REVISTA BOLETÍN REDIPE: 14 (3) MARZO 2025 ISSN 2256-1536
RECIBIDO EL 16 DE OCTUBRE DE 2024 - ACEPTADO EL 18 DE DICIEMBRE DE 2024

Modeling an internet of things architecture to manage atmospheric carbon emissions in cities

Modelado de una arquitectura de la internet de las cosas para gestionar la emisión de carbono atmosférico en las ciudades

Mauricio Orlando Bermúdez Amaya[1]

Universidad Distrital Francisco José de Caldas
UDFJC

Abstract

Climate change caused by global warming and in turn produced by the excess of greenhouse gases GHG emitted into the Earth's atmosphere by anthropogenic action –specifically carbon dioxide CO2– must be kept under control due to the disastrous consequences for the conservation of life on the planet. Within the wide range of alternatives, there is the option of optimizing electricity consumption by electronic devices for daily use through the Internet of Things IoT, given that the energy to be used has already been generated in the respective power plants and generally by burning fossil fuels. This research concerns an initiative to model a flexible functional IoT architecture, leading to the mitigation and regulation of carbon gas emissions mainly in cities –since this is where most use is made of Machine to Machine M2M technology devices–, supported by syntactic and semantic interoperability through the integration of relevant, robust and recognized ontologies –such as the oneM2M project for the standardization of M2M/IoT communications, the SAREF smart device reference and for the energy domain SAREF4ENER, together with the OM measurement units–, in a single coherent and viable corpus whose mapping and simulation is ad hoc to the universe of electric energy discourse.

Resumen

El cambio climático causado por el calentamiento global y a su vez producido por el exceso de gases de efecto invernadero GEI emitidos a la atmósfera terrestre por acción antropogénica –específicamente dióxido de carbono CO2– debe mantenerse bajo control debido a las desastrosas consecuencias para la conservación de la vida en el planeta. Dentro del amplio abanico de alternativas, existe la opción de optimizar el consumo de electricidad en los dispositivos electrónicos de uso cotidiano a través del Internet de las Cosas IoT, dado que la energía a utilizar ya ha sido generada en las respectivas centrales eléctricas y generalmente mediante la quema de combustibles fósiles. Esta investigación se refiere a una iniciativa para modelar una arquitectura IoT funcional y flexible, que conduzca a la mitigación y regulación de las emisiones de gases de carbono principalmente en las ciudades –ya que es donde más se hace uso de los dispositivos de tecnología Máquina a Máquina M2M –, apoyada en la interoperabilidad sintáctica y semántica mediante la integración de relevantes, robustas y reconocidas ontologías –como el proyecto oneM2M para la estandarización de las comunicaciones M2M/IoT, la referencia de dispositivos inteligentes SAREF y para el dominio energético SAREF4ENER, junto con las unidades de medida OM–, en un único corpus coherente y viable cuyo mapeo y simulación es ad hoc al universo del discurso de la energía eléctrica.

Keywords
Internet of Things IoT, Machine to Machine M2M, ontology, oneM2M, SAREF4ENER, SAREF, OM, Atmospheric carbon CO2.

Palabras clave
Internet de las Cosas IoT, Máquina a Máquina M2M, ontología, oneM2M, SAREF4ENER, SAREF, OM, Carbono atmosférico CO2.

1. Introduction

Recently, the Internet of Things (IoT) has become a timely resource for the permanent connectivity of most objects in nature and civilizations. In these societies, people use all types of devices to satisfy their need to improve their quality of life, which requires a constant supply of energy for their operation. A large portion of this energy is produced by burning fossil fuels, a situation that causes global warming with its already experienced effects and whose eventuality needs to be mitigated and/or controlled.

Likewise, IoT is an appropriate and convenient alternative to moderate the aforementioned climate change within tolerable limits for the planet, connecting low-consumption electronic devices through machine-to-machine (M2M) technology, managing their respective energy expenditure online without compromising their performance. This allows the reduction of greenhouse gas emissions by using the amount of electrical energy specific for their functions and whose generation occurs in advance in the respective power plants by combusting organic materials and/or their derivatives.

In today's cities, one of the greenhouse gases released into the environment in the greatest quantity is carbon dioxide (CO2), making it difficult for nature to recycle, given its large proportion in the atmosphere. However, the emission of CO2 in the urban environment due to industrial, commercial, and service activities does not occur in the consumption of electric energy per se because when closing the circuit (i.e., using the machine) for the said energy to flow, it has already been produced in advance; therefore, the carbon dioxide generated has already been emitted and is circulating in the ecosystem.

This leads to an initiative to conceive a resource or means that allows, rather than mitigates, the management of carbon dioxide emissions in cities without undermining their sustainable development, being within the set of alternatives, the one that corresponds to the modeling of a flexible functional architecture of the Internet of Things, a pertinent, coherent, and viable option, and proposes to semantically interoperate the myriad of M2M devices in the world so that they self-regulate independently of the inherent and/or underlying information and communication technology.

2. Reference framework

The proposed architecture requires formal generalization of the concepts of a given universe of discourse based on its inherent properties and relationships. Such a universe is composed of domains corresponding to the Internet of Things (IoT), M2M devices, communication protocols, energy, and fundamental magnitudes, which constitute an ontology whose model lays the foundations for the machines in question to connect and manage the energy expenditure of their processes in real time and, therefore, manage CO2 emissions in cities.

The initiative of the flexible functional architectural model that makes use of both IoT and M2M technology is supported by the concept of "ontology". An ontology is an explicit specification of a conceptualization [1], which is an abstract and simplified vision of an environment that is desired to be represented for a defined purpose. It facilitates the construction of a body of knowledge formally represented in a declarative manner through objects, properties, and relationships described between them, in a determined universe of discourse [2] also called a domain, so –expressly and concretely– an ontology is the structured manifestation of a logical theory [3].

This ontologically based functional architecture is based on the proposed model of integration, mapping, and simulation of four ontologies (Table 1).

Acronym	Meaning
oneM2M	One Machine To Machine
SAREF	Smart Appliances REFerence
SAREF4ENER	Smart Appliances REFerence For Energy
OM	Ontology of units of Measure

Table 1. Ontologies that make up the proposed architectural model for the Internet of Things IoT (Own elaboration).

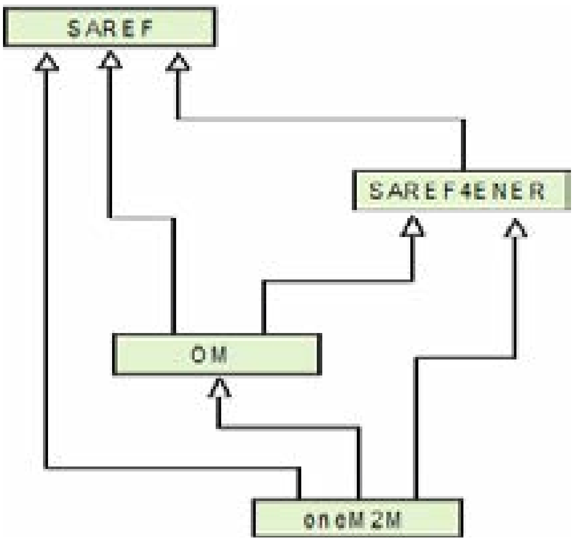

Figure 1. Conceptual model of a proposed IoT architecture (Own elaboration).

The proposed conceptual model of the IoT architecture comprises the four ontologies listed in Table 1, and its corresponding scheme is shown in Figure 1.

These ontologies are developed and supported by recognized organizations in the ICT sector, each of which comprises specific concepts with distinctive characteristics and associations within a specific universe of discourse or domain, as shown in Table 2.

