Applied Computing and Informatics 14 (2018) 202–213

Contents lists available at ScienceDirect

Applied Computing and Informatics

journal homepage: www.sciencedirect.com

Original Article

CAMeOnto: Context awareness meta ontology modeling

Jose Aguilar *, Marxjhony Jerez, Taniana Rodríguez

CEMISID, Dpto. de Computación, Facultad de Ingeniería, Universidad de Los Andes, Mérida, Venezuela

ARTICLE INFO

Article history:
Received 12 May 2017
Revised 17 July 2017
Accepted 13 August 2017
Available online 22 August 2017

Keywords:
Context-awareness
Ontologies
Meta ontologies
Context modeling

ABSTRACT

In order to model a context and adapt it to any domain, it is necessary an ontology that captures generic concepts to a higher level. The context model must provide mechanisms to extend the specific information of a context in a hierarchical manner. In this paper, we propose CAMeOnto, an ontology with these characteristics, based on the principles of 5Ws: who, when, what, where and why. CAMeOnto is used by CARMiCLOC, a reflective middleware for context-aware applications, and is instantiated in several case studies, in order to test how CAMeOnto works correctly and can reason to infer information about the context.

1. Introduction

In a smart environment, users want to be automatically provided with services. For that, it is required an appropriate contextual information: user's situation, location, time, devices existing in the environment, among other things. Nowadays, there are several domains where it is required the modeling of the context, such as: Internet of Things (IoT), Internet of Everything (IoE), Virtual Reality (VR), Recommendation Systems, and Intelligent Transport Networks like Vehicular Ad hoc NETworks (VANET).

Different challenges exist in the area of the Context Awareness. For example, in Intelligent Transport Systems (ITS) a VANET component requires the identification of effective context information [1]. In VR, IoT and IoE the main challenges are the privacy and the lacks of modeling of social issues [2,3].

The use of ontologies for expressing context is advantageous because they can express the different characteristics of the context. Examples of context aware modeling based on Ontology are presented in [4,5], which present the emergent concept "Semantic Web of Things" (SWoT) [6,7]. SWoT is the combination of five elements: human (such as users, service providers), machines (such as computer), physical things (such as sensors or devices), abstract things (such as data, information or services), and the working environment (such as the Ad hoc network, the Sensor Network, or the Web).

In [8] was presented CARMiCLOC, a reflective middleware for context-aware applications in the cloud, which offers various services for context-aware systems, among which is found the modeling. The context modeling is proposed using ontologies, because it is a technique that gains force as a standard, due to its semantic expressiveness and interoperability [8]. In this way, in this work we propose the modeling of the context ontology of CARMiCLOC.

Our ontology must be used by the services of CARMiCLOC to represent the context, to reason about it, to be shared, among other things, without the use of sophisticated tools and processes. This ontology is called CAMeOnto, which is an ontology that allows the modeling of the context, so that it can be used by the set of services of CARMiCLOC in order to allow the autonomy of the context-aware applications, so that they can discover it, analyze it, and based on it, make decisions. Our ontology provides a simple context modeling based on the 5Ws (who, when, what, where and why), with sufficient information about the context to reason and learn from it.

The organization of the paper is the following: Section 2 presents the relative works to this research, Section 3 presents the theoretical aspects, the Section 4 presents CAMeOnto, and finally, the Section 5 presents the results and the conclusions.

* Corresponding author at: EPN & UTP, Ecuador.
E-mail addresses: aguilar@ula.ve (J. Aguilar), marxjhony@ula.ve (M. Jerez), taniana@ula.ve (T. Rodríguez).

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.aci.2017.08.001
2210-8327/© 2017 The Authors. Production and hosting by Elsevier B.V. on behalf of King Saud University.

2. Relative works

In this section, we will present the state of the art related to the Context Aware Computing and the Context modeling.

In [9], Alegre presents an investigation about the context-aware systems and their applications, and illustrates how the systems understand the situations, provide services and adaptive their functionalities for very specific needs. Alegre presents different aspects related to the state of the art of Context-Awareness Systems (CASs), for example: 1) the methodologies used for the development of CASs, 2) the challenges and techniques to construct CASs, 3) and the conceptualization of CASs. It also presents the directions and challenges that should be considered in the future researches regarding context-aware computing. Especially, in the Engineering of the context aware systems, he defines the following investigations: principles of design of the Human-Computer Interaction, Architectural Patterns, Paradigms of programming, methodological supports, among others.

With respect to the Context modeling, Perera exposes in [10] that there is not a standard for context modeling. In [9,10], Alegre and Perera, respectively, show the most common techniques used for context modeling. Alegre exposes that each technique has some advantage. For example, the ontology-based modeling allows the semantic reasoning, the expressiveness in the context representation, a strong validation, context sharing and the independent utilization.

There are several works that have developed the modeling of context through ontologies [11,12,19]. The use of ontology allows, according to [11,4]: 1) the reasoning using a very well defined declarative semantic, 2) the knowledge sharing in dynamic systems and, 3) the interoperability. In [11], Guermah implements a framework of context-aware services. Guermah propose an ontology, called CONON (CONtext ONtology), which has two levels. The first level is the services platform, with the general representation of the context; and the second level is extensible and allows adding other specific ontologies. Finally, the framework allows the adaptation of services based on the WSMO (Web Service Modeling Ontology) [13]. They define the next set of classes: user, activity, service, device, and environment.

In [14], Skillen shows a user-centered ontology, and describes an ontological modeling of the user profile for the customization of context-aware applications, defining a method for capturing and representing user profiles for changing environments. The classes that he defines are: user, time, context, activity and location. In [12], Naqvi presents a mobile context awareness approach in a cloud, and an ontology where its main entities are the user, the platform and the service that is instantiated. It is deployed on a middleware that is defined as a Platform as a Service (PaaS).

In [5], Zhongu et al. present a Meta Context Ontology Model (MCOnt), where the context is divided in three categories: the internal context (it describes the user personal information, preferences and emotional states, provided directly from the user), the external context (it is the user-physical and environmental context) and the boundary context (it represents the information related to users' task-related activities and services). They use a three layer hierarchical approach to describe this meta-ontology in [4]. The MCOnt have three advantages: 1. Ontologies can be pragmatically organized in a modular manner, the modularity facilitates the maintaining and updating of the context ontology model. 2. The multi-dimensional submodels strengthen the semantic integrity of the whole context ontology model. 3. Each part of the internal model in MCOnt is a higher concept abstraction for a generic context. Nevertheless, this work lacks of the definition of the relation between user situation and the services.

In general, the models that have been presented in the previous works are for specific domains, and are not oriented to be developed as services. These ontologies have not been defined for an environment on which services can be executed about the context, for applications that require it. In particular, the ontology proposed in this article allows the autonomous behavior of context-aware applications, such that they can reason about the context, discover the context, among other things. In this sense, it is a general ontology, which allows the modeling of any environment. Our ontology is of general domain, to be used by CARMiCLOC [8,17], which is part of its knowledge base, in a way that can be used as a knowledge service.

3. Theoretical framework

3.1. Context-aware system (CAS)

In this section, we present several definitions:

Context: Dey expresses that for the computation [15]: "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application."

Context Aware: According to Dey, a context-aware system "provides relevant information and/or services to the user, where the relevance depends on the user's tasks".

Interaction with a Context Awareness System: it can be defined in the executions and configurations. The execution refers to the actions/behaviors of the system in a specific situation (e.g. silence the phone when someone is sleeping). The second refers to the adjustment of actions/behaviors that the system may have. These interactions can be active or passive.

Context Information Life Cycle: Contexts have a life cycle, which defines where data are generated and where are consumed. In this life cycle, context awareness can be seen as a service, called by some authors as Context-as-a-Service (CXaaS) (see Fig. 1) [9]:

- Acquisition of context. The techniques used to acquire the context can be varied, based on who is responsible, the frequency, the source, the process of acquisition, the type of sensor, among other things.
- Context Modeling. The context modeling is also widely referred to as context representation. Context models can be static or dynamic. In addition, they may have several states [2]: ready, running, suspending, resuming, expired, and terminated. There are different modeling techniques which are described in [1].

