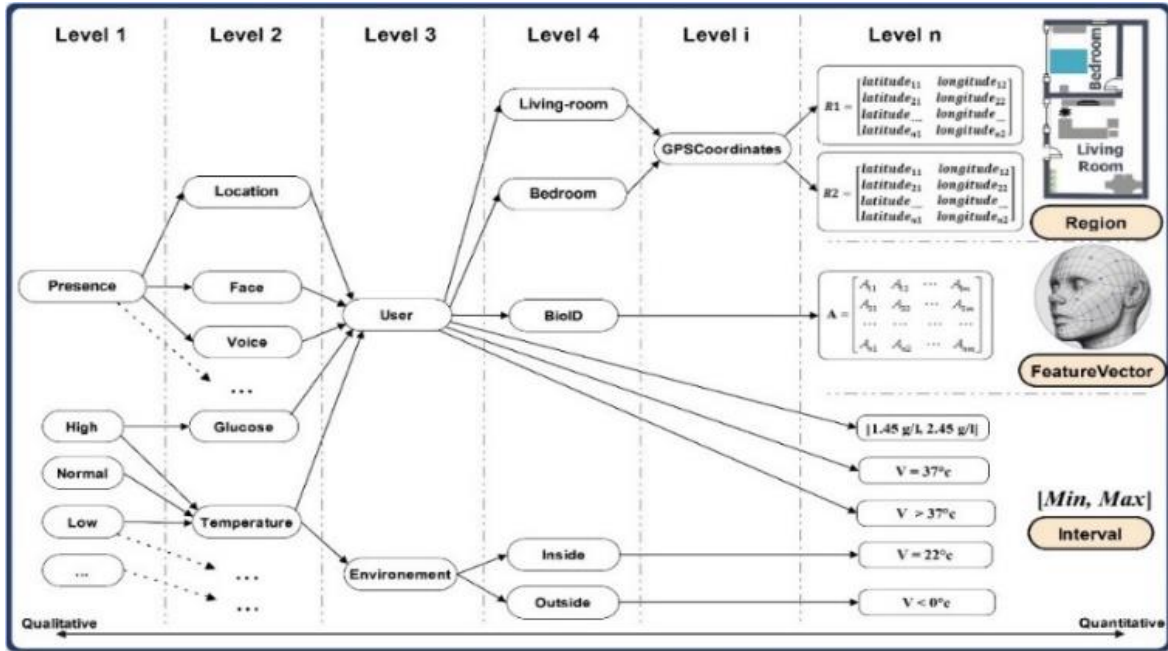


Figure 5. Interpretation of events extended form [17]

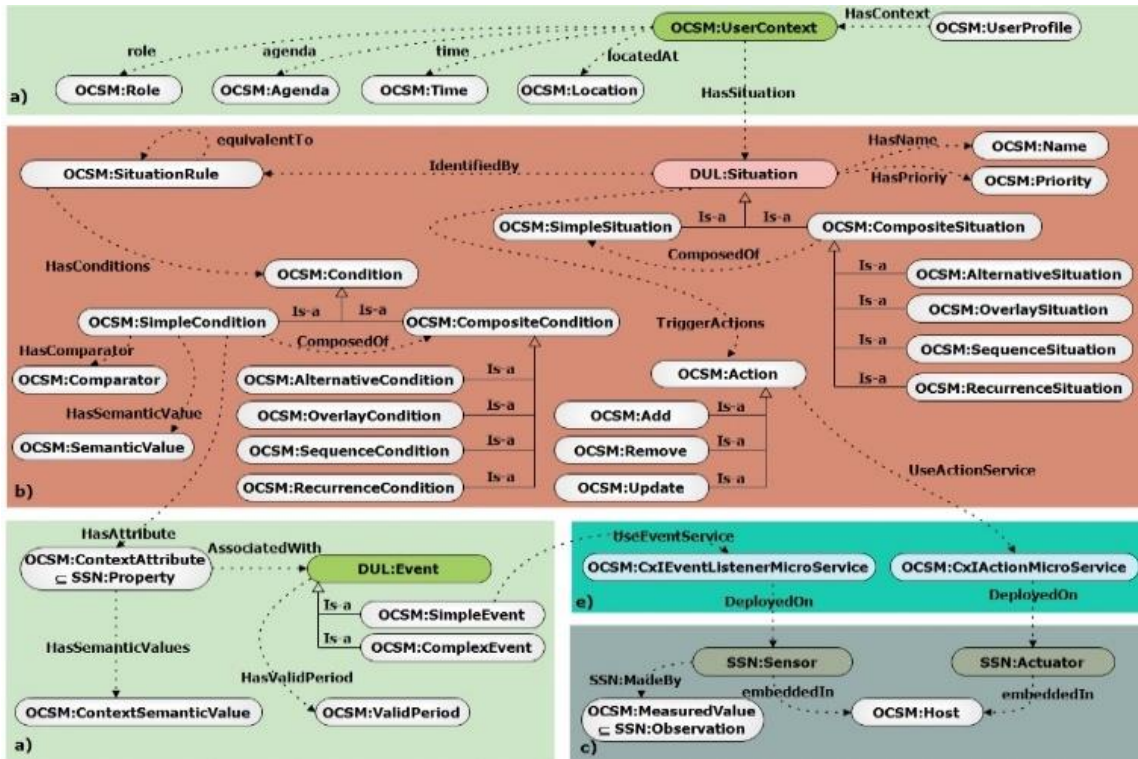


Figure 6. Situation Multi-OCSM sub-ontology

#### 4.5 Situation class

Situation is the key class of intelligent mobile application. Situation is a collection of events which are occurring at a specific place, time and the conditions that can check certain events, so the situation need to be identified [18]. Figure 6 shows the ontology model of situation, describing the context of the user, modeling the rules, semantic values of context data and situations. In various smart domains, a user has different sets of situations. A situation is identified by a situation rule. Each rule is defined by a combination of multiple conditions. Every condition has a context attribute (e.g., location, time,

