
Hybrid Architecture to Support Context-Aware Systems

Maricela Bravo, José A. Reyes-Ortiz,
Leonardo Sánchez-Martínez and
Roberto A. Alcántara-Ramírez

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/intechopen.69519>

Abstract

Any system that is said to be context-aware is capable of monitoring continuously the surrounding environment, that is, capable of prompt reaction to events and changing conditions of the environment. The main objective of a context-aware system is to be continuously recognizing the state of the environment and the users present, in order to adjust the environment to an ideal state and to provide personalized information and services to users considering the user profile. In this chapter, we describe an architecture that relies on the incorporation of intelligent multi-agent systems (MAS), sensor networks, mobile sensors, actuators, Web services and ontologies. We describe the interaction of these technologies into the architecture aiming at facilitating the construction of context-aware systems.

Keywords: multi-agent system, sensor network, web services, ontologies

1. Introduction

Context-awareness is the characteristic of a system that is capable of monitoring the environment continuously aided with physical sensors and mobile sensors. The goal of a context-aware system is to obtain real data from the context (user preferences, user logs, temperature, humidity, light, etc.) in order to build a multi-valued representation of the context in a particular time, and by means of intelligent processing and reasoning of such acquired data provide relevant information in a timely manner and support for decision-making, considering the physical space conditions. An important aspect of a context-aware system is the capability of internal representation of current context, including the presence of human beings and their profiles.

Three important considerations were to be taken into account during the design of the hybrid architecture reported in this chapter:

- a. *What are the tasks performed within a context-aware system?* Guermah et al. [1] described the challenges of context-aware systems: context capture, context representation, context interpretation and reasoning, service adaptation, context management and context reuse. In response to this question, the proposed hybrid architecture presented in this chapter provides technological support for these tasks to be performed.
- b. *What general concepts does a context cover?* Another important design decision of the architecture was to define the general concepts that constitute a context. According with Abowd et al. [2], context is divided into four classes: location, time, activity and identity. However, in specialized literature, reported context models include more or less concepts. It is out of the scope of this chapter to present a deeper analysis of the concept coverage of context. Instead, we present an extensible and flexible model that allows the amplification or reduction of the concept coverage.
- c. *What are the general functional requirements of a context-aware system?* In Ref. [3], Orsi and Tanca described an overview of the main functional requirements for context-aware systems organized in three aspects:
 - Communication is the capability to adapt content presentation to different channels or devices. Communication also covers the agreement and shared reasoning between users or agents.
 - Situation-awareness refers to the characteristic of modelling location and environment aspects; modelling the user personal situation, and adapting the information to the user needs. One of the most important requirements of personalized service provisioning is the ability to provide the correct information to the correct user in the correct moment.
 - Managing knowledge refers to the task of determining the relevant information and services to be delivered to the users. Abowd et al. [2] also stated that a system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task.

1.1. Hybrid solution approach

In order to attend the afore-mentioned design considerations, **Table 1** shows the technological approaches that were selected to integrate the hybrid solution approach. Current advances of these technologies present significant advantages that contribute to satisfy the complex requirements of any context-aware system. In this sub-section, we briefly describe the technological approaches and their contribution for the tasks and functionalities that should be supported by any context-aware system.

The rapid development of **sensor networks** that deliver network services, enabling remote control, remote supervision and automation of buildings, offices, hospitals, etc.; together with the emergence of new smart mobile devices integrated with sensors, wireless protocols and novel applications, provide the technological foundations to design and build applications that allow continuous **context data capture or acquisition**. Context data come from various information sources: from physical sensors, mobile sensors or from virtual sources such as web pages, logs, public databases, etc. The techniques used to acquire context can vary based on responsibility,

Technological approach	Task contribution	Functionality contribution
Sensor networks	Context acquisition	
Intelligent agents	Context acquisition	Communication support
	Context management	Managing knowledge
	Context interpretation	
Web services	Context acquisition	
	Service adaptation	
	Service provisioning	
Ontologies	Context representation	Situation-awareness
	Context interpretation	Managing knowledge
	Context reasoning	
	Context reuse	

Table 1. Technological approaches that integrate the hybrid solution.

frequency, context source, sensor type and acquisition process [4]. Mobile devices also represent the means by which personalized information and services can be delivered.

Web services are reusable software resources that can be shared, composed and invoked independently of the hardware, operating system and programming language used at the server and client side. In this sense, Web services allow the interoperability between hardware devices, intelligent agents and servers in order to personalize **service adaptation** and **service provisioning**.

According to Jennings and Wooldridge [5], an **intelligent agent** ‘is an encapsulated computer system that is situated in some environment, and that is capable of flexible, autonomous action in that environment in order to meet its design objectives. Of particular interest is the notion of an agent as a solver entity capable of showing flexible problem-solving behaviour. The abilities of individual agents to solve problems and communicate are fundamental to integrate a multi-agent system (MAS). Intelligent agents provide **communication** mechanisms to **control** and **monitor** the entire context-aware architecture. They are also capable of acquiring additional context data by invocation of services, maintain a shared context representation, interpret current state of the context and trigger actions that will adapt or affect the context.