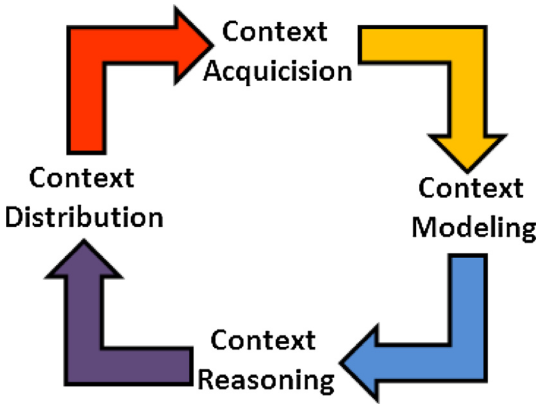


Fig. 1. Context life cycle [9].

- Reasoning of the Context. The reasoning of the Context can be defined as a method to deduce new knowledge from the context. It can also be defined as a process for deriving high-level contexts from a set of contexts. The reasoning requires to solve two characteristics of the context: the imperfection (i.e., it is unknown, ambiguous, imprecise or erroneous) and the uncertainty.
- Distribution of the Context. The two methods used for context distribution are: a) by inquiry. The user requests the context, such that the context management system answers to that query. b) by Subscription, also called publication.

3.2. CARMiCLOC

CARMiCLOC is a context-aware reflective middleware, based on the autonomic computing. In general, CARMiCLOC provides accurate services that are required for the management of the context life cycle based on the classical model of autonomic computing, which consist in a MAPE-K loop [28], with a set of services for Monitoring the context, a set of services for Analyzing the context, a set of services to plan actions on the Context-Aware Applications, and a set of services to execute these actions, but these services require a knowledge database (K) that is presented in this work as an ontology (see Fig. 2).

Specifically, CARMiCLOC incorporates seven services for the context management, the services can be provided internally (I) (required by the same CARMiCLOC services) or called independently from external services (E). CARMiCLOC Services may be consumed by Context-aware Applications and Not Context-Aware Applications. Table 1 shows the services provided by CARMiCLOC.

Fig. 2 shows the MAPE architecture of CARMiCLOC. At the bottom of Fig. 2 is the Middleware Base level, where the context and sensors are deployed. The context is discovered using the Se1 service (M), which filters and preconfigures the data. The Se2 Service creates the model of context (A), which is stored in the Knowledge base (K) that will be accessed by the MAPE cycle. Se3 Service allows the reasoning over K, in order to infer information about the context (P). The rest of the services (Se4, Se5, Se7, Se8) are specific services about the quality, security, among other things, of the context, offered by CARMiCLOC.

Table 1
CARMiCLOC services.

ID	Service offered	Intern/ extern	Description
SE_1	Context acquisition/pre configuration	I, E (B)	Service in charge of the aggregation and filtering of data, the discovery and annotation of the context
SE_2	Context modeling	I, E	Service in charge of knowledge management (Model), data storage, context sharing and configuration at run time
SE_3	Context reasoning	I, E	Service in charge of management and inference of knowledge, event management and data fusion
SE_4	Context distribution	E(A)	Service in charge of the dissemination of the context and services
SE_5	Quality of context (QoC)	I, E(B)	Service in charge of the validation and resolution of the conflicts of the context
SE_6	Context security	I	Service in charge of the security and privacy of the user, application and data
SE_7	Context service management		Service in charge of the management of services of the context

One important aspect of CARMiCLOC is that it allows the context switching using the services that it has implemented. When something changes in the base level, the Se1, Se3 and Se2 services work together to produce the new context. Particularly, the context modeling service (Se2) is one of the most important because it allows the maintenance of context information and can create new models based on new contexts acquired.

4. Meta-ontology for the context awareness

For the modeling of context, there are several techniques, however, what takes more value and is being positioned as a standard is the Ontology-Based Context Modeling. The design of our ontology is based on the 5Ws (who, when, what, where and why), which are considered in six contextual classes presented below.

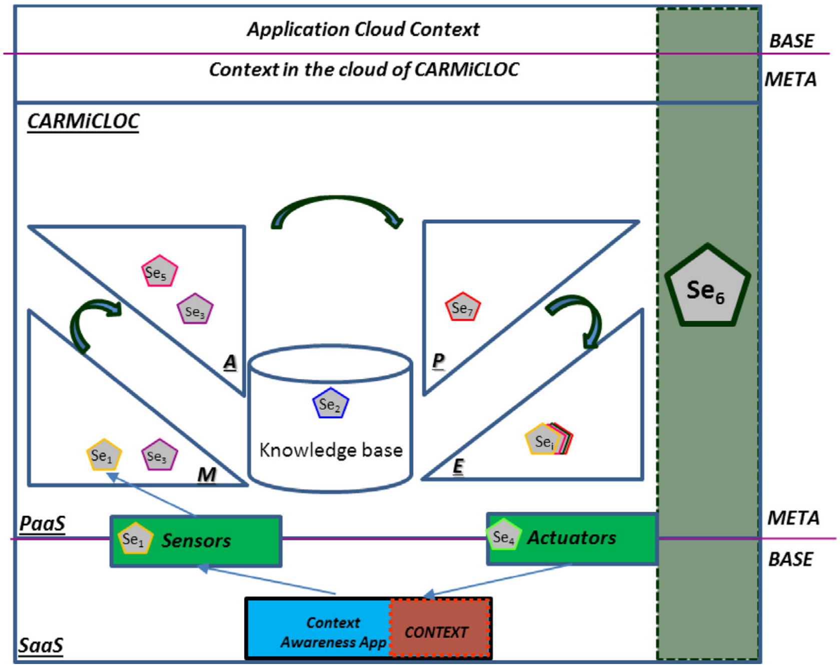


Fig. 2. CARMiCLOC MAPE architecture.

To develop our ontology, we are used the Neon methodology [18], which is a methodology to build ontologies based on the reutilization and the dynamic evolution of ontology networks. Particularly, CAMeOnto is based on the CONON [11] and MAOnt [5] ontologies. In specific, we propose a hierarchy of ontologies similar to the proposition in [5] (see Fig. 3).

In a context model, the context information must be defined in terms of attributes, characteristics, context relationships, context quality attributes, among other things. A simple and complete approach is the one used in [9], which contains the four W's that answer the questions Who? (Who is the user?), What? (What I am doing?), Where? (Where is my location?), and When? (When has happened?). Also, in the mobile applications, it is important to know who carry out the action and with whom, and the relationships between the environment and the services that are executed or that could be called [9]. All of them defines the "Why" that in CAMeOnto is added to express the motivation to do the "what".

In [5], the context is categorized in three: internal context, external context and boundary context. We are going to use the same approach. In specific, our context ontology is characterized by two hierarchical levels (see Fig. 7). The first one is a general and independent domain ontology, and the second one allows the addition of specific domain context ontologies. The first level is divided in three groups, inspired into the proposition in [5]. These groups have six contextual classes, which are:

- ***User:*** information about the user's profile, situation, and preferences (internal context).
- ***Activity:*** describes the different sets of activities that can be developed in the context (external context)
- ***Time:*** describes the notion of time in the context, which can be used to define the chronological situations in the context (external context).
- ***Device***: describes the device of software and hardware in the context (external context)
- ***Services:*** Describes the characteristics of the service required by the user, Quality of Service (boundary context)
- ***Location***: describes the location of the context, its indoor and outdoor space, and the property of the environment (external context).