role, or glucose level.). The context attribute may be attached to a semantic value (e.g. home for the context attribute location, morning for the context attribute time, student for the context attribute role and high for the context attribute glucose level) using a comparator (e.g. for place within near, close and far; for time within before, after, while) to enrich the description of the condition. Besides the situation identification process, if the situation has occurred (i.e. situation rule is checked), we give what the user wants. Thus, we determine what service actions to trigger when the situation is recognized by the situation-actions association. The output (i.e. precision and response time) differs according to the number of conditions,

importance of context attribute and the current situation of the user. The proposed approach provides classification of context attribute, simple and composite situations according to the context category and the context of user for efficient and effective situations managing. In case of emergency state, we provide alternative solutions with low response time and energy consumption compared to tedious operations. We can use a new intelligent service-based management algorithm that collects the current state of each operation then splits the queued operations on processor cores with the slightest load.

#### 4.6 Service class

The distributed smart mobile application is made up of a collection of distributed services. A service is an autonomous Context-aware Intelligent microservices that make up an application (CxIMicroService see Figure 7). The benefit of decomposing the application service into different small

services is that it improves modularity and flexibility to provide support in managing situations under the mobility of users. Each CxIMicroService has an autonomous functionality (i.e. task implements the runnable interface) that serves as the body of a thread. The latter will be deployed using the Service Executor of our framework. We extend the CxIMicroService with a CxISituationIdentifierMicroService, CxIEventListenerMicroservice, and CxIActionMicroservice to define situation rule identifier, situation’s event listener and situation’s actions respectively. These parallel microservices are capable of making some useful reasoning tasks on its context smart domain (location, context, etc.) with the help of an ontology-based model. Using our service Multi-OCSM, we can construct a hierarchy of collaborated CxIMicroservices using other specialized CxIMicroservice in the situation reasoning process. In this case, a supporting framework has the capability of sharing parallel-detected events and identified situations in a distributed smart environment.

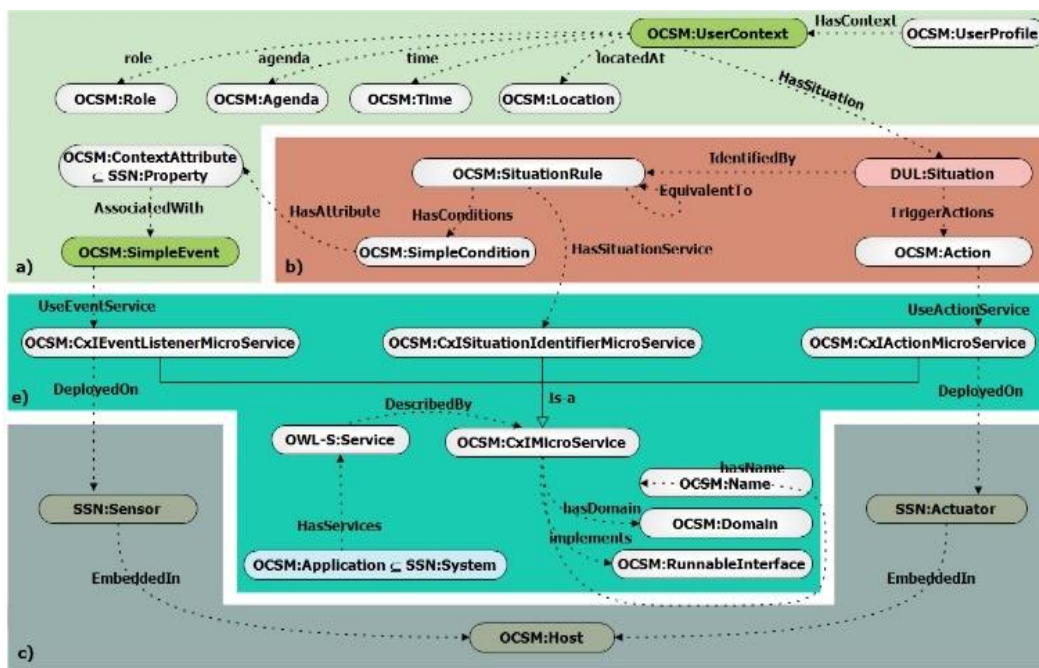


Figure 7. Service Multi-OCSM sub-ontology

## 5. IMPLEMENTATION AND VALIDATION

### 5.1 Implementing the ontology

After the conceptualization of the ontology Multi-OCSM, we can start the implementation process. The implementation process consists of two steps. First, we implement the ontology Multi-OCSM (classes, subclasses, and their object properties) and instantiates some individuals from each class, while the second is to fill it by defining some formal logics as well as interrogation rules. Both steps were generated using the Protégé 7.1 editor [19]. This visual modeling tool provides OWL (i.e. Ontology Web Language) for describing our Multi-OCSM ontology that has been used to reason about its defined classes and individuals. The Multi-OCSM ontology is made-up of four upper-level classes: Smart Objects and Devices, Context, Situation, and Service & Application. First, we create Multi-OCSM hierarchical classes and its subclasses (Figure 8.a). For instance, the class Context has seven sub-classes: ContextActivity, ContextDocument, ContextEnvironment,

ContextHost, ContextInteraction, ContextQoS and ContextUser. Each class consists set of datatype properties (Figure 8.b) and objects properties (Figure 8.c). For instance, the Situation class has an object property “HasCondition” which is linked to Condition class. This relationship means that the Situation class can have one or more conditions.