Ontologies are representational models based on description logic, logic programming and frame logic that allow the formal definition of concepts and relations comprising the vocabulary of a topic area as well as the axioms and rules for combining terms and relations to define extensions to the vocabulary [6]. During the last decade, ontologies have gained popularity for context modelling due to their expressiveness and reasoning support. Ontologies allow context representation, context interpretation by explicitly defining equivalences, context reasoning and context reuse.

In order to achieve the afore-mentioned requirements and facilitate the complex interactions that occur inside a context-aware system, in this chapter, we present a **hybrid solution** approach (see **Figure 1**) that leverages current technologies by incorporating a sensor network,

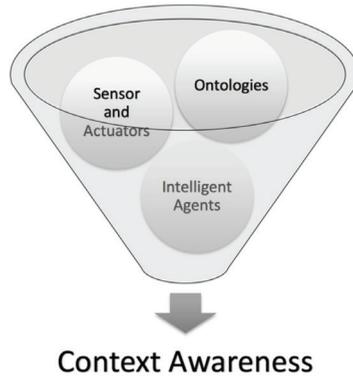


Figure 1. Integration of multiple technologies for context-awareness.

a set of specialized agents, a collection of software components deployed as Web services and context represented and reasoned by ontologies. We describe the complex interactions of these technologies facilitating the construction of context-aware systems.

The rest of the chapter is organized as follows: In Section 2, the general architecture is described; Section 3 describes the set of agents and their roles inside the architecture; Section 4 presents a multi-variable environment control system; Section 5 presents the ontological models defined for the architecture; Section 6 presents an overview of related work, and finally, in Section 7, conclusions are presented.

2. Description of the architecture

The proposed architecture is envisioned for a wireless networked **environment**, where users may be identified by their mobile device mac address or by a RFID card. Such an **environment** may be an office or laboratory into an academic institution or university, where users enter and leave the **environment** freely. The proposed architecture consists of five layers interconnected, which are described in this section. **Figure 2** shows the general description of the architecture.

2.1. Sensor network

This layer consists of a collection of physical sensors, mobile sensors and actuators. The objective of the network sensor is to obtain data from the physical context, user context and eventually activate some actuators. This layer aims at constant monitoring of environmental data such as temperature, lighting, humidity, smoke or fire and presence of humans into the environment. Another important objective of this layer is the possible identification of the users and the data generated by user interaction with the environment. The following types of sensors are considered as part of this layer:

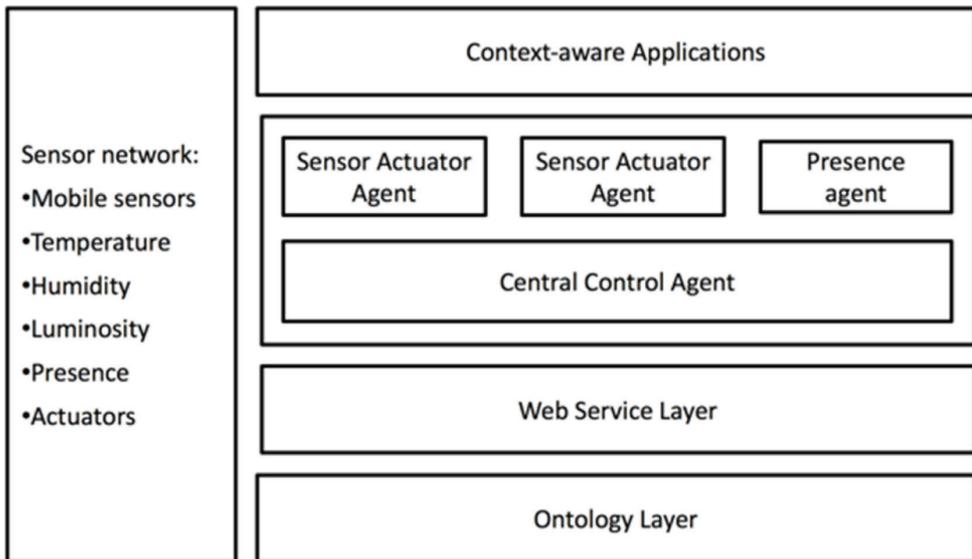


Figure 2. Hybrid architecture for context-awareness.

- Environmental sensors, which are used to obtain data of room temperature, humidity, luminosity and presence of persons.
- Mobile and wearable sensors, such as accelerometer, gyroscope, magnetometer, proximity sensor, light sensor, barometer, thermometer, pedometer, heart rate monitor, fingerprint sensors, etc.
- Automatic identification sensors carried by the user.
- Actuators represent the hardware devices through which actions are activated in order to achieve an ideal state.

2.2. Intelligent agents

Intelligent agents play an important role into the architecture. Every agent is a programme allocated into a microcontroller (Arduino or Raspberry Pi) which interacts with physical sensors and actuators to monitor and control the physical variables of the environment.