Ontology	Universe of discourse (domain)	Author
oneM2M	M2M System (Machine to Machine)	oneM2M Alliance
SAREF	Smart Devices	TNO, ETSI
SAREF4ENER	Energy (Consumption and/or Production)	TNO, ETSI
OM	Units of measurement	Wageningen UR

Table 2. Domain and developer of the ontologies of the proposed architectural model (Own elaboration).

The various ontologies that constitute the proposed architectural model for the Internet of Things (IoT) are briefly described below. The classes, object properties, and data properties of each ontology are not detailed, because they are extensive. Therefore, the reader is referred to the respective documents on the Web and bibliography, where these specifications are available.

2.1. Ontology of units of Measure OM

The Ontology of Units of Measure (OM), version 2.0, is a framework of essential concepts and relationships for scientific research [4] that presents a strong focuses on units, quantities, measures, and dimensions. These notions are organized as shown in Figure 2.

This ontology (OM) has been chosen as a conceptual framework for the present research, without intending to diminish the importance of other ontologies that also focus on metrology, e.g., the ontology of quantities, units, dimensions and data types QUDT, which is an active modeling effort in the Web Ontology Language with Descriptive Logic OWL-DL for the same field [5]. However, it is worth highlighting that OM, despite being an ontology of the domain of the mentioned concepts, also favors that the quantitative data of any pure or applied scientific work be more explicit; this with the purpose that they can be integrated, verified and reproduced [5].

.

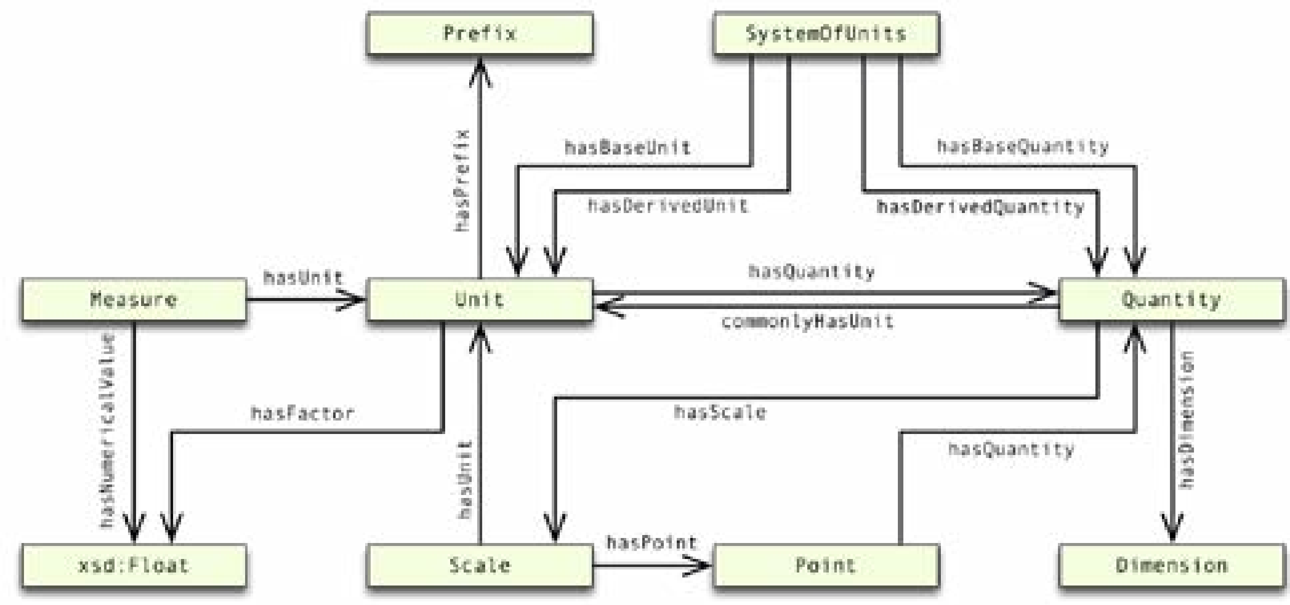

Figure 2. Basic conceptual structure of OM ontology [4].

It is worth noting that OWL-DL is one of the sub-languages or species of Web Ontology Language OWL, which facilitates inferences from implications and equivalences computable in finite time [6] through Descriptive Logic (DL), also called first-order predicate logic – understanding the predicate as an expression that designates the characteristics or relations of an individual or instance [7]–, whose purpose is to predict the behavior of a system (or the validity of reasoning) in a consistent and complete manner in its entirety, and in turn partially decidable at the level of monadic statements or some polyadic premises [7] in a given universe of discourse.

2.2. Smart Appliances Reference Ontology SAREF

The Smart Appliances REFerence Ontology – version 2.1.1– is a conceptual system built from the various existing standards, protocols and data models, also called "semantic assets" [8] [9] [10], in the domain of smart devices.

The idea of "smart device" refers to an artifact or machine that is used at home to perform domestic work [11], characterized by having the ability to communicate with other artifacts or devices and allowing their control through the Internet.

Such machines may be [8] [10] [11]:

- Sensors for homes or buildings (temperature, humidity, energy meters, etc.).
- Actuators (windows, doors, blinds, etc.).
- White goods (washing machine, oven, etc.).
- Brown goods (television, camera, etc.).
- Gray goods (mobile phones, tablets, etc.).
- Heating, ventilation, and air conditioning.
- Lighting (lamps, reflectors, photometers).
- Micro-renewables (solar panels and heaters, wind turbines, etc.).

The concept of smart devices applies not only to the domain of the home or private dwellings [8] [10]; it is also aimed at offices and ordinary private and/or public buildings, except for special equipment such as elevators, medical instruments, and dispensers. Furthermore, the recognition of the mentioned devices is performed only with respect to the semantic requirements for the relevant energy operations [10], such as switching on and standby, but not for content management, such as choosing a channel or station.

In the SAREF ontology, such semantic representations are essentially structured as shown in Figure 3 (developed by the author based on [9] [10] [11]) and following the guidelines of the Web Ontology Language OWL-DL, in which the "concepts and roles" of the descriptive logic are assimilated to "classes and properties" (whether the latter are attributes or relations), respectively [6].

The notation of these concepts or classes within the SAREF ontology is established by the prefix "saref", which is separated from the morpheme that indicates the concept referred to using the orthographic symbol ":", this is "saref:Device", "saref:Measurement","saref:Command",etc.[11]. Likewise, the roles or properties have the writing "prefix:relation" (also called "ObjectProperties") or "prefix:attribute" (named as "DataProperties"), whether it is a property of a relation category or an attribute category, accordingly [12] [13]. For example: "saref:offers", "saref:actUpon", "saref:hasTypicalConsumption", etc., for relationships; "saref:hasManufacturer", "saref:hasModel", "saref:hasDescription", etc., for attributes, these being characteristics of a predefined data type (text, integer, etc.) that can vary in terms of their content [12].