Our ontology follows the 5Ws (who, when, what, where and why) because the "user" class defines who, the "time" class defines when, the "location" class defines where, the "services" class defines the what, and with the "device" and "activity" classes is defined the why. Fig. 4 gives an example of instantiation of our ontology using our six classes, with information that could be described in each one. For example, in the "location" could be described the environment, which could have as attributes the temperature and illumination.

The relationships between the classes in our multidimensional models is shown in Fig. 5. The relationship between the user and its role in a domain, makes possible to find the why of an activity: the motivation. To complete the description of the motivation, it is required the relationship between the user and the devices, and the user/role and a given activity (see Fig. 5).

The relationships of our ontology allow describing the context: the environment has a location, and devices that provide services according to their specific description. Also, the devices are used by the users at this location, whom have roles in the activities carried out in the environment in a given moment. Now, we specify each class of our ontology.

4.1. User model

In a context aware, the context information is used to respond and adapt to users' behaviors. In [20] is shown an ontology to represent and capture the user profiles within a changing environment. The user model is a subclass of the context. We use this user profile ontology mixed with the ontology proposed in [5]. The user model is shown in Fig. 6. The user has a user profile, a schedule, and a role. The user profile describes the user's personal information, the user schedule defines the events which he/she reacts, and the role defines the responsibility of a user in a given moment.

According to Fig. 6, the user profile describes the user's personal information, such as his/her name, social safety number, and preferences. Also, it can be included an extension of user profiles for each domain will include, for example, health profile, professional profile, sporty profile. A series of user-related events compose the user schedule, where each event contains one or more activities, and each activity contains one or more tasks. The role class indicates the current state in which the user's action is situated, such as professor, student or guest. Therein, the role class mainly represents the lawful situation set of a user or an agent, which can be beforehand defined.

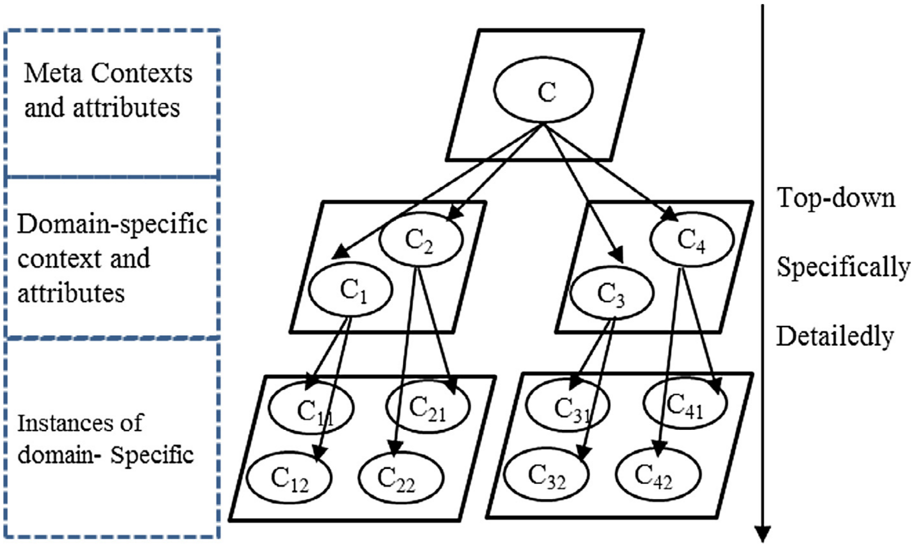


Fig. 3. Hierarchical design of a context ontology [5].

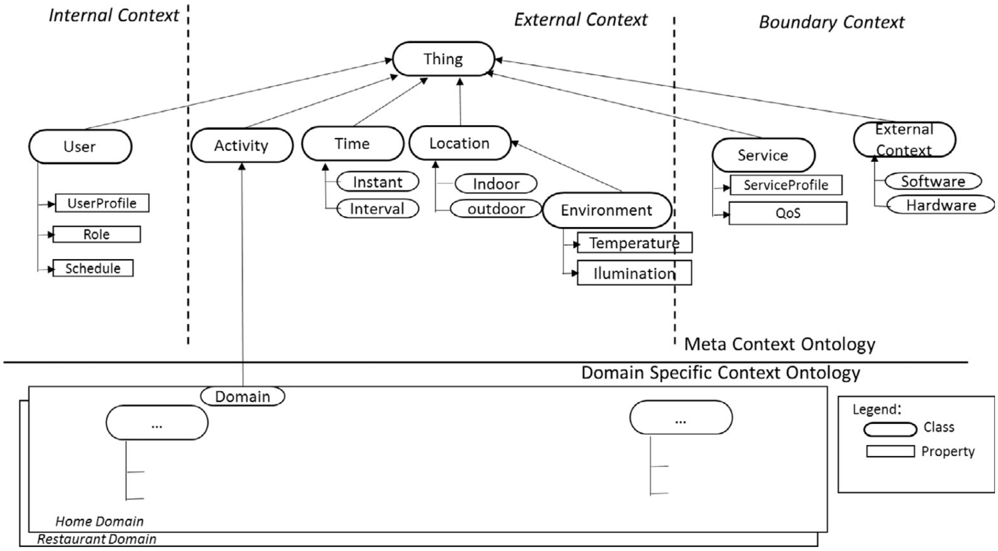


Fig. 4. Hierarchy of our context ontology.

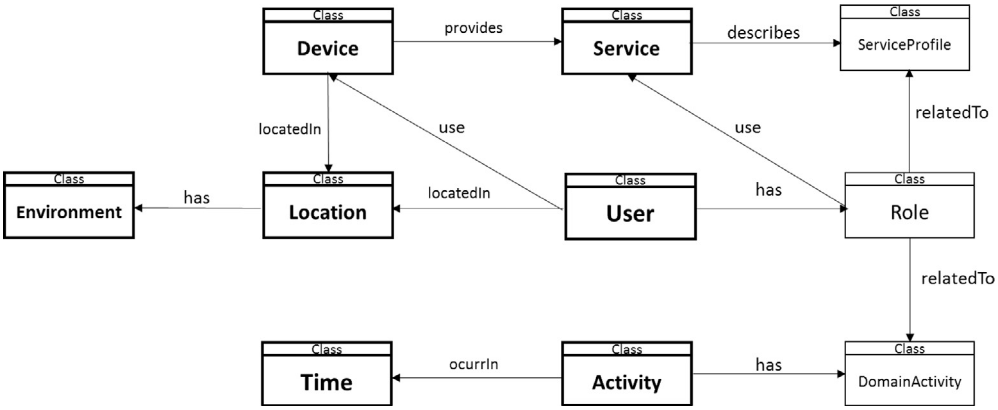


Fig. 5. Relationships between our classes.

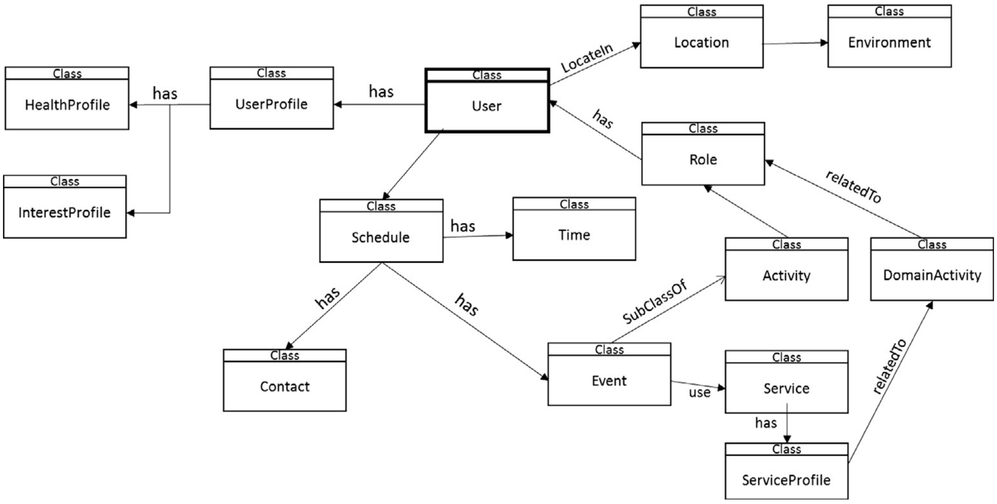


Fig. 6. User model.