Figure 9 illustrates a part of Multi-OCSM. It shows four fundamental classes: Situation, Condition, Event and Action. In addition, it shows the semantic links between these classes (e.g. Situation *has* Condition, SimpleCondition *Is-a* Condition, ComplexEvent *ComposedOf* SimpleEvent, etc.).

The main 4-classes (i.e. situations, events, conditions, actions) are defined in the model by means of simulations. With simulation, we included the dataset generator that creates a specified number of situations and generates various types of sensor events (states of doors, lights, home temperature, user location, user glucose level, user movement states), resulting in 50,000 sensor events. An operationnql analysis flowchart is illustrated in Figure 10.

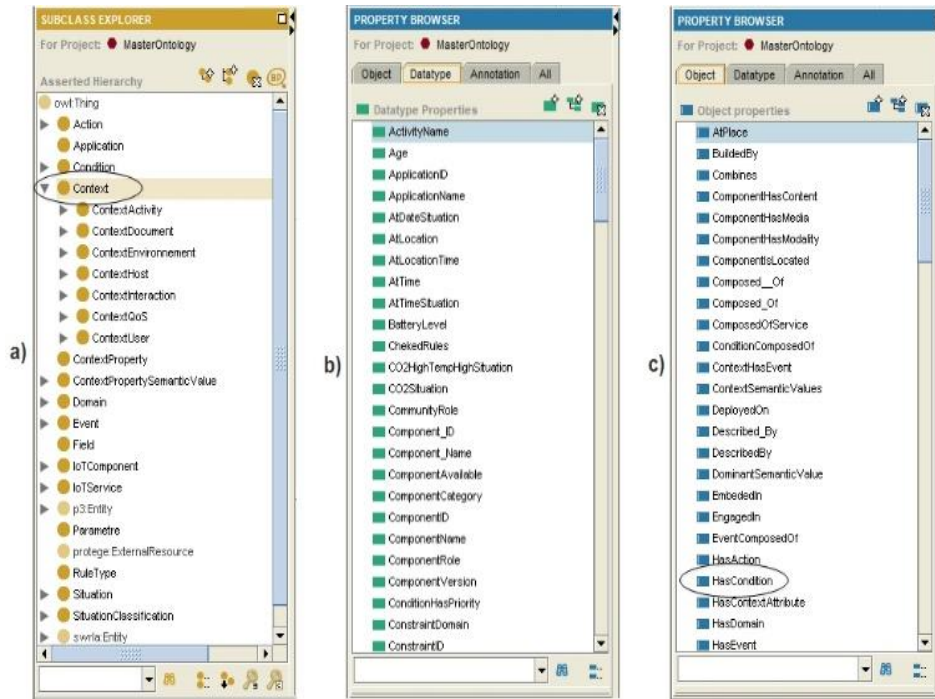


Figure 8. Implementation of multi-OCSM ontology in Protégé

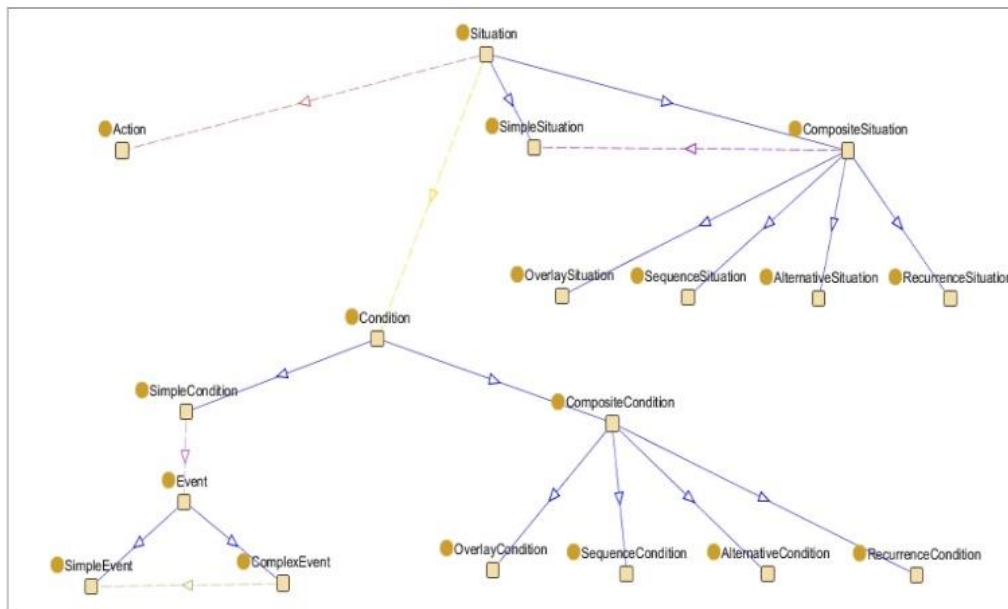


Figure 9. An illustrated example of a part of multi-OCSM generated with Jambalaya

## 5.2 Consistent rules of Multi-OCSM

Verifying the consistency of Multi-OCSM instances is an essential part of our implementation, especially the consistency structure of composite situations and user-defined constraints (with various context properties) while providing more formal semantics. To deal with it, we use the Description Logic (DL) [20] to represent Multi-OCSM in a formal and structured way. The Two components traditionally called Terminological Box (TBox) and Assertion Box (ABox) constitute Multi-OCSM expressed in Description Logic:

- TBox (assertions on concepts): describes terminological information by defining basic or derived concepts. Also, it defines how these concepts relate to each other.