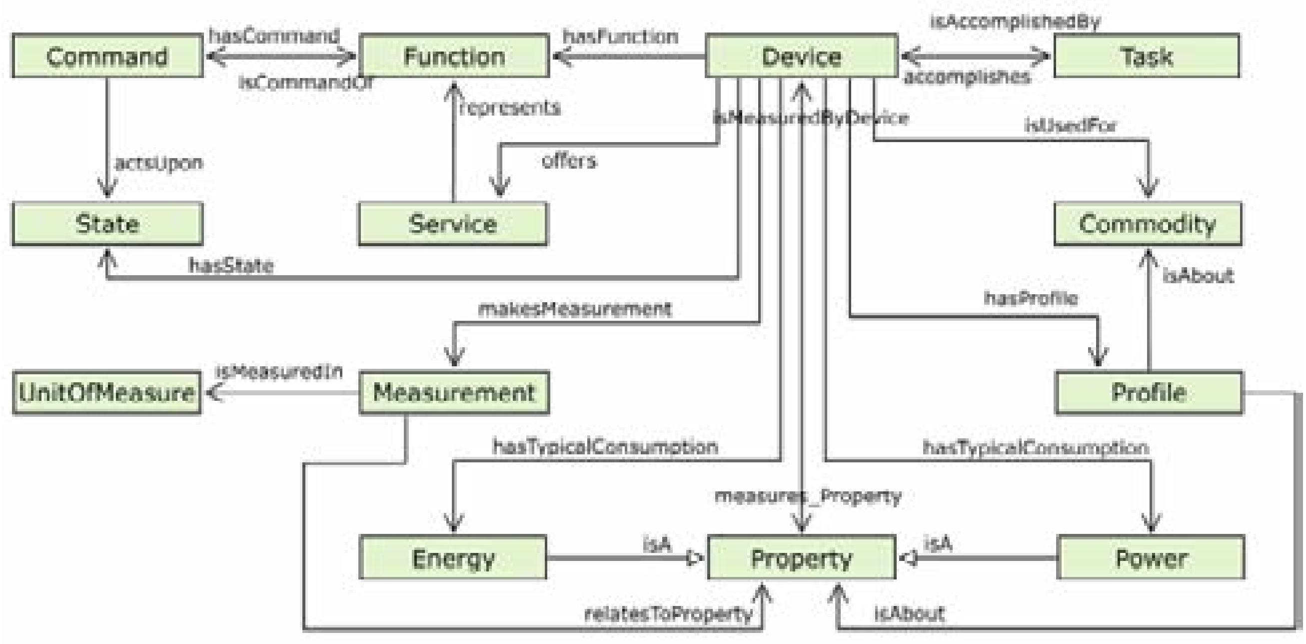

Figure 3. Basic conceptual structure of the SAREF ontology (Own elaboration based on [9] [10] [11]).

For a better understanding of the analogy between Descriptive Logic (DL) or first-order predicate logic, the Web Ontology Language for Descriptive Logic OWL-DL and the Reference Ontology for Smart Devices SAREF (Table 3) were constructed (by the author) according to [6] [7] [9] [10] [11] [12] [13] [14]:

Descrip-tive Logic (DL)	**OWL-DL**	**SAREF (Example)**
Concept	Class	saref:Device
		saref:Measurement
		saref:Command
Role	Property (as a Relationship): Object-Property	saref:offers
		saref:actUpon
		saref:hasTypical Consumption
	Property (as an Attribute): DataProperty	saref:hasManufacturer
		saref:hasModel
		saref:hasDescription
Indivi-dual	Instance	scheme://authority/path/query/fragment
		http://ontology.tno.nl/saref/washingmachine#45609WQ
		http://ontology.tno.nl/saref/lightswitch#890-09w
		http://ontology.tno.nl/saref/energymeter#EM5004-stv-16

Table 3. Analogy between Descriptive Logic, OWL-DL and SAREF (Own elaboration based on [6] [7] [9] [10] [11] [12] [13] [14])

The last aspect to consider in the various existing devices or artifacts is their exemplification as "individuals" from a descriptive logic perspective, their similarity in the OWL-DL web ontology language being an "instance" and just like in the SAREF ontology, its notation is done according to a Uniform Resource Identifier URI "scheme://authority/path/query/fragment" [15], where:

- scheme = specification that assigns identifiers such as a naming system [15], generally including the communication protocol for accessing a resource such as "http:", "ftp:", "mailto:", etc.
- authority = hierarchical element that identifies a domain name, through a nomination or registered title [15], as well as through the address of a server, for example, "//www.etsi.org", "//ontology.tn.nl", "//www.onem2m.org", etc.
- path = subordinate data that identifies a resource within the scope of the domain name and schema in the manner of a file system [15], as in "/deliver/etsi_ts", "/images/files/deliverables/Release2", "/saref/WashingMachine", etc.
- query = component whose data are not ordered by dependent levels but in the form

of “key = value” pairs [15] within a schema, domain and path determined from the question mark character “?”; for example, “/c=GB?objectClass?one” [15].

- fragment = element that allows the indirect identification of a secondary resource in reference to a main resource (schema and authority), based on a representation of the same, and indicated by the numeric sign character “#” [15]. This representation concerns a part, view, definition, or description of the main resource to which it belongs (i.e., the URI); its semantics are dependent on the action of retrieving such resources [15].

It should be noted that the format and resolution of a fragment is dependent on the “media type” of the Multipurpose Internet Mail Extensions MIME [15], which corresponds to the “application” category and is implemented to process discrete data by a program or application before it is available to an agent and is of practical use in file transfer [16].

The MIME media type used for a device instance falls into the “application/rdf+xml” category, because the syntactical and semantic information of said instance is defined in an “rdf/xml” resource [17], i.e., structuring the data in the Resource Description Framework (RDF) under the syntax of the eXtensible Markup Language (XML), which facilitates the encoding of such data in a file and transferring it over the network –a procedure also known as “serialization”– [17], as well as the description of said devices as resources and their respective interoperability over the network.

The RDF Resource Description Framework is used for the instantiation of devices, as resources with semantic descriptions that allow unambiguous communication, since it is a language to represent metadata about the Internet about such resources or elements [18], such as a document, an image, a video, an audio, a web page, among others; also, regarding tangible things –verbi gratia– a consumer item, a household appliance, and so on. This RDF representation is visualized as a graph consisting of a pair of nodes with an edge, in addition to being established from a triplet whose components are [19]:

- Subject (concept, class, node), denoted by a named or unnamed URI reference (empty node).
- Predicate (role, relationship, property, edge), denoted by a URI reference.
- An object (data, attribute, class, or node), denoted by a literal (typed) or URI reference, whether named or unnamed (empty node).

Encoding a device instance as a semantic descriptor resource from a syntax-correct and grammatically valid RDF/XML file allows for the exchange of information between disparate applications with common semantics [17], making this type of media (application/rdf+xml) neutral with respect to devices, platforms, and vendors, supported by both agents (user or machine) and integrated development tools [17].

The normative syntax for encoding any instance of a device in an RDF is RDF/XML [18], which makes it the most appropriate medium for communication between artifacts, as the Resource Description Framework (RDF) aims to make machine-processable statements. For the purpose of this description, several aspects must be considered [18]:

- A system of machine-processable identifiers to name or single out a subject, predicate, and object in a statement (i.e., an RDF triplet) such that there is no room for ambiguity when the identifier is used by another agent in cyberspace [18].
- A computer language that is machine-processable and thus can represent any statement (RDF triplet) and exchange it between various devices [18].

The first aspect mentioned concerns a Uniform Resource Identifier (URI), which allows the identification of anything on the network, such as an image, a service, a person, an abstract concept, an artifact, etc. [18]. The second aspect refers to the XML eXtensible Markup Language for RDF –XML/RDF–, which facilitates the elaboration of the semantic descriptor structure of any entity on the network (e.g., the instance of a device) and in said language to represent the RDF information of such entities and transfer them between machines [18].

However, XML/RDF is not the only serialization format for writing RDF triples, although various existing forms lead to exactly the same graphs and are therefore logically equivalent. These encoding formats are N-Triples, N-Quads, Turtle, TriG, JSON-LD, and RDFa [20], which are susceptible to conversion to XML/RDF by utility applications. Their development is due to the ease of expressing graphs in a manner similar to the syntax of SPARQL [20], which is the query language for RDF, allowing the development of search criteria and, therefore, inference rules.