There are specialized agents performing different roles:

- Sensor and actuator agent: This kind of agent is continuously sensing the environment to detect changes and report variations that are higher than a threshold. This type of agent is capable of receiving action commands to execute over the environment by activating actuators.
- Central control agent: This agent is responsible for reading and forwarding all communication messages incoming or outgoing between agents, while recording all those communications.

- **Context Server:** This is the main server that computes all events and fires the actions that are executed by the environment control system, where the ontology model resides and the reasoning services execute inference about events and trigger actions to be taken by the control system.

2.3. Ontologies for context modelling

In this layer, models based on ontologies for representing contextual data, data obtained from the user, data from sensors, events and physical spaces are included. Additionally, a set of query and inference rules are included in the definitions of ontologies in order to gather more related and relevant data. Ontologies offer a formal semantic representation of data and facilitate the inference about the stored data, which helps to retrieve information relevant to the user. Moreover, being a technology based on the Web, they can be shared by multiple applications and automatically processed by computers.

2.4. Web services

Web services are incorporated for two purposes. On the one hand, Web services for data management, information extraction, storage, retrieval and updating of information in ontologies. Moreover, Web services for inference, reasoning and verifying the consistency of the data will also be created. These Web services will be supported, in full, in considering the ontological model semantic relationships between data.

2.5. Context-aware applications

In this layer, mobile applications for user interaction are developed. This interaction involves light applications for information retrieval, voice and natural language communication interfaces, mobile applications with requests in natural language and mobile applications where relevant and timely information is provided to users. All these communication applications will be focused on evaluating the usability of the context-aware environment.

3. Intelligent agents

Intelligent agents play important roles in the context-aware architecture. Intelligent agents are autonomous programmes that are responsible for the detection of changes in the state of the environment, they also do intermediation sending and receiving messages with other agents and context servers, and they are responsible for firing and executing actions that change the state of the environment. In this architecture, every agent is a programme allocated into a microcontroller (Arduino or Raspberry Pi) which interacts with physical sensors and actuators that monitor and control the physical variables of the environment. Physical agents are responsible for monitoring temperature, humidity and luminosity variables and firing the respective actuators, whereas presence agents are responsible for user recognition and if possible user identification inside the environment. All agents participating in the environment

communicate with the central agent, where all data are concentrated and context-related decisions are taken. The following agent roles are defined:

- a. **Sensor/actuator agent (SAA):** This kind of agent is mounted on an Arduino electronic platform, which integrates a temperature sensor, a humidity sensor, luminosity and a presence sensor. Is responsible for continuous acquisition of these environment data, and for the activation of respective actuators.
- b. **Central control agent (CCA):** This agent is mounted on a Raspberry Pi card-sized computer. It is an intermediary that reads and forwards all communications from SAAs to the context server (CS) while recording all those communications.
- c. **Context server (CS):** This is the main server that computes all events and fires the actions that are executed by the environment control system, where the ontology model resides and the reasoning services execute inference about events and trigger actions to be taken by the control system.

In this section, we describe the **interaction protocol** that was defined for communication purposes between all agents. A protocol specifies the rules of interaction between agents by restricting the range of allowed utterances sequences for each agent at any stage during a communication interaction [7]. According to Foundation for Intelligent Physical Agents (FIPA) specifications [8], an agent communication language (ACL) message structure contains one or more of the parameters described in **Table 2**, accordingly the only mandatory parameter is performative.

All messages defined for the interaction protocol are described in **Table 3**.

Even though agent-communicating messages were designed based on FIPA specifications, messages are translated to byte arrays packages of longitude 2 or 3. All packages were defined trying to optimize the available transmission channel. Therefore, it is sought that the size of the package is the minimum possible.

Figure 3 shows the interaction protocol implemented for communication purposes between all agents participating in the context-aware environment.

Element type	Message parameters
Type of communicative act	Performative
Participant in communication	Sender, receiver, reply-to
Content of message	Content
Description of content	Language, encoding, ontology
Control of conversation	Conversation-id, reply-with, in-reply-to

Table 2. Elements of a message according to FIPA specification.

Performative	Message format
Request	REQUEST <sender Id, receiver Id, date, time>
Info	INFO <sender Id, receiver Id, date, time, temperature, humidity, luminosity>
Presence	PRESENCE <sender Id, receiver Id, date, time, user Id>
Event	EVENT <sender Id, receiver Id, date, time, increase decrease, temperature humidity luminosity, value>
Action	ACTION <sender Id, receiver Id, date, time, increase decrease, temperature humidity luminosity, value>

Table 3. Description of the message format for each type of performative.