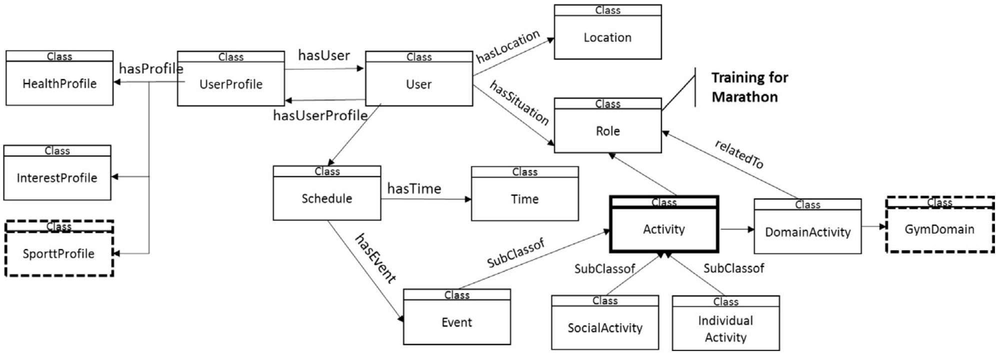


Fig. 7. Activity model.

4.2. Activity model

We based our model in [21,23], where they present an Activity Ontology. The activity model defines he different features and abilities than are required in an activity. The activity model has the next subclasses: Domain of the Activity that describes all the relative to the domain where the activity is involved, the events generated or used during the activity, and the social and individual activities executed in it (see Fig. 7).

Fig. 7 shows an instantiation of the Activity Model, where the domain activity is related to "Gym Domain", the role performed by the user is "Training for the Marathon", and the User profile is "Sport-profile".

4.3. Time model

The time model is the representation about how the time is immersed in the context awareness. This model is shown in Fig. 8, and it's based on the "Time" Ontology proposed in [22].

4.4. Device model

The device model is shown in Fig. 9. A device has a profile (its characteristics) and current status (running, fault, etc.), and can be of software or hardware

4.5. Service model

The service model is shown in Fig. 10. A service is defined by a type (e.g. a web service), and has a profile (it defines its characteristics). Additionally, the service can have a quality of service (QoS) to offer.

4.6. Location model

The location model is shown in Fig. 11, and describes the site where is the environment. The location describes its outdoor and indoor components, its position, among other things.

After all the key classes have been defined within the ontology, specific object and data properties are defined in these classes, which are shown in Table 2. These properties are used to relate one class to another via objects, or specify relations via datatypes. They have an important role to relate key concepts and to infer new information. For example, the "Userprofile" class could contain a property "hasUserProfile", in order to determine the state of a user's profile, and this could be linked to the "DomainActivity" class. Therefore, this could determine that a user has a particular activity, such as "marathon training", and so, infer that there are kilometers to run today.

In this way, the relationships between different users can be modeled, and our middleware can learn to adapt to different user patterns. Fig. 12 presents an example of the hierarchical view of the classes of our ontology, their object properties and datatype properties, displayed in Protégé.

5. Case studies

To test the ontology described in section IV, we are going to consider several case studies.

5.1. Smart classroom

This first case is the Smart Classroom presented in [16], which has a middleware to support an Intelligence Learning Environment,

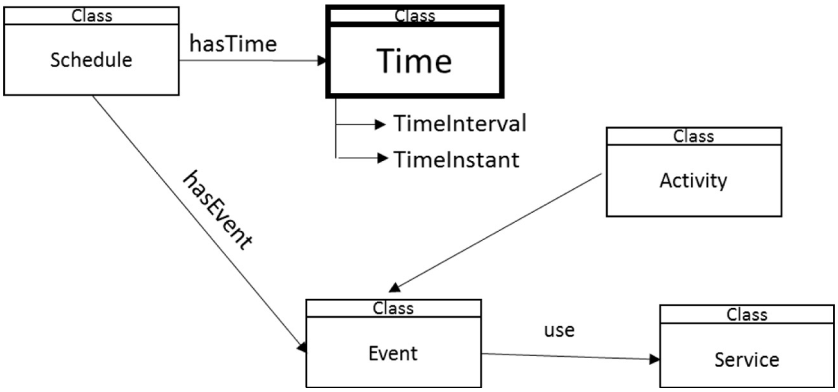


Fig. 8. Time model.

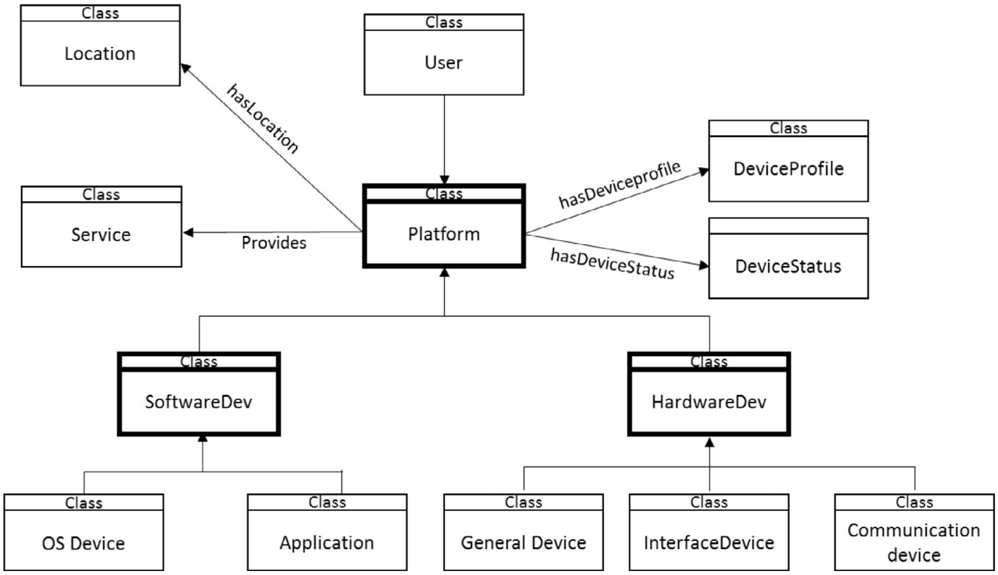


Fig. 9. Device model.

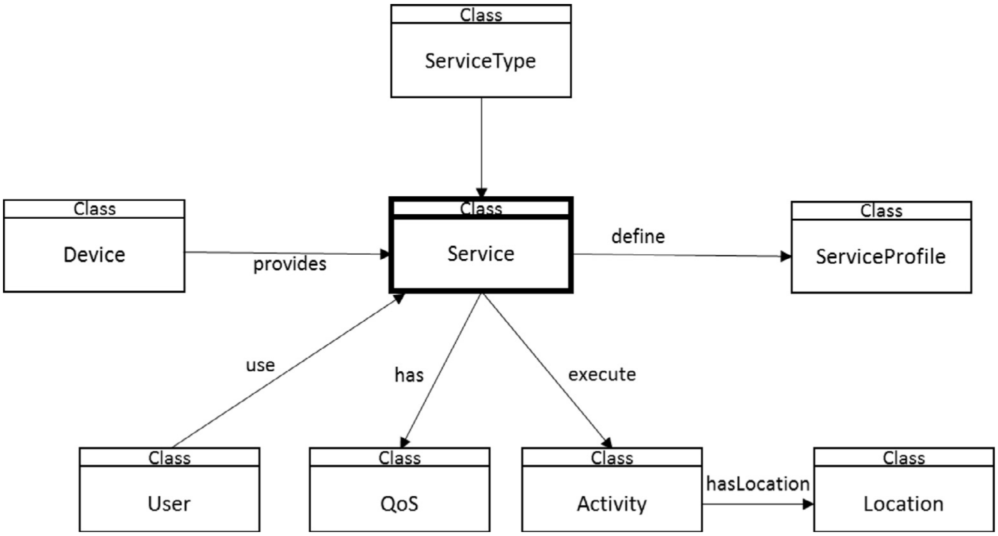


Fig. 10. Service model.

called IECL [24,25]. The instantiation of the case study is shown in Fig. 13.