Likewise, expressing the RDF graphs corresponding to the instance of a device with its various classes, relationships, and attributes leads to the construction of an ontology whose schema is feasible to serialize using Web Ontology Language OWL under the XML syntax, which results in the media type "application/owl+xml" [14] [21]. This category is –by W3C– in the process of registration and standardization in the Internet Assigned Numbers Authority IANA, so it has not yet acquired the status of technical specifications to be massively implemented by applications and development tools [22]. However, OWL/XML serialization is currently used because it allows the optimization of RDF in terms of simple and multiple inheritance associations (aggregation and composition) between classes, which is relevant to the semantic interoperability of the entities connected to the network [6] [23].

Finally, since a device instance is denoted by a URI, which is a compact sequence of characters that identifies an abstract or concrete resource, whose syntax is defined from a set of US-ASCII or UTF-8 symbols [15], it has become necessary to expand the character repertoire to successfully handle all possible unambiguous URI spellings, for which a protocol element called Internationalized Resource Identifier (IRI) [24] has been defined, with the UCS (Universal Character Set) being the set of characters encoded to represent resources in a generic way, facilitating the mapping between URI and IRI [24].

The descriptions within the SAREF ontology, both of the main concepts or classes, as well as of the object and data relationships or properties, are available in detail for consultation in the technical specification TS 103 264 V2.1.1 [11]. For reasons of space, they cannot be included in this research, and the reader is invited to review them if necessary.

2.3. **Ontology Smart Appliances Reference for the Energy SAREF4ENER**

The Smart Appliances REFerence ontology for the ENERgy domain, SAREF4ENER, version 1.0, is a scheme of concepts and main relationships that derives or extends from a higher ontology called the Smart Appliances REFerence ontology (SAREF), which is characterized by managing a specific discourse universe consisting of the interconnection of smart devices to cover demand response scenarios for energy on the smart grid [25].

The SAREF4ENER ontology was created in collaboration with Energy@Home (http://www.energy-home.it) and EEBus (http://

www.eebus.org/en), which is the main Italian-German association of the industry, to allow the interconnection of their respective data models [26] [27]. SAREF4ENER focuses on erecting and exposing a primitive conceptual system that satisfies the final consumption of electricity, facilitating customers to offer flexibility to the Smart Grid to manage their smart home devices through a Customer Energy Manager (CEM). The CEM is a logical function that allows the spending and/or production of energy to be optimized and can reside both in a domestic gateway and in the cloud [27].

The various scenarios covered by SAREF4ENER ontology are described based on the following use cases:

- **Use case 1:** Concerns about the configuration of devices that want to connect to each other in the home network; for example, registering a new dishwasher in the list of devices managed by the CEM client energy manager [25] [26] [28], as shown in Figure 4.

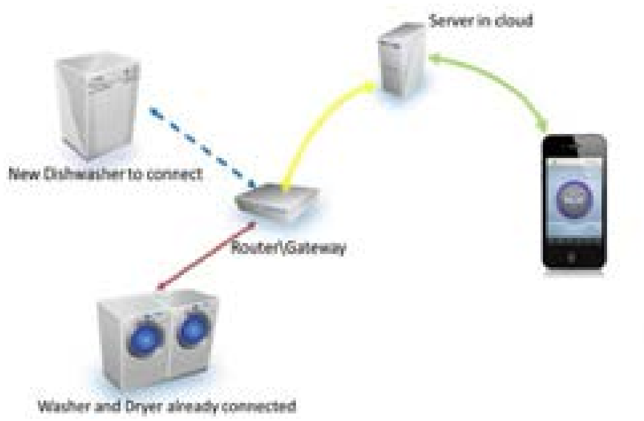

Figure 4. Use case 1: Registration of a new device, in the SAREF4ENER ontology [25] [28].

The new device must become visible to the CEM client's energy manager to be added to the list of devices to be managed (known as "provisioning"). The new machine must provide a CEM with both its ID and capability [28]. The mobile application must also provide the user with simple guidelines on how to complete the provisioning process, that is, the registration process [28].

- **Use case 2:** Intelligent power management by (re)programming devices to certain modes and chosen times, using power profiles to optimize energy efficiency and meet customer preferences [25][26][28], according to Figure 5.

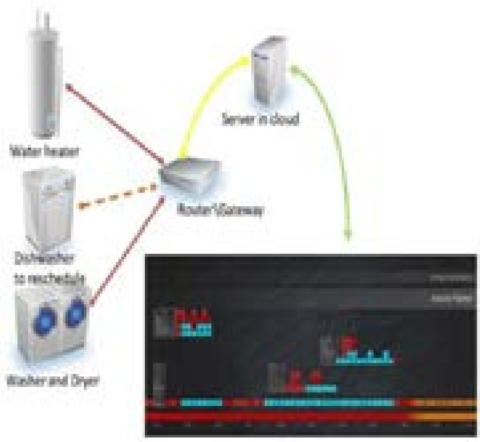

Figure 5. Use case 2: Device reprogramming using power profiles, in the SAREF4ENER ontology [25] [28].

Device scheduling is performed when the CEM receives information from the appliance that it is ready to start either in the remote-control mode or manually. In their mobile application, the user can see the tariffs for the day and the incentive areas for energy consumption or savings, which allows them to plan when to start the appliance remotely and see which another machine is already running or is planned to start [28].

Automatic reprogramming of the device is also possible, as the CEM receives information from the utility company or intermediary, which indicates that there is a more convenient time for the device to perform its task. In the mobile application, the user can see the new schedule for the start of the chosen appliance and determine a new plan to avoid peak hours [28].

- **Use case 3:** Monitoring and controlling the startup and status of devices [25] [26] [28] (see Figure 6).

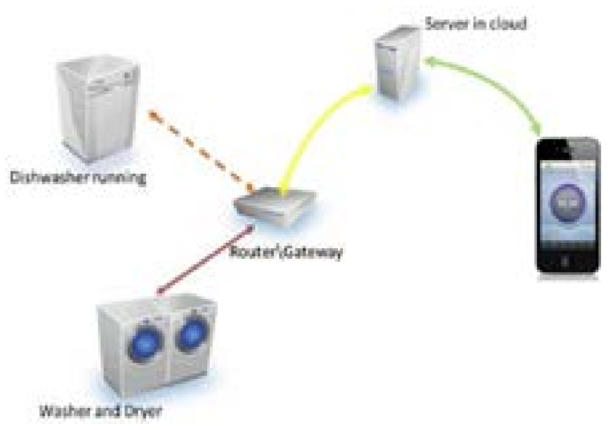

Figure 6. Use case 3: Monitoring and controlling the status of a device, in the SAREF4ENER ontology [25] [28].

Information about the status of a device, for example, a dishwasher (states such as on/off, time remaining, power profile, scheduled start time, etc.), is provided to the CEM client energy manager, even if the machine has been manually set to start [28].

- **Use case 4:** response to special requests from the smart grid, for example, incentives for lower consumption depending on current energy availability (Figure 7) or in emergency situations that require a temporary reduction in electricity consumption in terms of power [25] [26] [28].

The SAREF4ENER smart device reference ontology for energy is an initiative derived from the work of Daniele et. al, commissioned by the Netherlands Organization for Applied Scientific Research (TNO), on the primitive conceptual model SAREF, who developed the first version called SAREF4EE [28]. The continuation of this effort was undertaken by the European Telecommunications Standards Institute ETSI the following year, developing the technical specification ETSI TS 103 410-1 and publishing it two years later [26], under the title “Smart M2M; Smart Device Extension for SAREF; Part One: Energy Domain”, taking charge of both its maintenance and its implementation and annual update.

Figure 7. Use case 4: Response to requests from the Smart Grid, in the SAREF4ENER ontology [25] [28].

The semantic representations within the SAREF4ENER ontology are essentially structured as shown in Figure 8, developed by the author based on [26] [27] [28] [29], and in accordance with the guidelines of the OWL-DL Language, in which the “concepts and roles” of descriptive logic are assimilated to “classes and properties”, the latter being both attributes and relations, respectively [6].