- ABox (assertions on individuals): describes the information that characterizes individuals. This information is expressed by specific or local assertions.

The first aspect was ensured via the protégé editor [19], where concepts, relationships between concepts, concepts properties and creation of individual were created using the visual interface of Protégé. Besides that, the reasoning mechanisms is fundamentally based on three parts: 1) – the source of knowledge Multi-OCSM, 2) – the interaction between the ontology and the system through the SPARQL query language and 3) – the reasoning engine based on parallel semantic-based composite situation identification framework. Table 3 presents DL example of our ontology.



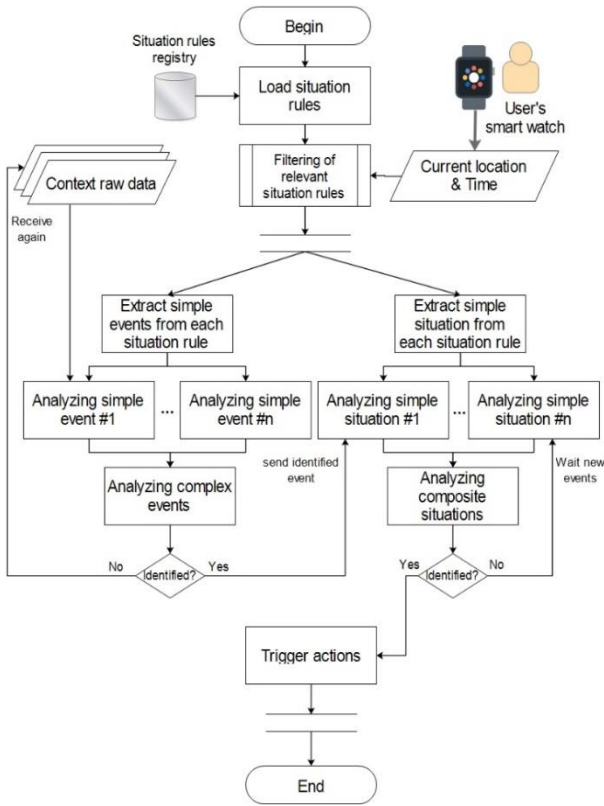


Figure 10. Flow chart of events and situations analysis

Table 3. DL example of our ontology

TBox	ABox
$Context \sqsubseteq Thing$	$UserContext(User\_Rahim)$
$UserContext \sqsubseteq Context$	$HasID(User\_Rahim, U\_1)$
$\geq HasID \sqsubseteq String$	$HasName(User\_Rahim, Rahim)$
$\geq HasName \sqsubseteq String$	$HasAge(User\_Rahim, 25)$
$\geq HasAge \sqsubseteq Int$	$HasSituation(User\_Rahim, S1)$
$\geq HasSituation \sqsubseteq Situation$	$asSituation(User\_Rahim, S2)$
$Situation \sqsubseteq Thing$	$Situation(Sit\_S1)$
$\geq HasID \sqsubseteq String$	$HasID(Sit\_S1, S\_1)$
$\geq HasName \sqsubseteq String$	$HasName(Sit\_S1, H\_Glucose)$
$\geq HasCondition \sqsubseteq Condition$	$asCondition(Sit\_S1, Condition1)$
$\geq HasPriority \sqsubseteq double$	$HasPriority(Situation\_S1, 0.81)$
$\geq HasService \sqsubseteq Condition$	$HasService(Sit\_S1, Inject\_Insulin)$

### 5.3 SPARQL queries for multi-OCSM

SPARQL (Protocol and RDF Query Language) [21, 22] is used to express queries across diverse data sources that can be used for semantic data extraction. The use of SPARQL is based on two functional properties, simplicity and rapidity. We use SPARQL queries to extract semantic data from multi-OCSM ontology. Thus, we develop many SPARQL queries based on Multi-OCSM. The primary goal of these queries is extracting the semantic features related to monitored user context, then filtering relevant situations of the user in order to accelerate the situation identification process.

In the following we present some selection and filtering queries.

The **Query 1** selects the user's location event. It takes as parameter "user name".

```
PREFIX OSCMonto: <http://www.owl-ontologies.com/Ontology1430403694.owl#>
```

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?idEvent
WHERE {
  ?user OSCMonto:UserName \"\"+user.getUserName()+\"\".
  ?user OSCMonto:HasLocationEvent ?locationEvent .
  ?locationEvent OSCMonto:IdEvent ?idEvent .
}
```

The **Query 2** selects all the situations that do not have any spatiotemporal constraint. It takes as parameter "user name".