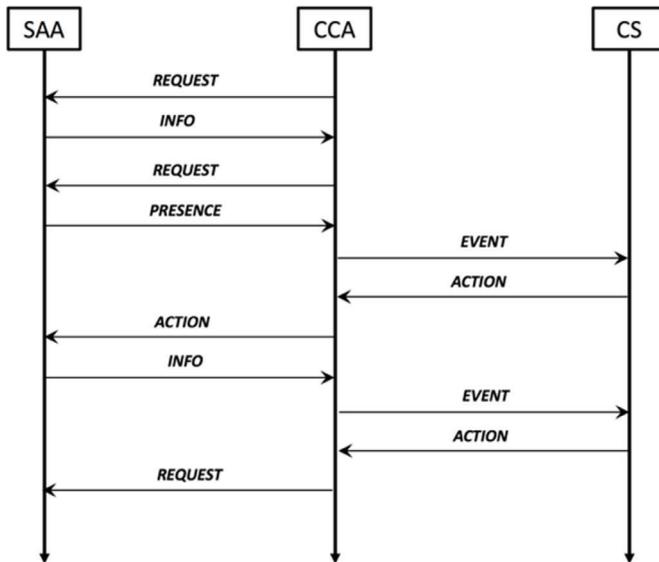


Figure 3. Agent interaction protocol.

Communication is a key requirement in any context-aware system, in this architecture, communication is carried out between agents. **Figure 3** shows that the central control agent (CCA) initiates the communication with each of the sensor/actuator agents (SAA) deployed into the environment. The CCA *requests* for information to the SAA agent, then the SAA agent delivers an *info* message with attached environmental data. In response to a *request* message, the SAA agent may deliver a *presence* message, indicating the detection of a person inside the environment, adding the unique identification of the person (RFID or MAC address). The CCA communicates with the context server (CS) using *event* messages and receiving *action* messages. An *event* message is issued whenever the value of the

environmental variables changes significantly. As a response to an *event* message, an *action* message may be issued by the CS. The principal difference between an *event* and an *action* is that the former is a change in the environment that was caused by natural reasons, and *action* is a command to change the environment through an environmental control system.

3.1. Multi-variable environment control system

The set of intelligent agents deployed in the physical environment communicate with a multi-variable environment control system (MECS), which is a closed-loop system with feedback. **Figure 4** shows the components of this particular multi-variable control system.

3.2. Environment setting

Environment setting consists of a set of actions that are executed by the MECS at any moment inside the context-aware environment in order to achieve an ideal environment state. Considering a networked environment where users enter and leave the environment, the desired environment state is the set of values defined by the administrator of the environment considering mainly the activities and type of works to be carried out in the physical environment, the kind and number of possible users, and the geography where the environment is located. For the purpose of this work, the environment state is represented as a three-valued vector using the variables in **Table 4**.

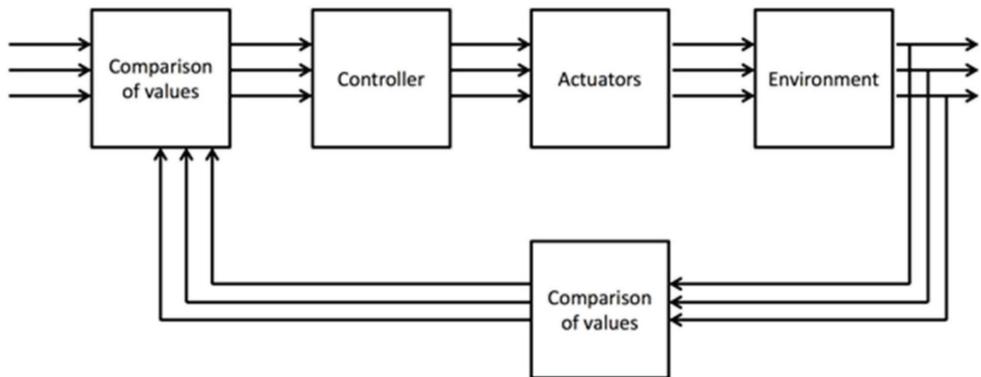


Figure 4. Multi-variable environment control system.

Variables	Humidity (%)	Temperature (°C)	Luminosity (lux)
Allowed values	30–60	–20 to 50	100–500
Desired values	45	23	200

Table 4. Environment state vector.

Activity	Types of work	Average illuminance (lux)	Minimum measured illuminance (lux)
Work requiring limited perception of detail	Kitchens, factories assembling large components, potteries	100	50
Work requiring perception of detail	Offices, sheet metal work, bookbinding	200	100
Work requiring perception of fine detail	Drawing offices, factories assembling electronic components, textile production	500	200

Table 5. Lighting recommended values.

Each of these variables defines a range of allowed values and a range of desirable values. Indoor humidity levels should be between 30 and 60%, with the ideal level being about 45%. The temperature value depends on the season of the year, the geographical location of the environment and the number of persons inside the environment. The most important aspects that influence the indoor temperature are heat from persons, heat from lights, heat from electric equipment and machines, among others. However, it is not necessary to measure all these particular data, the architecture only requires the initial specification of the ideal range of temperature to function normally and securely. Environment climate allows values ranging from the -20 to 50°C .