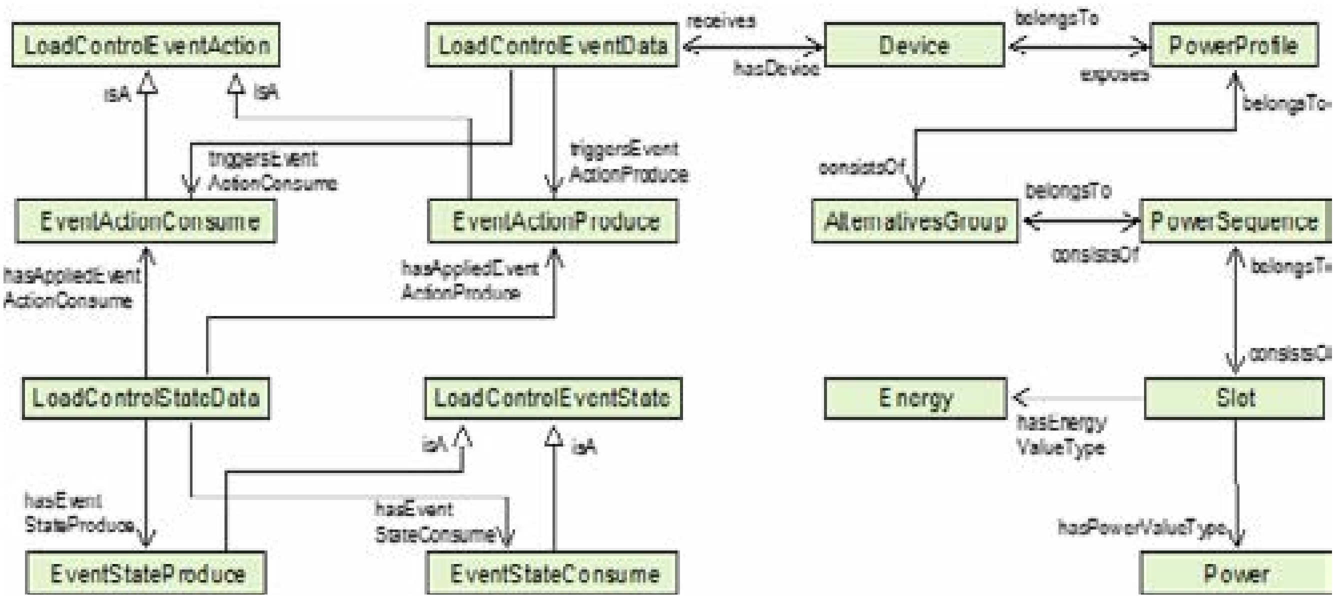

Figure 8. Basic conceptual structure of the SAREF4ENER Ontology

(Own elaboration based on [26] [27] [28] [29]).

The notation of these concepts or classes within the SAREF4ENER ontology is established by the prefix "s4ener", which is separated from the ending that indicates the concept referred to using the orthographic symbol ":", for example "s4ener:PowerProfile", "s4ener:LoadControlEventData", "s4ener:Device", etc. [26]. Likewise, the roles or properties have the writing "prefix:relation" (also called "ObjectProperties") or "prefix:attribute" (named as "DataProperties"), whether it is a property of the relation category, or of the attribute category, as appropriate [12] [26]. There are, for example: "s4ener:hasEnergyValueType", "s4ener:hasEventStateConsume", as well as "s4ener:belongsTo", etc., for the relationships. Likewise, the "s4ener:sequenceRemoteControllable", "s4ener:slotActivated", "s4ener:hasTimeStamp", etc., for the attributes, because the latter are characteristics of a predefined data type (text, integer, etc.) that may vary in terms of content [12].

All of the above were established by ETSI through a Technical Committee, which defined a strategy to identify and develop SAREF ontology derivations in addition to the energy demand response (or energy domain), such as the environment, buildings, e-health/ageing, and agriculture [25]. Information on the requirements of these domains (known as extensions) and the definition of the guidelines for their maintenance and publication were summarized in the technical report ETSI TR 103 411 of 2017 with the nomination "SmartM2M; Smart Appliances; SAREF extension investigation", which facilitates the collection of requirements and the inclusion of the contributions of the main industry players regarding machine-to-machine technology [25].

Both the object and data classes and properties belonging to the SAREF4ENER ontology can be consulted in detail in technical specification TR 103 411 V1.1.1 [25]. The readers of this research are encouraged to examine this document, available on the web, to learn more about its peculiarities.

2.4. Monadic Machine to Machine Ontology oneM2M

This conceptual specification has been designed by oneM2M, which is a partnership project between ICT standards development organizations worldwide. Since 2012, it has acted as an initiative for the standardization of Machine-to-Machine M2M communications and the Internet of Things IoT [30].

The purpose of oneM2M is to present in an orderly and comprehensive manner the components, characteristics, and operation of the machine-to-machine digital information exchange [31] in such a way that they direct the fulfillment of the needs of a common service layer between machines, such as low-energy consumption devices. This layer must be integrated in a simple manner into various computer technologies (hardware and software) [30], with the conviction of communicating fully and reliably the enormous number of electronic devices existing in the world with application servers M2M without human intervention [31].

In Section 2, a detailed definition of the notion of ontology can be found; however, both Grønbæk and oneM2M allow us to state that ontology is a vocabulary [32] with a structure within an API that is applied to a specific domain of interest (for example, measurement, household appliances, medicine, etc.), containing concepts that are used within the said universe of discourse [33] [34].

Since an ontology should capture a shared understanding of a domain of interest and simultaneously provide a formal model of that domain that is interpretable by any machine [33] [34], it is necessary to name and enumerate the concepts relevant to the domain in question with well-defined meanings. For this reason, the concepts are called "Classes" according to the OWL standard [21] [23], known as Web Ontology Language, because the latter identifies the types, categories, or classes of individuals [33] [34].

According to the OWL, the structure of the ontology is established by well-defined and agreed relationships between its concepts [21] [23]; these relationships are known as the name of "ObjectProperty," which allow to link a concept of 'subject' with a concept of 'object' [33] [34]. The scheme of this structure is as follows.

Subject concept **==>** *Relationship* **==>** *Object concept*

In the Web Ontology Language, it would be:

Domain Class **==>** *Object Property* **==>** *Range Class*

An example of such a configuration according to the SAREF ontology is the following:

Device **==>** *Accomplish* **==>** *Task*

Another example under the oneM2M ontology:

Service **==>** *Exposes* **==>** *Function*

A synopsis of the ontological structure and its examples regarding the object-property relationship is presented in Table 4.

Ontology	Scheme (of linkage)
Nominal	*Subject Concept* ==> *Relationship* ==> *Object Concept*
OWL	*Domain Class* ==> *Object Property* ==> *Range Class*
SAREF	*Device* ==> *Accomplish* ==> *Task*
oneM2M	*Service* ==> *Exposes* ==> *Function*

Table 4. Relationship between concepts within an ontology in terms of *Object*

***Property* (Own elaboration).**

There is a second type of property in OWL known as "DataProperty", which allows linking a piece of information to a Subject Class [33] [34]. It is optional for said data to have a typology, but it is recommended because it facilitates its management. The respective models are presented below:

Domain Class **==>** *Data Property* **==>** *Range Class*

An example according to the SAREF ontology is the following:

Device **==>** *hasManufacturer* **==>** *Literal*

And according to the oneM2M ontology:

Operation **==>** *(Provide the) Method* **==>** *PlainLiteral*

Table 5 shows a summary of the ontological structure and its examples regarding the data-property relationship.