Now, we describe the instantiation of our ontology in this case study. The main classes are:

- Users: students, professors, guests.
- Location: the place where is the smart classroom, and conditions (temperature, light, etc.).
- Activity: Course
- Time: the date of the course.
- Device: the different resources in the smart classroom (RDF sensors, smart-board, book, repository database, etc.).
- Services: Describes the characteristics of the services provided in the smart classroom, such as the prediction of the results of the students, among others. The set of services has been proposed in [29].

In Table 3 is shown an example of the instantiation of CAMeOnto for the Smart classroom.

Event class can have different instantiations. For example, suppose the next instantiation of this class:

Event (Notify Users): This event can notify to users by email, twitter, etc., according to the user profiles and environment devices, e.g., the beginning of the course. Table 4 shows an example of the instances of the classes for this event, which can use the rules of inference of our ontology.

The rules for the “NotifyUser” event are the following, in the case that the course starts at 8am on the day that it is invoked.

1. `Person(?p) ^ isEnrolled(?p, ?c) ^ Courser(?c) ^ startTime(?c, 8) -> NotifyUsers(?p)`
2. `Person(?p) ^ isTutorCourse(?p, ?c) ^ Courser(?c) ^ startTime(?c, 8) -> NotifyUsers(?p)`

These rules are described using SWRL (Semantic Web Rule Language), which is a language for the Semantic Web that can be used to express the rules [27]. Once the rules are executed, we suppose,

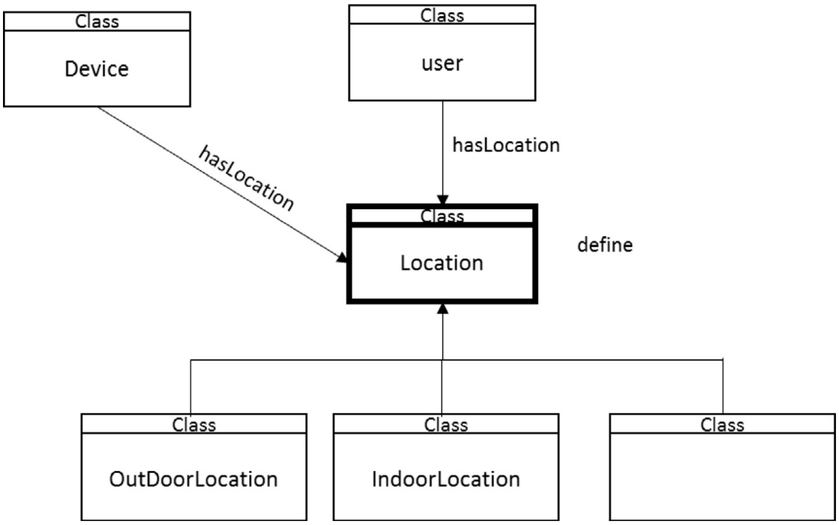


Fig. 11. Location model.

Table 2
Description of the object and DataType properties in CAMeOnto.

object property	description
hasUserProfiel (isuserProfileOf)	Every user has one user profile
hasActivity	Every user perform some activity at one point
hasSituation	Every user has a situation
hasEvent	Every schedule has an event
hasLocation	Every user has a location
Data property	description
hasName	Every user has a name
timeInstant	Every event has a timeInstant
timeInterval (timeStart, timeEnd)	Every schedule has a timeInterval
Address	Every location has an address

for example, that two students and the tutor are notified, as is shown in Fig. 14.

Therefore, NotifyUsers is a subclass of event, and it has notified Student1 according to rule 1, which defines that the student is a person and is enrolled in course 1 that begins the class at 8am, as is shown in Fig. 15.

Also, for the preparation of the class, different activities can be carried out. For example, the room lights must be on (see event in Fig. 16).

The following rule is executed:

```
Courser(?c) ^ startTime(?c, 8) ^ Room(?r) ^
IsDictatedIn(?c, ?r) -> turnLights(?r, true)
```

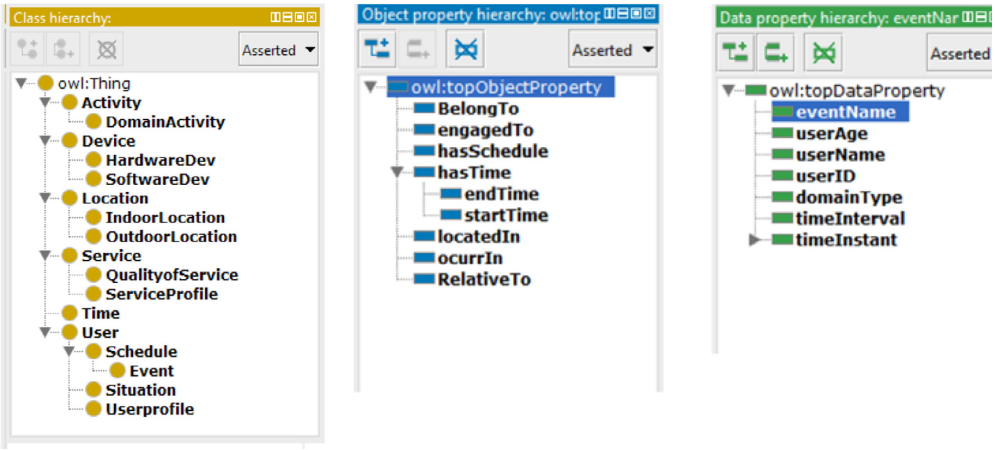
Similar rules must be executed for the air on. Once the lights and air conditioning are turned on, the "EnviromentRoomOk" event is triggered (see Fig. 17).

In this case, the following rule is fulfilled:

```
Courser(?c) ^ startTime(?c, 8) ^ Room(?r) ^ turn-
Lights(?r, true) ^ turnOnAire(?r, true) ^ roomTem-
perature(?r, ?temp) ^ swrlb:greaterThan(?temp, 21) ^
swrlb:lessThan(?temp, 23) -> RoomEnviromentOK(?r)
```

Therefore, the classroom 123 meets all the conditions of the environment to start the class. Similar rules are designed for all the devices, resources, etc., required to start the class.

Once verified all the conditions of the environment, etc., the smart classroom starts with the authentication of the users. Also, our ontology can be used to authenticate the students, guests and teachers. If the STUDENT X is authorized to access the class, then he/she can obtain the learning resources, books, etc. of the class. The ontology can use the next rules triggered by the "StudentMaterial" event.


Class hierarchy:
Asserted
Object property hierarchy: owl:top
Asserted
Data property hierarchy: eventNar
Asserted

- owl:Thing
  - Activity
    - DomainActivity
  - Device
    - HardwareDev
    - SoftwareDev
  - Location
    - IndoorLocation
    - OutdoorLocation
  - Service
    - QualityofService
    - ServiceProfile
  - Time
  - User
    - Schedule
      - Event
    - Situation
    - Userprofile

- owl:topObjectProperty
  - BelongTo
  - engagedTo
  - hasSchedule
  - hasTime
    - endTime
    - startTime
  - locatedIn
  - ocurrIn
  - RelativeTo

- owl:topDataProperty
  - eventName
  - userAge
  - userName
  - userID
  - domainType
  - timeInterval
  - timeInstant

Fig. 12. CAMeOnto classes, object properties and datatype properties.