Lighting levels depend on various factors, such as the time of the day (morning light versus night light poses different requirements). In order to define an ideal lighting level, the administrator should consider mainly the number of persons inside the environment, the particular lighting requirements (in case that users present in the environment have sight difficulties). The amount of light falling on a surface is called ‘illuminance’, and it is measured in lux. This is the measurement used to optimize visual comfort because building regulations and standards use illuminance to specify the minimum light levels for specific tasks and environments. Lighting recommended values are shown in **Table 5**.

4. Ontologies for context modelling

In this section, we describe the ontologies that were designed for context modelling and reasoning. We define the design principles that guided the construction of the ontology models. Ontologies are representational models that can help to characterize and specify all of the entities needed to describe the environmental context [9] and the user profiles. A context can be composed of contextual items such as location, physical data and activity, instrumental and social context [10]. In particular, in this work, the context is divided into two general classes: **environmental context** and **user profile context**.

The logical foundation of ontologies allows the explicit specification of the user preferences and user profiles, and the reasoning facilities offer mechanisms to gather more related information in order to provide pertinent and opportune information and services to users [11, 12].

4.1. Motivating scenario

Considering a traditional academic institution in which professors are teaching subjects to students in classrooms, pre-graduate students are developing their thesis, there is a chief for each department directing and supervising administrative activities; there are academic coordinators attending student academic issues, aided with the support of administrative staff (secretaries, janitors, etc.), and visitors who come for various reasons.

The ontology model consists of three ontologies that are included into another general context ontology system. **Figure 5** shows the general ontology model.

The general ontology model consists of three ontologies:

1. The **Person** ontology was designed to represent all the information related to persons that may exist in a typical academic scenario where professors, students, staff and visitors assist. An important characteristic of this ontology was to define a unique identifier for every type of person that would be present inside the sensor-enabled context. **Figure 6** shows the general model of the **Person** ontology and **Table 6** presents some classes definitions of the **Person** ontology.
2. The **PhysicalSpace** ontology was designed to represent any kind of physical location such as cubicle, classroom, office, parking lot, plaza, green area, etc. The **PhysicalSpace** class is sub classified into **IndoorSpace** and **OutdoorSpace** subclasses. **Figure 7** shows the general class hierarchy of the **PhysicalSpace** ontology.
3. The **Device** ontology was designed to represent electronic devices located within the context-aware environment. The **Device** class is sub classified into **smartphone**, **RFID card**, **sensor** and **actuator** subclasses. **Figure 8** shows the general model of the **Device** ontology. An important issue of any sensor device is its capability of measuring; therefore, devices are semantically related with **physical measurement** subclasses of **light intensity**, **humidity**, **temperature** and **distance**.

The current version of the ontology was implemented in OWL 2 ontology language, and contains 35 classes, 14 object properties, 83 data type properties and has an ALCRQ(D) expressivity. **Table 7** shows the classes, object properties and data type properties defined for the ontology.

4.2. Ontology design principles

The set of ontology models reported in this chapter address particularly clarity and coherence design principles.

- **Clarity design principle:** According to Ref. [13], ontology should communicate the intended meaning of defined terms. Definitions should be objective. Definitions should be stated in formal axioms, and a complete definition (defined by necessary and sufficient conditions) is preferred over a partial definition (defined by only necessary or sufficient conditions). In order to accomplish clarity, we designed ontologies defining equalities in axiomatic class definitions.
- **Coherence design principle:** This principle is also referred as soundness or consistency. Coherence specifies that ontology definitions should be individually sound and should not contradict each other [14].

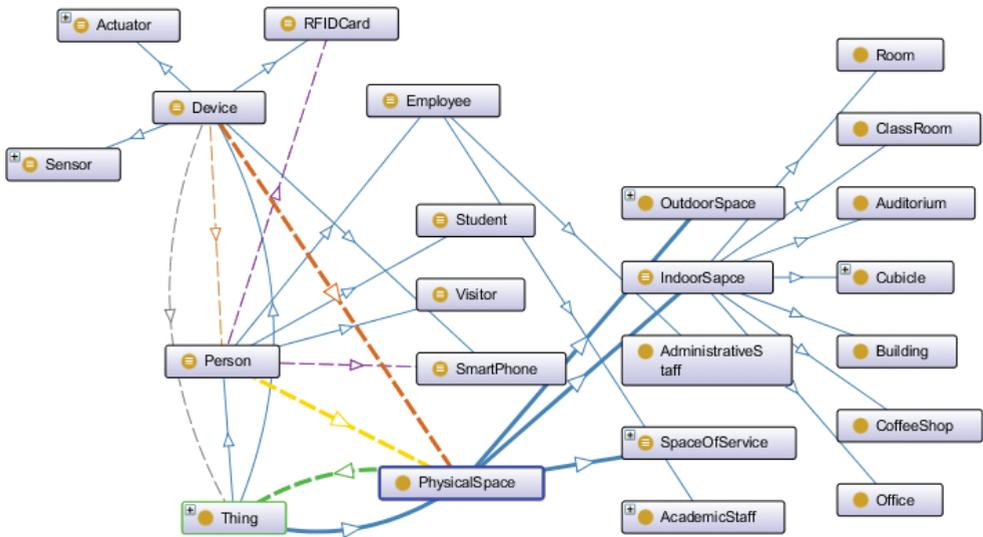


Figure 5. General context ontology model.