```
PREFIX OSCMonto: <http://www.owl-ontologies.com/Ontology1430403694.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?situation ?idSituation ?locationTime
WHERE {
  ?user OSCMonto:UserName \"\"+user.getUserName()+\"\".
  ?user OSCMonto:UserHasSituation ?situation.
  ?situation OSCMonto:IdSituation ?idSituation .
  ?situation OSCMonto:AtLocationTime ?locationTime .
  FILTER( ?locationTime = \"\"Anywhere#AnyTime\"\" ).
}
```

The **Query 3** returns all situations that have a spatiotemporal constraint. It takes as parameter "user name, user location and time".

```
PREFIX OSCMonto: <http://www.owl-ontologies.com/Ontology1430403694.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?situation ?idSituation ?locationTime
WHERE {
  ?user OSCMonto:UserName \"\"+user.getUserName()+\"\".
  ?user OSCMonto:UserHasSituation ?situation.
  ?situation OSCMonto:IdSituation ?idSituation .
  ?situation OSCMonto:AtLocationTime ?locationTime .
  FILTER(?locationTime = \"\""+location+"#"+time+"\" ||
  ?locationTime = \"\""+location+"#"+time+"\" ||
  ?locationTime = \"\"Anywhere#"+time+"\" ).
};
```

The **Query 4** returns all the sub situations of a given composite situation. It takes as parameter "id of situation".