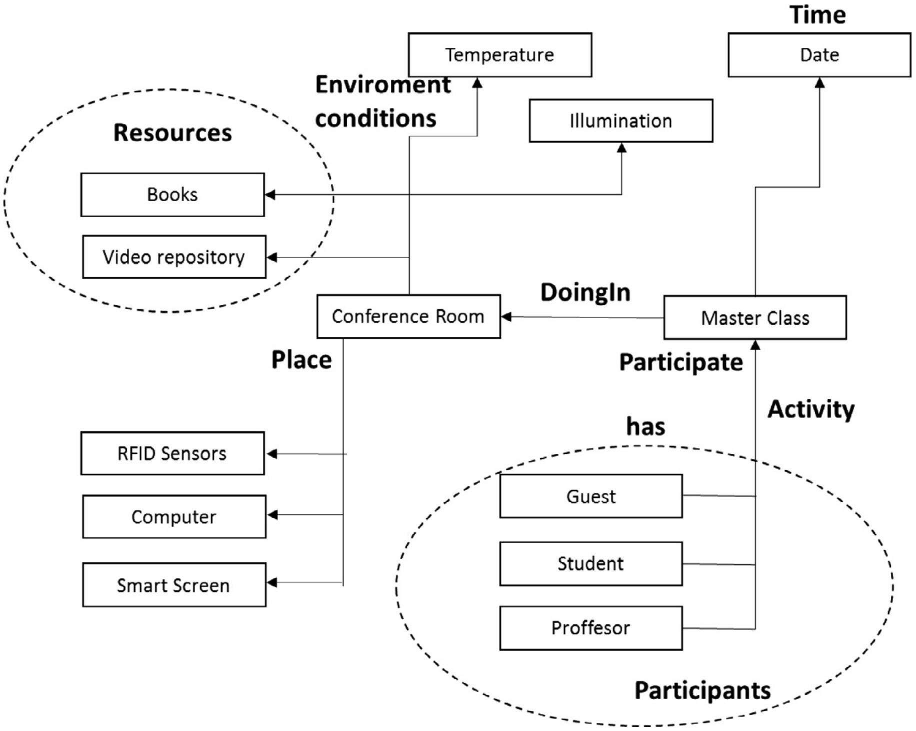


Fig. 13. Smart class instantiation.

Table 3
Example of instantiation.

CAMeOnto	Smart classroom
User	Guests, students, professors
Event	Master course
Location (Indoor)	Conference room
Location	Temperature, illumination
Time: TimeInterval	timeStart, timeEnd
DomainActivity	Programming course ontology (didactics resources, etc.)
DidacticsResources	Books, video repository
HardwareDevice	RFID sensors, desktop, smart screen

Table 4
Notify users.

CAMeOnto	Smart classroom
User	Guests, students, professors
Event	Notify users
Time	Start time
Domain activity	Programming course

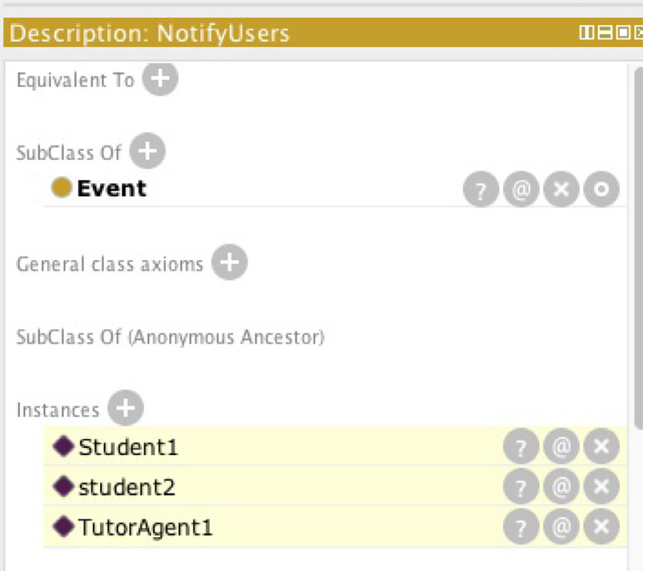
With the following rule, the book resource is activated for the students who are enrolled in the course

```
Person(?p) ^ isEnrolled(?p, ?c) ^ Courser(?c) ^
startTime(?c, 8) -> bookResource(?c, true),
```

Similar rules are also defined for each one of the resources in the smart classroom.

5.2. Surgery room

Now, suppose we are going to instantiate our ontology in a surgery. It is an elaborate process in progressive stages. The "Perioperative" term generally refers to the three phases of surgery, namely preoperative, intraoperative, and postoperative. A Context awareness perioperative system can be modelled using CAMeOnto. Fig. 18 shows the system architecture presented in [23].



Description: NotifyUsers

Equivalent To

SubClass Of

Event

General class axioms

SubClass Of (Anonymous Ancestor)

Instances

Student1

student2

TutorAgent1

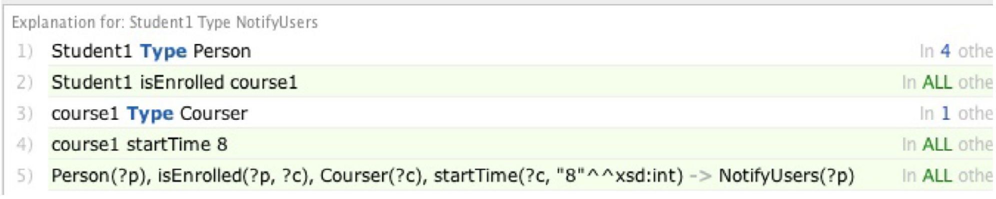
Fig. 14. "Notify User" Event on Protégé.

Table 5 shows how the elements of the perioperative process are introduced in CAMeOnto.

Now, we can describe the different situations in this environment using CAMeOnto

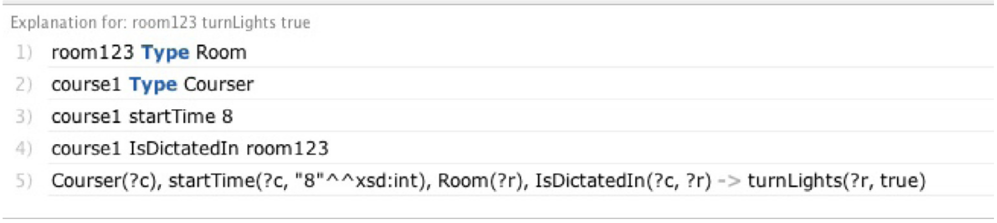
Situation 1. Locate the surgery room where the patient will be operated.

In this case, the patient information will be displayed on the computer screen that is on the door of the operation, using the next rule:



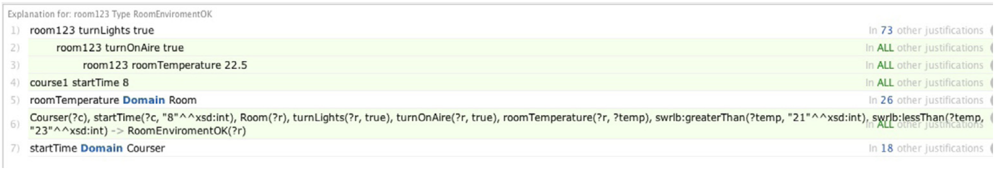
```
Explanation for: Student1 Type NotifyUsers
1) Student1 Type Person                                                              In 4 othe
2) Student1 isEnrolled course1                                                       In ALL othe
3) course1 Type Courser                                                              In 1 othe
4) course1 startTime 8                                                               In ALL othe
5) Person(?p), isEnrolled(?p, ?c), Courser(?c), startTime(?c, "8"^^xsd:int) -> NotifyUsers(?p)   In ALL othe
```