Figure 6. Class hierarchy of the Person ontology.

Ontology consistency checking was executed to verify that none of the class definitions and axioms had logical contradictions, or the individual's instantiated into the ontology. This final activity consists of executing the reasoning tasks of taxonomy classification, compute inferred types and consistency checking. The most important design principles were considered and verified through protégé tools such as Fact++ reasoner and DL-query tool. After execution of Fact++, individuals were correctly classified. For instance, Professor Ricardo Lopez was correctly classified as member of the **Professor** class. As a result, the ontology models accomplish the clarity and coherence design principles.

Concept	Axiomatic definition	Human definition
Person	(hasAge some int) and (hasGender some string) and (hasPersonName some string) and (hasWeight some float)	A Person is an individual that has age, has gender, has name and has weight
Employee	Person and (hasEconomicNumber some string)	Is a Person that has an economic number
Smartphone	Device and (hasMacAddress some string) and (hasIMEI some string)	Is a Device that has MAC address and has IMEI
Course	(hasCourseName some string) and (hasCredits some int) and (hasCourseKey exactly 1 string)	A course is an individual that has course name, has credits, and has primary key

Table 6. Some classes definition from the Person ontology.

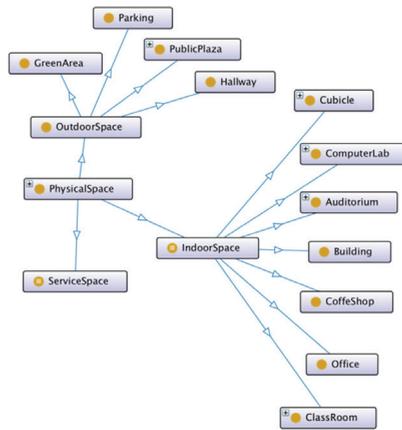


Figure 7. General model of the physical space ontology.

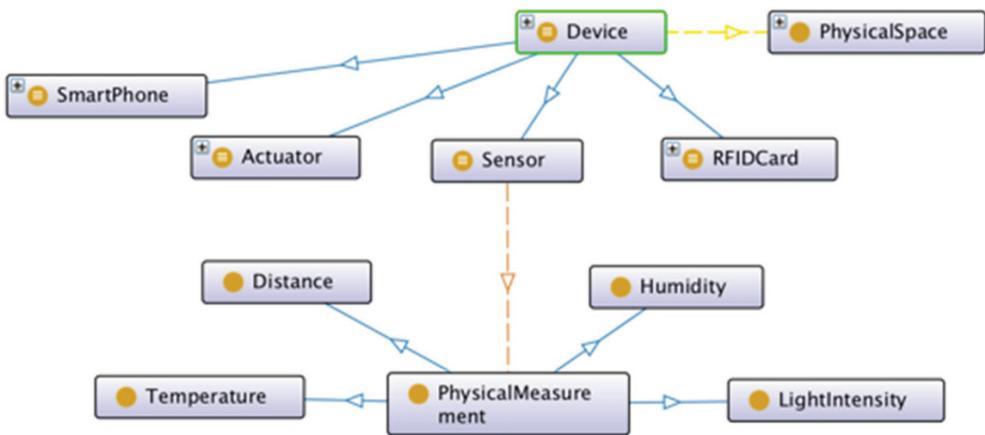


Figure 8. General model of the device ontology.