```
PREFIX OSCMonto: <http://www.owl-ontologies.com/Ontology1430403694.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?subSituation ?idSubSituation
WHERE {
  ?situation OSCMonto:IdSituation \"\""+situation.getId()+\"\".
  ?situation OSCMonto:HasSubSituation ?subSituation.
  ?subSituation OSCMonto:IdSituation ?idSubSituation
}
```

### 5.4 Evaluation of multi-OCSM

Here we suggest a specific study that supporting users in different smart environments (home, car, office, etc.). These smart environments can react to user's daily-life events occurring in different domains (e.g., health, social and professional). Figure 11 illustrates three environments (smart-home, smart-car and smart-office) with both scalar and multimedia smart objects. The application's purpose is to manage all the context data of the health domain and daily-life activities in an efficient way by enhancing the monitoring and reporting of critical health conditions.

As we can see in Figure 12, there are various smart objects. Each smart object serves to monitor information such as a motion sensor to monitor the movement, a smart camera to monitor the activity (e.g., watches TV, drinks a coffee or

checks emails), a smartwatch to monitor the user’s locations and other sensors to detect other contextual information (temperature and humidity level, states of doors, lights, user glucose and user movement states). In our case study, the smart home acts as a pervasive environment which is composed of many rooms: kitchen, living room, bathroom, study, garage and two bedrooms (bedroom#1 and bedroom#2) (see Figure 12).

Each room is equipped with smart objects. Table 4 details a list of available smart objects and devices in the smart home.

The context information is collected by the set-top-box (e.g., position, time and video). This contextual information is useful to identify a set of users' situations.

Now let us suggest this scenario: we are interested in the daily life of a user (Adam). He lives in the smart home described above. He is a regular worker of a bank office. He has specific needs based to his current needs, preferences and situations. In order to recognize urgent situations, the system has to continuously monitor his health status. He is equipped with wearable devices (glucose meter and smartwatch). Adam sleeps in the bedroom#1.

He has an agenda, which accurately describes his planned

activities. These activities can be performed in a sequential and/or parallel order. Adam’s agenda can be used to identify various situations during the morning period, from waking up until going to the office. Below, we give a short detail about his daily-life activities. The system checks his personal agenda on Google to find appropriate way to handle his working days. Then it sets the alarm service on his smartphone. When time is ready, the framework increases the lighting of bedroom#1 steadily and automatically. The camera in the bedroom#1 can automatically recognize if he wakes up or not. When he wakes up, the system checks Adam's next scheduled task, which is: (takes a shower). The system adapts the water temperature as he leaves bathroom and shows it on the bathroom intelligent mirror. When he’s going to the kitchen; the device sends a command to intelligent coffee maker to prepare the coffee. After that, he goes to the study where he peacefully drinks his coffee. The framework deploys email service on his personal computer with all tools that he finds helpful in his area work and deploys morning news on his Smartphone. Adam’s smartwatch regularly sends GPS coordinates to the system to locate his precise location.

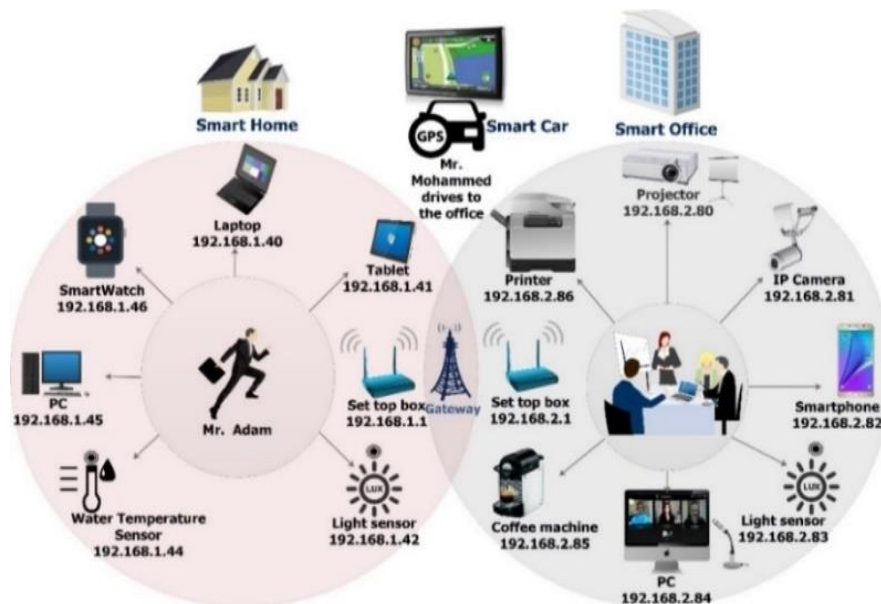


Figure 11. Different smart environments



Figure 12. An overview of a smart home

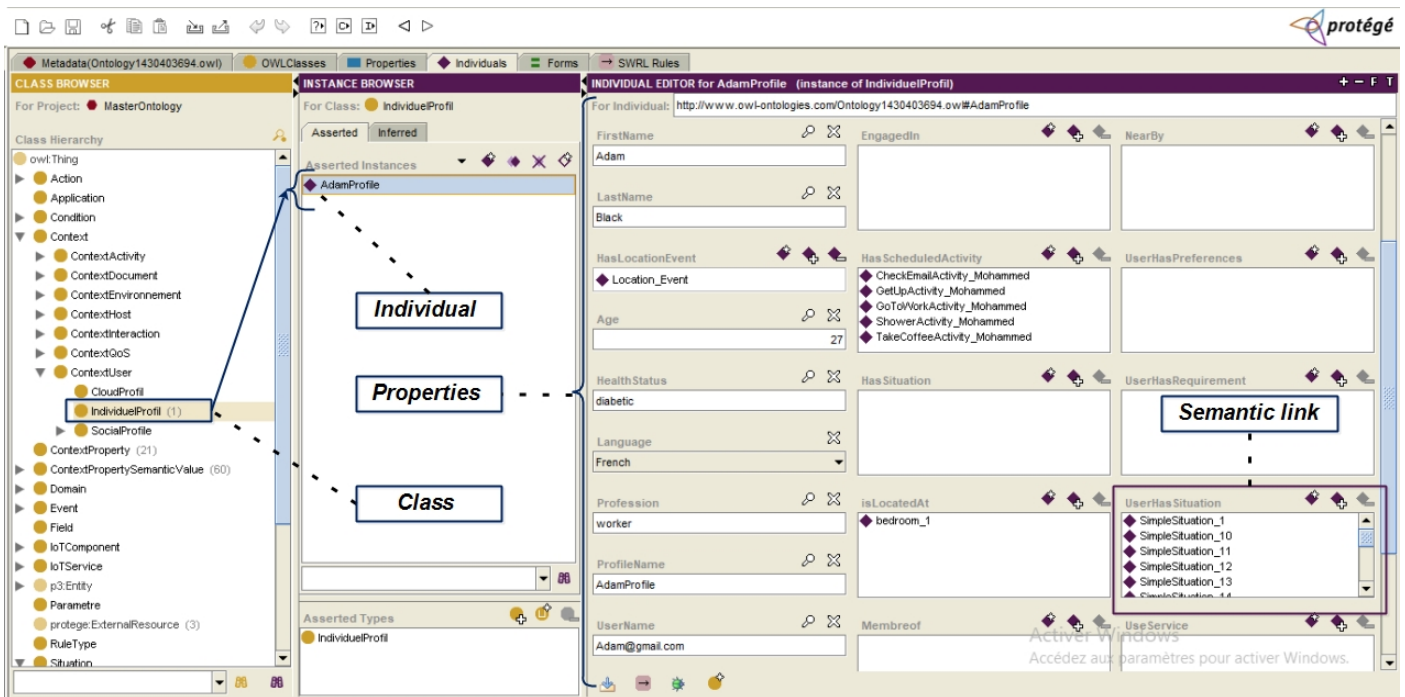


**Table 4.** Different smart objects in the smart home

Smart home-rooms	Sensors	Actuators
Bedroom#1 Bedroom#2	Temperature sensor, light sensor, state of door sensor, IP camera.	Light switch, door locking.
Living room	Temperature sensor, light sensor, motion sensor, IP camera.	Light switch. Laptop, Tablet, Smart TV. Light switch actuator, smart machine coffee.
Kitchen	Light sensor, IP camera, coffee machine sensor.	Smart TV.
Bathroom	Temperature sensor, humidity sensor, water sensor, motion sensor, Light sensor.	Light switch actuator, Smart mirror.
Study	Temperature sensor, light sensor, IP camera	Light switch actuator, PC
Garage	State of garage door, IP camera.	Garage door locking.

Figure 13 shows an instance of individual profile and its properties. As we can see, the individual AdamProfile of the class IndividualProfile has object properties (e.g. isLocatedAt, HasScheduledActivity, HasSituation, etc.) and datatype properties (e.g. UserName, LastName, Age, Profession, etc.).

After the implementation of the multi-OCSM ontology, we would like to show a concrete example represented by OWL. Figure 14 shows an example of a concrete model based on the Multi-OCSM ontology represented by OWL. This example describes a garage within a smart home. The garage is a subdomain of the domain smart home (line 8). The datatype property “PlaceID” (line 4) represent a unique identifier of the garage. The garage has a domain where we can define provided smart objects (i.e. sensors and actuators) (lines 9-20). As we can see in Figure 14, the garage has a camera sensor (lines 10-13) and a garage door opener actuator (lines 16-20). The camera sensor has an ID (line 11) and uses the service “Service\_GetCamera” (line 12). In addition, the garage door opener actuator has ID (line 17) and uses the service “Service\_Garage\_Door\_Opener” (line 18).



**Figure 13.** Description of Adam profile in Protégé