Fig. 15. Explanation of the Student 1 in the "NotifyUser" event.



```
Explanation for: room123 turnLights true
1) room123 Type Room
2) course1 Type Courser
3) course1 startTime 8
4) course1 IsDictatedIn room123
5) Courser(?c), startTime(?c, "8"^^xsd:int), Room(?r), IsDictatedIn(?c, ?r) -> turnLights(?r, true)
```

Fig. 16. "TurnLight true" event.



```
Explanation for: room123 Type RoomEnviromentOK
1) room123 turnLights true                                                           In 73 other justifications
2)     room123 turnOnAire true                                                       In ALL other justifications
3)         room123 roomTemperature 22.5                                              In ALL other justifications
4) course1 startTime 8                                                               In ALL other justifications
5) roomTemperature Domain Room                                                       In 26 other justifications
6) Courser(?c), startTime(?c, "8"^^xsd:int), Room(?r), turnLights(?r, true), turnOnAire(?r, true), roomTemperature(?r, ?temp), swrlb:greaterThan(?temp, "21"^^xsd:int), swrlb:lessThan(?temp,
   "23"^^xsd:int) -> RoomEnviromentOK(?r)                                            In ALL other justifications
7) startTime Domain Courser                                                          In 18 other justifications
```

Fig. 17. The rule about the room is OK.

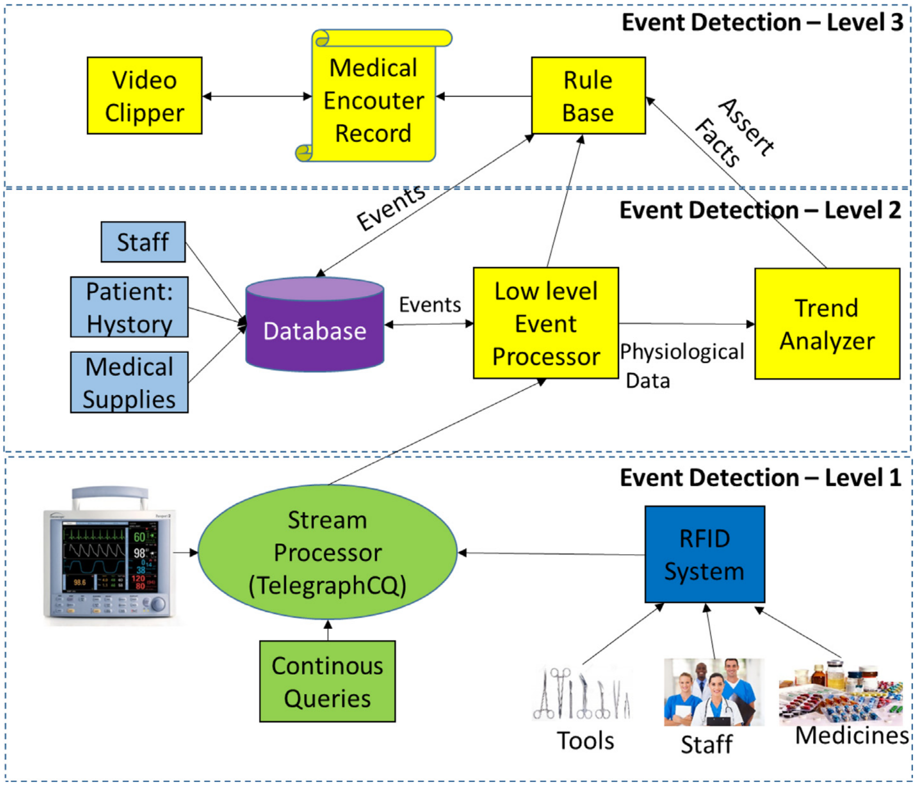


Fig. 18. System architecture of a surgery room [23].

Table 5
Surgery room instance.

CAMeOnto	Operational room
User	Staff (Nurses, MD), patient Patient: Health Profile (history medicine Allergy…)
Event	Hearth surgery
Location (Indoor)	Operation room 2
Location (Outdoor)	John Hospital 1234 Av. Richmond,…
Environment	Temperature, illumination
Time: TimeInterval	timeStart, timeend
Domain activity	Heath: surgery Room: medical equipment, medicine pharmacy list, specialist medical, list….
HardwareDevices	RFID sensors, life monitors

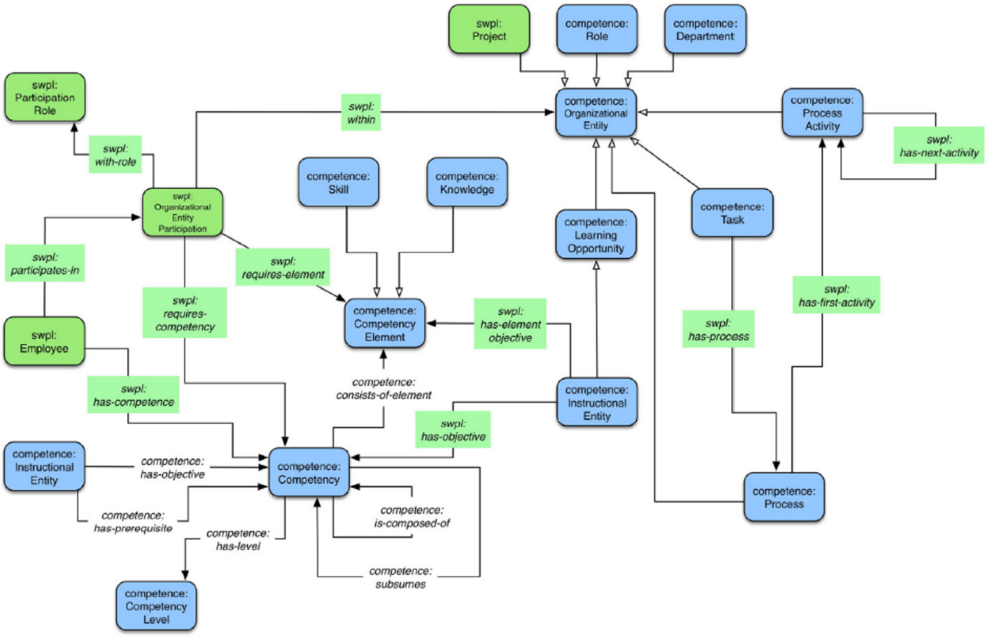


Fig. 19. SWOLF. Ontology based model.

```
Patient(?p) ^ SurgeryRoom(?sr) ^ Doctor(?d) ^
reserveSurgeryRoom(?d,?sr) ^ isPatient (?p,?d) ^
operationType(?p,"heart opertation") ^ startSurgery
(?sr,8) -> UserProfile(?p).
```

Situation 2. The patient is in a critical situation in the operation

If the patient has a temperature greater than 40 degrees, then the condition of the patient is critical. In this case, it will trigger the "PatientCondicion" event, to see the condition of the patient. The next rule is applied:

```
Parient(?p)^hasTemperature(?p,?t)^swrl:greater
Than(?t,40)->PatientCondition(?p,critica)
```

In this way, the system searches for which drug should be applied according to the patient's health profile, and indicates which medication should be applied to the patient.

5.3. Semantic workplace learning framework

In this section, we present an additional test for CAMeOnto. In this case, we model the S-WOLF (Semantic Workplace Learning Framework) framework proposed in [26], which exploits semantic technologies to model, represent and share knowledge within organizations. This framework allows the description of a wide range of workplace learning experiences in an organization. In this subsection, we present the instantiation of CAMeOnto in this case study.

Table 6
S-WOLF modeled with CAMeOnto.

CAMeOnto	S-wolf
User	Employee: role participant
Activity	Learning
Domain	Learning
Time	It is not defined in S-WOLF
Location	Organization
Environment	It is not defined in S-WOLF
Service	Learning service
External context	System used in an organization

Fig. 19 presents the core ontology of S-WOLF, and Table 6 presents how S-WOLF is mapped using CAMeOnto, such that each concept in S-WOLF is introduced in CAMeOnto. Particularly, the "user", "location", "activity" and "services" classes of CAMeOnto can be used to model S-WOLF. The "Time" and the details about the "environment" classes of CAMeOnto are not considered in the S-WOLF ontology but can be considered if is required later.