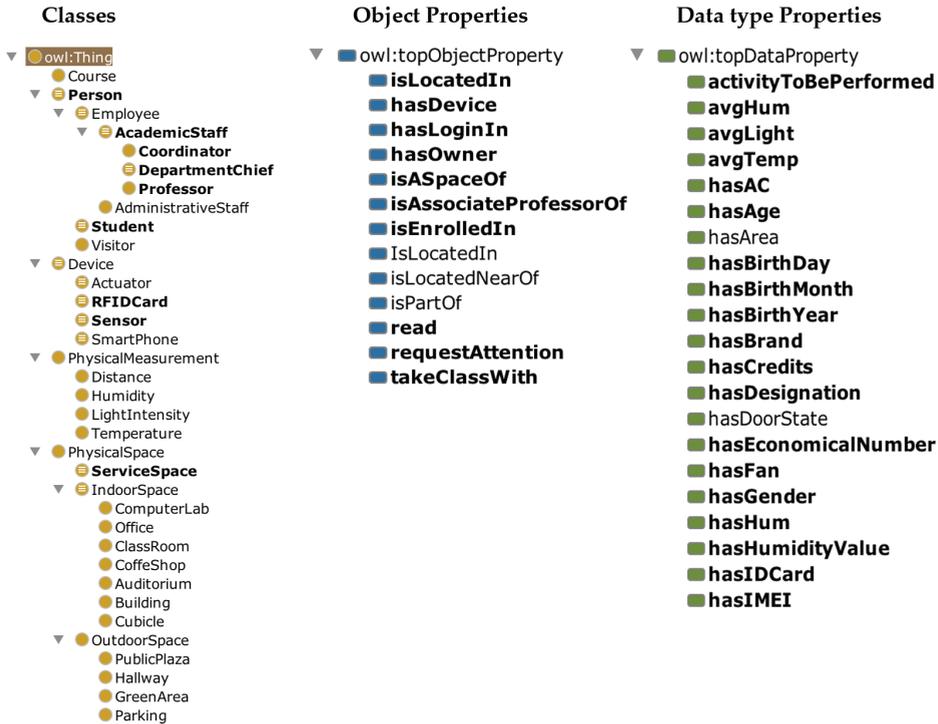


Table 7. Classes, object properties and data type properties of the ontology.

5. Related work

The use of ontologies for context modelling is not a new research topic; there are many works in literature that describe the utilization of ontologies to support context-awareness or pervasive environments. In this section, a chronological overview of works reporting ontologies, architectures and frameworks for context modelling is presented, highlighting the main differences (see Table 8).

Chen, Finin and Joshi [15] described **CoBrA**, a context broker agent architecture that is capable of managing a shared model of the context and reasoning support for context-aware applications. The objective of CoBrA is to facilitate knowledge sharing and reasoning between agents.

Razmerita, Angehrn and Maedche [16] presented in 2003 **OntobUM**, a generic ontology-based user modelling architecture. This architecture integrates three ontologies: the user ontology, the domain ontology and the log ontology. Later in 2007 [17], authors augmented their **OntobUM** model by representing the behaviour of user's concept, such as level of activity, type of activity, level of knowledge sharing, etc. They present a conceptual layered architecture integrated with a presentation layer, a middleware layer and a storage layer. This later

Characteristics of the architecture or framework		Context concepts represented			Context management		
Agent oriented	Service oriented	Person or user profile	Physical context	Activities	Context representation model	Context reasoning	Context acquisition
Yes	Yes	No	Yes	Yes	Ontologies	Flora-2	Automatically by Sensors and mobile devices
No	Yes	Yes	Yes	Yes	Ontologies	DL reasoning	Manually introduced by ontology designers
Yes	Yes	Yes	No	Yes	Ontologies	No	Manually introduced by users
No	Yes	Yes	Yes	Yes	Ontologies	No	Manually introduced by ontology designers
No	No	Yes	Yes	Yes	Ontologies	No	Manually introduced by ontology designers
No	No	Yes	No	Yes	Ontologies	No	Manually introduced by ontology designers
No	No	Yes	Yes	Yes	Ontologies	No	Manually introduced by users

Table 8. Related work of ontologies for context-aware system.

architecture is similar to the architecture proposed and described in Section 2; however, the purpose of their applications differs, **OntoBUM** is intended for knowledge sharing between users inside an organization; whereas our proposed architecture is abstracted from a particular organization and it was designed to support context-aware environments and context-aware systems.

Wang et al. [18] described in 2004 **CONON**, an ontology for modelling context in pervasive computing environments. Authors propose an ontology model divided into *upper* ontology and *specific* ontology. The upper ontology model defines computational entity, location, person and activity as the most important entities of a context model. Later in 2004 [19], authors presented **SOCAM**, a service-oriented context-aware middleware architecture to support the construction of context-aware services in intelligent environments. **SOCAM** architecture incorporates **CONON** ontology.

Preuveneers et al. [20] presented **CoDAMoS**, an extensible context ontology for ambient intelligence, which describes four main concepts: user, environment, platform and service. Authors described the requirements for ambient intelligence: application adaptability, resource awareness, mobile services, semantic service discovery, code generation and context-aware user interfaces.

In 2010, Poveda-Villalón et al. [21] presented **mIO!** ontology network for a mobile environment. **mIO!** ontology consists of 11 modular ontologies: user, role, environment, location, time, service, provider, device, interface, source and network. This ontology covers a wide range of concepts related with context representation, however; authors do not present any reasoning results.

Skillen et al. [22] presented in 2012 a user profile model for context-aware application personalization; authors concentrated on concepts to model a dynamic context: user time, user location, user activity and user context.

6. Conclusions

The work reported in this chapter incorporates various technological paradigms, such as intelligent agents, network sensors, Web services and ontologies. The main objective of integrating these technologies was to support the development of more complex and intelligent context-aware applications.

The use of models implemented with ontologies offers significant advantages: the ability to exchange, expand, extend and maintain the individual ontologies. An example is the **Person** ontology, which can be interchanged as needed to adapt to new application needs.

The incorporation and exploitation of agents, Web services and ontological models is a clear trend that promises to improve the automatic selection and invocation of legacy and new Web services.