Ontology	Scheme (of linkage)
Nominal	*Subject Concept ==> Relationship ==> Object Concept*
OWL	*Domain Class ==> Data Property ==> Range Class*
SAREF	*Device => hasManufacturer => Literal*
oneM2M	*Operation ==> (Provide the) Method ==> PlainLiteral*

Table 5. Relationship between concepts within an ontology regarding *Data Property* (Own elaboration).

Because every ontology is susceptible to being interpreted by machines and for which it must be codifiable in a computer language, this characteristic –as already mentioned in Sections 2.1 and 2.2 above–, allows the building of data search sentences that meet one or several criteria (e.g., the SPARQL query language) [20], thereby obtaining a set of individuals or instances of classes with specific relationships. These sentences are elaborated following the rules of descriptive logic and/or predicate logic, which is sufficient reason for the oneM2M ontology to have been codified in OWL, ensuring that such arguments or syllogisms are decidable [33] [34].

Another particularity of oneM2M ontology and its representation in OWL, concerns the provision of syntactic and semantic interoperability with external systems that have their respective universe of discourse encoded in said language [33] [34], thereby guaranteeing the mapping of the same –in terms of concepts and relations–, to the base ontology of oneM2M through a subclassification, an equivalence, etc., by facilitating the characterization of specific types of devices (e.g., in the SAREF ontology) or, more generally, the description of "Things" of the real world (such as buildings, rooms, cars, cities, etc.) within the scope of oneM2M with their respective grammaticality and meaning [33] [34].

When referring to syntactic interoperability, concerns to interworking with non-oneM2M devices in both local and area networks. This is achieved by means of an ontology represented as an OWL file, containing the typing of network-specific communication parameters –for example, operation names, input/output parameter names, their types and structures, etc.– which is used for the configuration of an Interworking Proxy Entity (IPE) [33] [34], facilitating the allocation of oneM2M system resources such as an Application Entity (AE) or a container, to perform activities that read from –or write to– these resources, so that the IPE can serialize the data and send it to –or receive it from– the devices in the network [33] [34].

As for semantic interoperability, this it applies to the description of functions that are provided by devices that are compatible with the oneM2M system –that is, Machine to Machine devices–, which are mapped into oneM2M resources such as "Containers" "resourceNames", "child-Resources" and "contentTypes" [33] [34].

An example of such semantic interoperability concerns the "washing machine" device, which, being compatible with oneM2M and having different versions of such artifacts, can perform functions such as "wash", "dry", and "select temperature", regardless of the fact that these tasks have different resource names, descendant structures, and content types, which are defined as resources of the oneM2M system [33] [34]. Therefore, an OWL file corresponding to the ontology contains the previous functions grouped in a common or application service offered by the M2M device.

From the above, a relevant aspect of the oneM2M ontology emerges, which concerns the interpretation of a class, not only as a representation of a concept or as a set of individuals, but also its definition through the distinctive properties –or relations– of the instances of said class. This method is known as "restriction". The classes that are specified from a restriction are called "anonymous classes", which group all instances that satisfy such conditions [33] [34]. There is a classification of restrictions namely: universal, existential, and cardinality.

• The universal constraint describes a domain or class of individuals that, for a given property, has only one relationship according to that property, with individuals that are members of a range class. For example, given that a subclass "Water valve" of the superclass "Device" (Class:Device), has only one function or task (ObjectProperty:hasFunction) named "Open or close valve" and belonging to the subclass "Function" (Class:Function), then the subclass "Water valve" (Class:Watervalve) is a superclass of the anonymous class formed by the object property "Has a function" (ObjectProperty:hasFunction) and that has a single relationship with the class "Open or close valve" (Class:Open_or_Close_Valve) [33] [34].

• The existential restriction details the class of individuals –or starting set–, which participates in at least one relationship according to a given property, with individuals that are members of the codomain class. For example, because a class "Device" (Class:Device) has at least one function or activity (ObjectProperty:hasFunction), in relation to the class "Function" (Class:Function) and that said artifact performs, then the device in question (Class:Device) is a subclass of the anonymous class constituted by the object property "Has a function" (ObjectProperty:hasFunction) and that has –at least– some association with the class "Function" (Class:Function) [33] [34].

• The cardinality restriction refers to the class of individuals that acts as a domain –or defining set– and that, for a given property, only have a specific number of relationships along (or according to) that property, with individuals that are members of the class that act as a range or ending set [33] [34]. For example, if for a class "Service" (Class:Service), there is an operation that is mandatory (Class:Operation), then the related object property "Has an operation" (ObjectProperty:hasOperation), will have a property restriction with a cardinality of "only one" [33] [34].

The final facet of the oneM2M system concerns its Base Ontology (BO), which is designed to provide the minimum number of concepts, relationships, and constraints required for the semantic discovery of entities with their corresponding representation in classes, object properties, and constraints (anonymous classes); that is, in oneM2M resources, also called containers. The detection of the elements (or entities) belonging to the ontology of a oneM2M technology –with their respective specificity– by said BO, must also be performed through the descriptive connotation of the same as containers and then associate them as subclasses and by analogy, with the classes already predefined in the Base Ontology, which facilitates ensuring semantic interoperability [33] [34].

Likewise, the design of the Base Ontology provides a framework for the creation of ontologies that belong to "non-oneM2M" technologies, which, by describing their own data model in containers (classes, object properties, and restrictions), allows for syntactic interoperability; in other words, strengthening the interworking with and between such non-oneM2M devices, making use of the Interworking Proxy Entity IPE inherent to oneM2M [33] [34].