In this case, we have shown how can be used our ontology to model a specific context.

6. Conclusions

In this article, we have proposed a Meta ontology for the context modeling, called CAMeOnto. It is a general ontology, without being biased to a single application domain. We have tested in different contexts, for example, in the context of a smart classroom or in a Surgery Room, and in each case our ontology can model the context. We were able to create the relationships between the activity domains, the situation the user is going through, among other things.

This meta-ontology describes the main components of knowledge to be considered in a CAS, which can be extended with the specific information about the domain of application in its second level. Our ontology allows the modeling of the context based on the 5Ws, using a set of models for each dimension. In this way, we describe the main details of a context.

In future works, we are going to test the creation and integration of domain-specific ontologies, in order to observe how the proposed ontology could be adapted in real time. Also, CARMiCLOC has a set of services (S21 and Se2) to automatically populate our ontology, in order to avoid a human-based population process of the knowledge base. Currently, we are working on the implementation of these services using an approach that analyses the exchanges of messages in an environment using DPI (Deep Packet Inspection) techniques, in order to discover information of the context in order to instantiate CAMEOnto.

In the case studies, the Pellet reasoner is used, nevertheless it is just a test. In future works, we are going to evaluate the performance of CAMEOnto into a real environment with large data. Finally, other important work is to test the context switching with CARMiCLOC based on its services, in a way to update CAMEOnto with the new information of the context in realtime.

Acknowledgments

Dr. Aguilar has been partially funded by the Prometheus Project of the Ministry of Higher Education, Science, Technology and Innovation of the Republic of Ecuador.

References

[1] H. Vahdat-Nejad, A. Ramazani, T. Mohammadi, W. Mansoor, A survey on context-aware vehicular network applications, Vehicular Commun. 3 (2016) 43–57.

[2] J. Grubert, T. Langlotz, S. Zollmann, H. Regenbrecht, Towards pervasive augmented reality: Context-awareness in augmented reality, IEEE Trans. Visual Comput. Graphics 23 (6) (2017) 1706–1724.

[3] E. de Matos, L.A. Amaral, F. Hessel, Context-aware systems: technologies and challenges in internet of everything environments, in: J. Batalla, G. Mastorakis, C. Mavromoustakis, E. Pallis (Eds.), Beyond the Internet of Things, Springer, 2017, pp. 1–25.

[4] X.H. Wang, D.Q. Zhang, T. Gu, H.K. Pung, Ontology based context modeling and reasoning using OWL, Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications (2004) 18–22.

[5] L. Zhong-Jun, L. Guan-Yu, P. Ying, A method of meta-context ontology modeling and uncertainty reasoning in SWoT, Proceedings of International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (2016) 128–135.

[6] A. Jara, A. Olivieri, Y. Bocchi, M. Jung, W. Kastner, A. Skarmeta, Semantic web of things: an analysis of the application semantics for the iot moving towards the IoT convergence, Int. J. Web Grid Serv. 10 (2–3) (2014) 244–272.

[7] J. Ye, S. Dasiopoulou, G. Stevenson, G. Meditskos, E. Kontopoulos, I. Kompatsiaris, S. Dobson, Semantic web technologies in pervasive computing: A survey and research roadmap, Pervasive Mobile Comput. 23 (2015) 1–25.

[8] J. Aguilar, M. Jerez, E. Exposito, T. Villemur, CARMiCLOC: Context awareness middleware in cloud computing, in: In Proceedings of Latin American Computing Conference, 2015, pp. 1–10.

[9] U. Alegre, C. Augusto, T. Clark, Engineering context-aware systems and applications: a survey, J. Syst. Softw. 117 (2016) 55–83.

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

[11] H. Guermah, T. Fissaa, H. Hafiddi, M. Nassar, A. Kriouile, An ontology oriented architecture for context aware services adaptation, Int. J. Comput. Sci. 11 (2) (2014).

[12] M. Zaigham, N. NayyabZia, P. Davy, B. Yolande, Cloud computing: a mobile context-awareness perspective, Cloud Computing: Computer Communications and Networks, Springer, 2013, pp. 155–175.

[13] K.L. Skillen, L. Chen, C.D. Nugent, M.P. Donnelly, W. Burns, I. Solheim, Ontological user profile modeling for context-aware application personalization, Ubiquitous Computing and Ambient Intelligence, Springer, 2012, pp. 261–268.

[14] D. Fensel, H. Lausen, A. Polleres, J. de Bruijn, M. Stollberg, D. Roman, J. Domingue, Enabling Semantic Web Services: The Web Service Modeling Ontology, first ed., Springer, 2010.

[15] A. Dey, Understanding and using context, Personal Ubiquitous Comput. 5 (1) (2001) 4–7.

[16] P. Valdiviezo, J. Aguilar, J. Cordero, M. Sánchez, Conceptual design of a smart classroom based on multiagent systems, Proceedings on the International Conference on Artificial Intelligence (ICAI) (2015) 471–475.

[17] J. Aguilar, M. Jerez, M. Mendonça, M. Sánchez, Emergencia ontológica basada en análisis de contexto, como servicio para ambientes inteligentes, DYNA 84 (2017) 28–37.

[18] M. Suárez-Figueroa, A. Gómez-Pérez, M. Fernández-López, The NeOn methodology for ontology engineering, in: Ontology Engineering in a Networked World, Springer, 2012, pp. 9–34.

[19] C. Emmanouilidis, R.A. Koutsiamanis, A. Tasidou, Mobile guides: Taxonomy of architectures, context awareness, technologies and applications, J. Network Comput. Appl. 36 (1) (2013) 103–125.

[20] K. Skillen, L. Chen, C. Nugent, M. Donnelly, W. Burns, I. Solheim, Ontological user profile modeling for context-aware application personalization, in: Proceedings International Conference on Ubiquitous Computing and Ambient Intelligence, Springer, 2012, pp. 261–268.

[21] D. Riboni, C. Bettini, Context-aware activity recognition through a combination of ontological and statistical reasoning, in: Proceeding International Conference on Ubiquitous Intelligence and Computing, Springer, 2009, pp. 39–53.

[22] OWL Time Ontology https://www.w3.org/TR/owl-time/#toc.

[23] S. Agarwal, A. Joshi, T. Finin, Y. Yesha, T. Ganous, A pervasive computing system for the operating room of the future, Mobile Networks Appl. 12 (2–3) (2007) 215–228.

[24] M. Sánchez, J. Aguilar, J. Cordero, P. Valdiviezo, Basic features of a reflective middleware for intelligent learning environment in the cloud (IECL), Proceeding of the Asia-Pacific Conference on Computer Aided System Engineering (APCASE). (2015) 1–5.

[25] M. Sánchez, J. Aguilar, J. Cordero, P. Valdiviezo, A smart learning environment based on cloud learning, Int. J. Adv. Inf. Sci. Technol. 39 (39) (2015) 39–52.

[26] M. Gaeta, V. Loia, F. Orciuoli, P. Ritrovato, S-WOLF: Semantic workplace learning framework, IEEE Trans. Syst., Man, Cybernetics: Syst. 45 (1) (2015) 56–72.

[27] https://www.w3.org/Submission/SWRL/.

[28] J. Vizcarrondo, J. Aguilar, E. Exposito, A. Subias, MAPE-K as a service-oriented architecture, IEEE Latinoamerica Trans. 15 (6) (2017) 1163–1175.

[29] J. Aguilar, J. Cordero, L. Barba, M. Sanchez, P. Valdiviezo, L. Chamba, Learning analytics tasks as services in smart classroom, Universal Access Inf. Soc. J. (2017).