All these technologies together (Web services, intelligent agents and ontologies) are key facilitators for the wise management of context-based systems.

Author details

Maricela Bravo*, José A. Reyes-Ortiz, Leonardo Sánchez-Martínez and Roberto A. Alcántara-Ramírez

*Address all correspondence to: mcbc@correo.azc.uam.mx

Autonomous Metropolitan University, Delegación Azcapotzalco, Ciudad de México, México

References

- [1] Guermah H, Fissaa T, Hafiddi H, Nassar M, Kriouile A. An Ontology Oriented Architecture for Context Aware Services Adaptation. arXiv preprint arXiv:1404.3280
- [2] Abowd GD, Dey AK, Brown PJ, Davies N, Smith M, Steggles P. Towards a better understanding of context and context-awareness. In: International Symposium on Handheld and Ubiquitous Computing; September 1999; Berlin, Heidelberg: Springer; pp. 304-307
- [3] Orsi G, Tanca L. (2011). Context Modelling and Context-Aware Querying. In Datalog Reloaded. Berlin, Heidelberg: Springer; 2011. pp. 225-244
- [4] Perera C, Zaslavsky A, Christen P, Georgakopoulos D. Context aware computing for the internet of things: A survey. IEEE Communications Surveys & Tutorials. 2014; 16(1):414-454
- [5] Jennings NR, Wooldridge M. Agent-oriented software engineering. Journal of Artificial Intelligence, 2000;117:277-296
- [6] Neches R, Fikes RE, Finin T, Gruber TR, Patil R, Senator T, Swartout WR. Enabling technology for knowledge sharing. AI Magazine. 1991;12(3):16-36
- [7] Endriss U, Maudet N, Sadri F, Toni F. Logic-based agent communication protocols. Advances in Agent Communication Languages. 2004;2922:91-107
- [8] FIPA ACL Message Structure Specification [Internet]. Available from: <http://www.fipa.org/specs/fipa00061/SC00061G.html>
- [9] Devaraju A, Simon H. Ontology-based context modeling for user-centered context-aware services platform. In: Information Technology, 2008. ITSIM 2008. International Symposium on. Vol. 2. New York: IEEE; 2008. pp. 1-7
- [10] Paganelli F, Bianchi G, Giuli D. A context model for context-aware system design towards the ambient intelligence vision: Experiences in the eTourism domain. In: Universal Access in Ambient Intelligence Environments; Berlin, Heidelberg: Springer; 2007. pp. 173-191
- [11] Chen G, Kotz D. A survey of context-aware mobile computing research. Technical Report TR2000-381, Department of Computer Science, Dartmouth College 2000;1(2.1)

- [12] Gruber TR. A translation approach to portable ontologies. *Knowledge Acquisition*. 1993;5(2):199-220
- [13] Gruber Thomas R. Toward principles for the design of ontologies used for knowledge sharing? *International Journal of Human-Computer Studies* 43(51995):907-928
- [14] Morbach Jan, Andreas Wiesner, Wolfgang Marquardt. OntoCAPE—A (re) usable ontology for computer-aided process engineering. *Computers & Chemical Engineering* 2009;33(10):1546-1556
- [15] Chen H, Finin T, Joshi A. An ontology for context-aware pervasive computing environments. *The Knowledge Engineering Review*. 2003;18(03):197-207
- [16] Razmerita L, Angehr, A, Maedche A. Ontology-based user modeling for knowledge management systems. In: *International Conference on User Modeling*; June 2003; Berlin, Heidelberg: Springer; pp. 213-217
- [17] Razmerita L. *Ontology-Based User Modeling*. In *Ontologies US*: Springer; 2007. pp. 635-664
- [18] Wang XH, Zhang DQ, Gu T, Pung HK. (2004, March). Ontology based context modelling and reasoning using OWL. In: *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*; March 2004; Washington, DC, USA.: IEEE; pp. 18-22
- [19] Gu T, Wang XH, Pung HK, Zhang DQ. (2004,). An ontology-based context model in intelligent environments. In: *Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference*; January 2004; San Diego, California, USA. 2004: pp. 270-275
- [20] Preuveneers D, Van den Bergh J, Wagelaar D, Georges A, Rigole P, Clerckx T, De Bosschere K. (2004, November). Towards an extensible context ontology for ambient intelligence. In *European Symposium on Ambient Intelligence*; November 2004; Berlin, Heidelberg: Springer; pp. 148-159
- [21] Poveda Villalon M, Suárez-Figueroa MC, García-Castro R, Gómez-Pérez A. A Context Ontology for Mobile Environments. In *Proceedings of the 1st Workshop on context, information and ontologies*, Heraklion, Greece. 2010
- [22] Skillen KL, Chen L, Nugent CD, Donnelly MP, Burns W, Solheim I.. Ontological user profile modeling for context-aware application personalization. In: *International Conference on Ubiquitous Computing and Ambient Intelligence*; December 2012; Berlin, Heidelberg: Springer; pp. 261-268