```

1  <Garage rdf:ID="Garage_1">
2  <<SuperID rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
3  >Home_1</SuperID>
4  <PlaceID rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
5  >Garage_1</PlaceID>
6  <HasDomain>
7  <<Community rdf:ID="Garage">
8  <<SubDomain rdf:resource="#SmartHome"/>
9  <Provides>
10 <<CameraIPSensor rdf:ID="CameraIPSensor_garage">
11 <<IdSensor rdf:datatype="http://www.w3.org/2001/XMLSchema#string">42</IdSensor>
12 <<UseService rdf:resource="#Service_GetCamera"/>
13 </CameraIPSensor>
14 </Provides>
15 <Provides>
16 <<AutomationActuator rdf:ID="AutomationActuator_Garage_Door_Opener">
17 <<IdActuator rdf:datatype="http://www.w3.org/2001/XMLSchema#string">A_2</IdActuator>
18 <<UseService rdf:resource="#Service_Garage_Door_Opener"/>
19 </AutomationActuator>
20 </Provides>
21 </Community>
22 </HasDomain>
23 <In rdf:resource="#Smart_Home_1"/>
24 </Garage>

```

**Figure 14.** A concrete example of OWL representation for multi-OCSM

We have evaluated the execution time of several SPARQL queries on some ontology individuals. The goal is to demonstrate how Multi-OCSM ontology can perform queries interrogation based on combining multiple queries. Note that all evaluations have been performed on a PC running Windows 10 (64 bits) on an Intel i5-2430 2.4 GHz and 8 GB RAM. We evaluate the response time of various SPARQL queries (query 1, query 2, query 3, and query 4 described in Section 5.3) on Multi-OCSM with 100, 200, 300... 1000 times. First, we run query 1 and query 2 separately multiple times. Then we combine query 1, query 2, query3 and query 4 using sequential operator and run them multiple times. Figure 15 depicts the response time of requesting Multi-OCSM with different SPARQL queries. The execution time of a single query repeated 500 times shows an average of 1000ms. Nevertheless, combining queries shows a significant increase in the response time of 2000ms. Our objective is to optimize the execution time while combining several queries. Thus, we will reduce the execution time while combining multiple queries using a parallel microservices-based approach.

In this case study, urgent situations such as high glucose situation, time sensitivity, etc. attempt to highlight the sensitivity challenges to consider when a person is interacting with the system. While situations such as user's role, user's mobility, etc., attempt to highlight the flexibility challenges that affects the way any person is switching to another available service in a short time. For example, "Care service" can be used to alert patient via alarm, etc., depending on the user preferences, constrained devices and available service. The rules of inference for the "Care service" in the case the battery level is low are as follows:

---