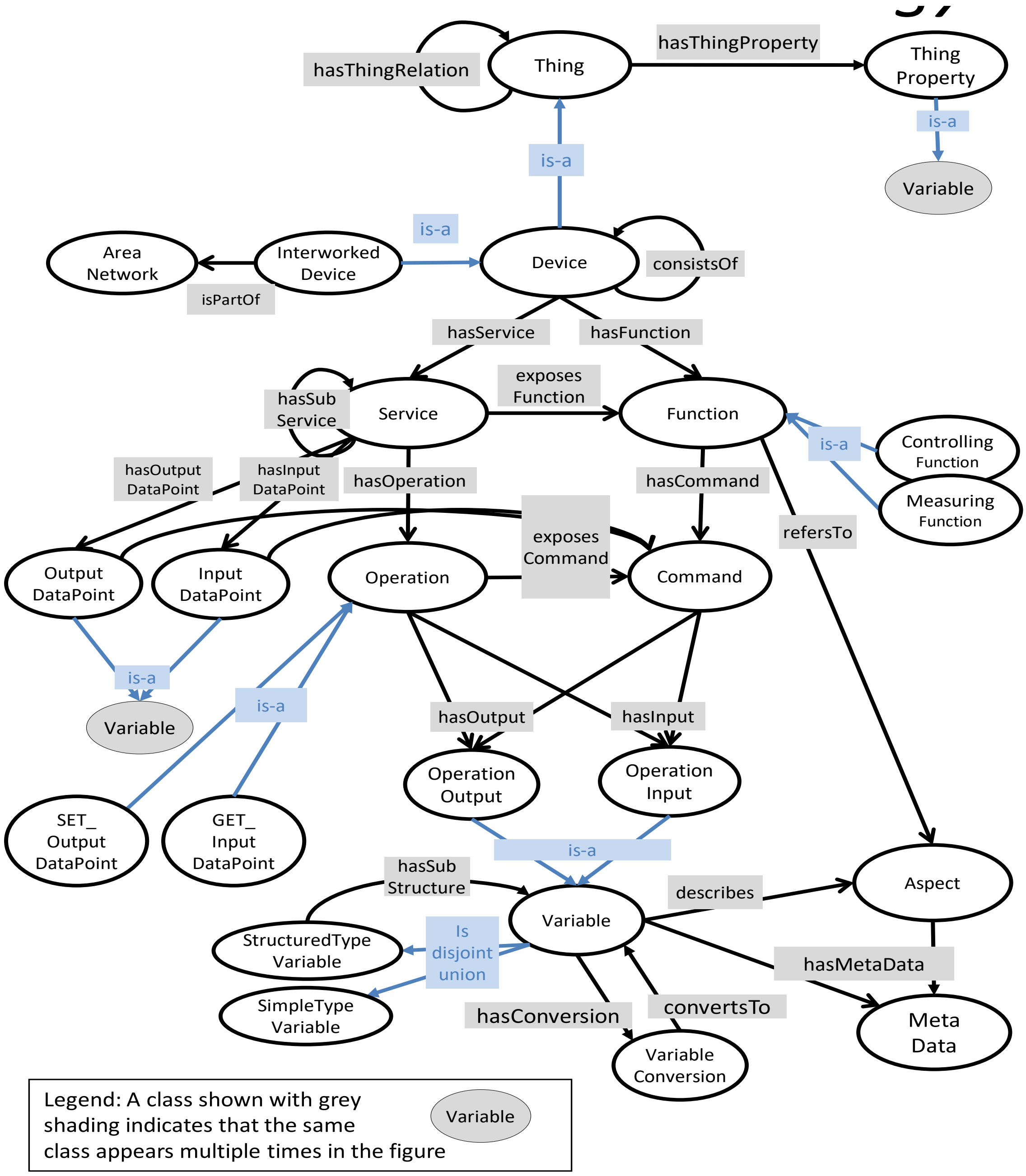

Figure 9. oneM2M Base Ontology or fundamental [33] [34].

Additionally, the oneM2M Base Ontology was developed in the OWL Web Ontology Language and is available in a cloud repository [35], with its respective updates; therefore –and for the purposes of this study–, this specification is available at the web address "https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl", with its corresponding schematization in Figure 9 as a graph, facilitating the visualization of the concepts and relationships that make up this Base Ontology; id est, the classes and their object properties [33] [34]. These concepts are denoted by ellipses (nodes) and the relationships are indicated by arrows (edges). It should be noted that in the illustration, a class with gray shading and the name "Variable" can be seen, meaning that this concept is repeated multiple times in the oneM2M model.

A detailed description of both the object and data classes and properties concerning the oneM2M ontology is available online through technical specification TS-0012-V3.7.3 [33]. As with previous ontologies, space reasons do not facilitate their inclusion in the present research work, so the reader is invited to consider consulting the said document.

3. Methodology

3.1. Framework definition

Because the research design is conceived to obtain the desired information [36], the framework is ascribed –in terms of software development– to what is the Unified Process, both in its initial phase, as well as in its elaboration, construction, and transition phase [37]. Likewise, this strategy corresponds to the simulation of a project, given that its definition of being "the process of designing a model of a real system and performing experiments, with this model in order to understand the behavior of the system and/or evaluate various strategies for the operation of the system", is the most appropriate for achieving the proposed objectives [38].

The linkage between the Unified Software Development Process and the simulation process is shown in Table 6.

Unified Software Development Process [37]	**Systems Simulation Process** [38]
Inception Phase.	Steps 1 to 6 (Problem Definition, Project Planning, System Definition, Conceptual Model Formulation, Preliminary Experimental Design, Input Data Preparation).
Elaboration Phase.	Steps 7 to 8 (Model Translation, Verification and Validation).
Construction Phase.	Steps 9 to 10 (Final Experimental Design, Experimentation or Simulation).
Transition Phase.	Steps 11 to 12 (Analysis and Interpretation, Implementation and Documentation).

Table 6. Articulation between the Unified Software Development Process [37] and the systems simulation process [38] (Own elaboration).

The breakdown of the steps mentioned in Table 6, is as follows:

3.2. Inception Phase

Steps 1–6 of the system simulation process [38] comprise the initial phase of the research design [37], whose application in the present research is broken down as follows:

3.2.1. Problem Definition – Step 1

As formulated in [38], one must “clearly define the objectives of the study so that we know the purpose, that is, why is this problem being studied and what questions are expected to be answered?”.

The situation concerns the need to manage greenhouse gas emissions – specifically CO2 – in cities to enhance the tasks of mitigate global warming and, therefore, climate change. One of the ways to carry out such management is to optimize the energy efficiency of smart devices through the Internet of Things (IoT), because the improvement in the appropriate consumption of these machines in terms of their power source –the electric energy generated by the power plants–, would allow such complexes to produce only the energy required to carry out the functions of the devices in question, regulating by default and over time, the concentration of carbon dioxide in the environment; since it is known that it is the power plants that release CO2 into the air when generating energy at the time of the transformation of the fossil fuel used for their production.

3.2.2. Project Planning – Step 2

According to what is stated in [38], it must be ensured that there are sufficient and adequate staff, administrative support, hardware, and software resources to do the work.

Because the research project is scientific and reflective in nature, the work is carried out individually by the author, training in the areas of knowledge necessary to achieve the final purpose, assuming the administration costs are both fixed and variable over time, and obtaining the computing resources necessary to carry out the experimental phase of the project in terms of data processing and transfer.

3.2.3. System Definition – Step 3

Considering what is expressed in [38], the limits and restrictions that will be used to define the system (or process) must be determined, and how it works must be investigated.

This end-to-end system works from a client (intelligent agent) consisting of an autonomous device that connects to the Internet or, failing that, an electronic component that connects to a gateway, which is in turn also connected to an M2M server via the Internet, and optionally, a terminal (tablet, mobile phone, laptop, etc.; i.e., a user agent or an app) that connects via the Internet to the M2M server to manage the energy consumption of the client at the opposite end (i.e., the intelligent agent).

The system is intended to dispense with the direct participation or intervention of humans in its operation, therefore the user agent must be avoided as far as possible, since the intelligent agent (client), being a software entity resident in a machine and communicating with another machine, which offers a service or set of services (server agent) through a gateway (intermediate agent), carries out actions based on the perception of its environment or in response to the data delivered to the “hosting”. For this project, the answer is to optimize the energy efficiency of the intelligent agent through the Internet of Things (IoT), which results in the management of CO2 emissions in cities.

Within the wide range of possibilities for designing the system, the oneM2M alliance has chosen the one developed to date, given that its work in generating standardized and open-source technical specifications in ICT has great potential worldwide to establish syntactical and semantic interoperability between various intelligent agents, regardless of their hardware and protocols.

3.2.4. Conceptual Model Formulation– Step 4

As set out in [38], a preliminary model must then be developed either graphically (e.g., block diagram or process flow diagram) or in pseudocode to define the components, descriptive variables, and interactions (logic) that constitute the system.

The conceptual model of the system for the research project –in the form of a pseudocode– is as follows:

- **Input**: If the intelligent agent is a sensor, the input is the deterministic data captured by the sensor. Otherwise, if the intelligent agent is an actuator, the input is stochastic data generated by a task programmed in the M2M server or by data taken by a sensor.
- **Process**: For an actuator, the energy required for its activation during the time it must execute its respective task is recorded. For a sensor, the energy required to remain active in the task of sensing a signal was recorded.
- **Output**: For a sensor, the output is the signal sent to the server and its respective record, in terms of the energy demanded. For an actuator, the output is the execution of the respective task with its corresponding record of the energy required to perform the task.

For this conceptual model to be effective, the syntactic and semantic interoperability mentioned in the previous step (Section 3.2.3) must be guaranteed, based on the definition of a corpus in the universe of discourse concerning the energy efficiency of smart devices, which facilitates the management of urban atmospheric carbon. In other words, the configuration of an ontology that acts as an inter-functional theoretical framework is necessary.