```
IF [BatteryLevel = 'Low']
THEN Migrate[Inject_Insulin_Service]
```

---

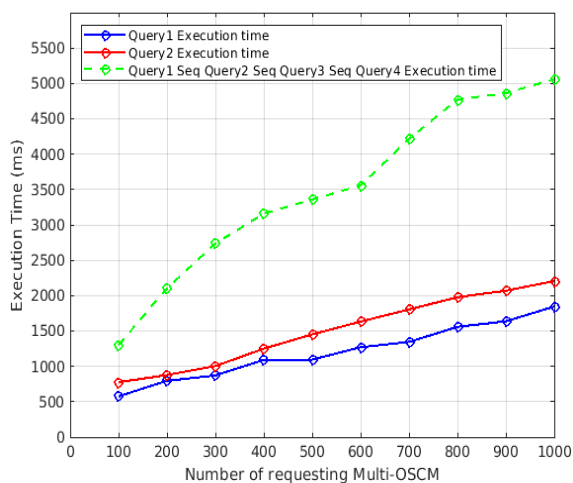


Figure 15. Multi-OCSM Query execution time comparison

## 6. CONCLUSIONS AND FUTURE WORKS

In this paper, we have proposed a novel generic user-centric composite situation ontology called Multi-OCSM (multi layered ontology for composite situation management) for representing the majority of concepts of smart-\* domains and user's profile. This ontology contains five key classes: *Context*, *Situation*, *Smart objects*, *Domain*, and *Service*. It describes both simple and composite situation rules and its semantic

description including composition operators (parallel, sequence, recurrence and alternative). This ontology hides the dynamicity of user's needs and the heterogeneity of smart objects in terms of hardware/software features, provides semantic abstraction concepts related to the context data for events simultaneous interpretation, interoperability of different incoming events and highlights the semantic rules of user. Moreover, multi-OCSM supports SPARQL queries, which allow users to explore the Ontology and to customize the situations filtering according to their constraints and preferences. The SPARQL queries are created using Protégé 7.2. In the future work we will present a novel parallel semantic-based composite situation identification framework for optimizing the situation identification process. It also presents the whole process of situation identification illustrated with a uses case and possible scenarios. To cope with that, we have based on the multi-OCSM ontology that plays a fundamental role in our approach.

## REFERENCES

- [1] Matassa, A., Riboni, D. (2020). Reasoning with smart objects' affordance for personalized behavior monitoring in pervasive information systems. *Knowledge Information System*, 62(4): 1255-1278. <http://dx.doi.org/10.1007/s10115-019-01357-y>
- [2] Pradeep, P., Krishnamoorthy, S. (2019). The MOM of context-aware systems: A survey. *Computer Communication*, 137: 44-69. <http://dx.doi.org/10.1016/j.comcom.2019.02.002>
- [3] James, A.B. (2009). Activities of daily living and instrumental activities of daily living. In: Schell, B.A.B., Gillen, G., & Scaffa, M. (Eds.), *Willard and Spackman's Occupational Therapy* (12th ed). Philadelphia: Lippincott Williams & Wilkins.
- [4] Okeyo, G., Chen, L.M., Wang, H. (2013). An agent-mediated ontology-based approach for composite activity recognition in smart homes. *J. UCS*, 19(17): 2577-2597. <http://dx.doi.org/10.3217/jucs-019-17-2577>
- [5] Da, K., Dalmau, M., Roose, P. (2014). Kalimucho: Middleware for mobile applications. *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, Gyeongju, Korea, pp. 413-419. <https://doi.org/10.1145/2554850.2554883>
- [6] Yus, R., Mena, E., Ilarri, S., Illarramendi, A. (2014). SHERLOCK: Semantic management of location-based services in wireless environments. *Pervasive and Mobile Computing*, 15: 87-99. <http://dx.doi.org/10.1016/j.pmcj.2013.07.018>
- [7] Angsuchotmetee, C., Chbeir, R., Cardinale, Y. (2020). MSSN-Onto: An ontology-based approach for flexible event processing in multimedia sensor networks. *Future Generation Computer Systems*, 108: 1140-1158. <http://dx.doi.org/10.1016/j.future.2018.01.044>
- [8] Compton, M. (2017). The SSN ontology of the W3C semantic sensor network incubator group. *Web Semantics: Science, Services and Agents on the World Wide Web*, 17: 25-32. <http://dx.doi.org/10.2139/ssrn.3198991>
- [9] Mansour, E., Chbeir, R., Arnould, P. (2019). HSSN: An ontology for hybrid semantic sensor networks. In *Proceedings of the 23rd International Database Applications & Engineering Symposium*, pp. 1-10.

- <http://dx.doi.org/10.1145/3331076.3331102>
- [10] Zografistou, D. (2012). Support for context-aware healthcare in ambient assisted living. Master's thesis, Computer Science Department, University of Crete.
- [11] Alirezaie, M., Renoux, J., Köckemann, U., Kristoffersson, A., Karlsson, L., Blomqvist, E., Tsiftes, N., Voigt, T., Loutfi, A. (2017). An ontology-based context-aware system for smart homes: E-care@ home. *Sensors*, 17(7): 1586. <http://dx.doi.org/10.3390/s17071586>
- [12] Bermudez-Edo, M., Elsaleh, T., Barnaghi, P., Taylor, K. (2017). IoT-Lite: a lightweight semantic model for the internet of things and its use with dynamic semantics. *Personal and Ubiquitous Computing*, 21(3): 475-487. <http://dx.doi.org/10.1007/s00779-017-1010-8>
- [13] Ramírez, J.M.N., Roose, P., Dalmau, M., Cardinale, Y. (2018). An event detection framework for the representation of the AGGIR variables. 2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Limassol, Cyprus. <https://doi.org/10.1109/wimob.2018.8589105>
- [14] OGC / Semantic Sensor Network W3C Incubator Group. Semantic Sensor Network (SSN). <https://www.w3.org/TR/vocab-ssn/>, accessed on Sep. 5, 2020.
- [15] DOLCE Ultra-Light (DUL) Ontology. <http://www.ontologydesignpatterns.org/ont/dul/DUL.ol>, accessed on Sep. 5, 2020.
- [16] Kamalendu, Pal. (2018). Ontology-based web service architecture for retail supply chain management. *Procedia Computer Science*, 130: 985-990. <https://doi.org/10.1016/j.procs.2018.04.101>
- [17] Cédric, D., Laborie, S., Roose, P. (2013). A semantic generic profile for multimedia document adaptation. *Intelligent Multimedia Technologies for Networking Applications: Techniques and Tools*, 225-246. <http://dx.doi.org/10.4018/978-1-4666-2833-5.ch009>
- [18] Gull, K., Sudip, P., Jain, S. (2019). A comparative analysis of lexical/NLP method with WEKA's bayes classifier. *International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC)*, 5(2): 221-227. <http://dx.doi.org/10.11591/ijece.v9i3>
- [19] Protégé Editor URL: <https://protege.stanford.edu/>, accessed on Sep. 5, 2020.
- [20] Baader, F., Horrocks, I., Lutz, C., Sattler, U. (2017). *Introduction to Description Logic*. Cambridge University Press. <http://dx.doi.org/10.1017/9781139025355>
- [21] Cardoso, J., Pinto, A.M. (2015). The web ontology language (OWL) and its applications. In *Encyclopedia of Information Science and Technology*, Third Edition, IGI Global, pp. 7662-7673. <http://dx.doi.org/10.4018/978-1-4666-5888-2.ch755>
- [22] De Giacomo, G., Oriol, X., Rosati, R., Savo, D.F. (2016). Updating DL-Lite ontologies through first-order queries. In *International Semantic Web Conference*, pp. 167-183. [http://dx.doi.org/10.1007/978-3-319-46523-4\\_1](http://dx.doi.org/10.1007/978-3-319-46523-4_1)