3.2.5. Preliminary Experimental Design – Step 5

As stated in [38], such a design consists of the selection of the effectiveness measures to be used, the factors to be varied, and the levels of those factors to be investigated, that is, what data should be collected from the model, in what form, and to what extent.

The development of step 5 is detailed in Section 4 of this research because it involves the compilation of the ontologies oneM2M, OM, SAREF4ENER, and SAREF, in terms of their respective links to web resources in OWL or RDF/XML format, with a fragment of those.

3.2.6. Input Data Preparation – Step 6

As formulated in [38], Step 6 involves identifying and collecting the input or input values to the system, that is, the input data required by the model.

The deployment of this step of the system simulation process is specified in Section 5, based on the ontology design methodology [39] and supported by the detailed study contained in Sections 2.1, 2.2, 2.3, and 2.4, regarding the ontologies oneM2M, SAREF4ENER, OM, and SAREF, respectively.

3.3. Elaboration Phase

Steps 7 and 8 of the system simulation process [38] establish the research design development phase [37], which is detailed below.

3.3.1. Model Translation – Step 7

Continuing with what was stated in [38], step 7 of the system simulation process concerns formulating the model in an appropriate simulation language or failing that in a Unified Modeling Language such as UML.

It is advisable to structure the model in UML, because it facilitates building the system and representing it in a visual or graphical way, given that the ontology to be proposed and simulated is composed of classes or concepts that are related or associated with each other, both at the level of properties and attributes in a specific domain. Likewise, UML allows mapping of the ontologies oneM2M, OM, SAREF4ENER, and SAREF, which are integrated to form the final ontology.

The development of Step 7 is detailed in Section 6, as it involves combining the ontologies mentioned under the UML.

3.3.2. Verification and Validation – Step 8

According to [38], in step 8 of the system simulation process, it is confirmed that the model works as intended by the researcher or analyst. The development of this step is detailed in Section 7. Once the ontological model observed in Figure 10 has been refined, the resulting configuration must be corroborated; that is, the proposed architectural model is functional, flexible, credible, and representative of the output of the real system.

3.4. Construction Phase

Steps 9 and 10 of the system simulation process [38] are consistent with the construction phase of the research design [37], whose breakdown is as follows:

3.4.1. Final Experimental Design – Step 9

As expressed in [38], in step 9 of the system simulation process, an experiment is designed to provide the desired information and determine how each of the test activities specified in the preliminary experimental design [37] will be executed; that is, the one corresponding to step 5 (item 4) in the inception phase.

The deployment of this step is specified in Section 8, starting with the identification of the classes of the lower level of the oneM2M hierarchy and their corresponding detailed associations with the SAREF4ENER, OM, and SAREF hierarchies, in the same way; that is, the classes or concepts of their respective lower levels.

3.4.2. Experimentation or Simulation – Step 10

Continuing with what was stated in [38], step 10 of the system simulation process, the experimentation must be executed to generate the desired data and perform a sensitivity analysis.

The development of this step is detailed in Section 9, since the simulation is carried out from the W3C validator for ontologies in OWL and/or RDF/XML, with their respective triplets of the proposed ontological model including its graph, which is the result of its grammatical analysis or **"parsing".**

3.5. Transition Phase

Steps 11 to 12 of the system simulation process [38] constitute the transition phase of the research design [37], whose application is detailed below.

3.5.1. Analysis and Interpretation – Step 11

As formulated in [38], step 11 of the system simulation process consists of extracting inferences about the data generated by the simulation or experimentation executions [37].

The deployment of this step is specified in Section 10, in which the deductions inevitably lead to the optimization of the energy efficiency of intelligent agents through the IoT and therefore, to the management of urban CO2 emissions.

3.5.2. Implementation and Documentation – Step 12

As expressed in [38], in step 12 of the system simulation process, the results must be reported, made available for use, the findings recorded, and the model and its implementation documented.

The final step (N°. 12) is specified in Sections 11 and 12, which consist of the preparation of the written document regarding design, simulation, analysis, results, conclusions, and future works.

4. Preliminary experimental design

Step 5 of the system simulation process [38] concerns the preliminary experimental design, which refers to the selection of the effectiveness measures to be used, the factors to be varied, and the levels of those factors to be investigated, that is, what data should be collected from the model, in what form, and to what extent.

For the above purposes, the various ontologies that constitute the proposed architectural model (oneM2M, SAREF4ENER, OM, and SAREF) in the OWL language are presented below. Given their extensive scope, only one fragment of each is included, accompanied by the corresponding web link.

4.1. oneM2M Ontology

Web address in OWL or RDF/XML format (also known as "raw") oneM2M ontology

https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl

Example fragment for the concept "Thing" [35]:

<Class rdf:about="https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#Thing">

<rdfs:subClassOf>

<Restriction>

<onProperty rdf:resource="https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#hasThingProperty"/>

<allValuesFrom rdf:resource="https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#ThingProperty"/>

</Restriction>

</rdfs:subClassOf>

<rdfs:comment>A Thing in oneM2M (Class: Thing) is an entity that can be identified in the oneM2M system…

</rdfs:comment>

</Class>

4.2. OM Ontology

Web address in OWL or RDF/XML (raw) format of OM ontology

https://raw.githubusercontent.com/HajoRijgersberg/OM/master/om-2.0.rdf

Example fragment for the concept "Unit" [4]:

<owl:Class rdf:about="&om;Unit">

<rdfs:label xml:lang="en">unit</rdfs:label>

<om:alternativeLabel xml:lang="en">unit of measure</om:alternativeLabel>

<om:alternativeLabel xml:lang="en">unit of measurement</om:alternativeLabel>

<rdfs:comment xml:lang="en">A unit of measure is a definite magnitude of a quantity, defined and adopted by convention or by law. It is used as a standard for ….

</rdfs:comment>

</owl:Class>

4.3. SAREF4ENER Ontology

Web address in OWL or RDF/XML format ("raw") of the SAREF4ENER ontology

: https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf

Fragment of the concept "LoadControlEventAction" [40]:

<owl:Class rdf:about="https://saref.etsi.org/saref4ener/LoadControlEventAction">

<rdfs:label xml:lang="en">Load Control event action</rdfs:label>

<rdfs:comment xml:lang="en">An action type used to express the action to be performed as a consequence of an event used to send overload...</rdfs:comment>

</owl:Class>

<owl:ObjectProperty rdf:about="https://saref.etsi.org/saref4ener/...">…

</owl:ObjectProperty>

4.4. SAREF Ontology

Web address in OWL or RDF/XML format (raw) of SAREF ontology:

https://saref.etsi.org/core/v3.1.1/saref.rdf

Example fragment for the concept "Property" [41]:

<owl:Class rdf:about="https://saref.etsi.org/core/Property">

<rdfs:label xml:lang="en">Property</rdfs:label>

<rdfs:comment xml:lang="en">A quality of a feature of interest that can be measured; an aspect of a… >

<rdfs:subClassOf>

<owl:Restriction>

<owl:allValuesFrom>

<owl:Class rdf:about="https://saref.

```
etsi...”/>
            </owl:allValuesFrom>…
         </owl:Restriction>
      </rdfs:subClassOf>…
   </Class>
```

5. Input data preparation

Step 6 of the system simulation process [38] refers to the preparation of the input data, that is, the identification and collection of the input or input values that the model needs.

Because these data are part of energy efficiency, understood as the appropriate consumption of electrical power to perform tasks –singular or diverse– by the device (i.e., energy spending), the configuration of its scope to optimize said spending concerns the specification of an ontological model, which –following the guidelines suggested in [39]–, is described below.

5.1. *Definition of purpose and scope*

Based on what is expressed in [39], the context and ultimate purpose of the ontological model must be established to answer the following three questions:

1.) What is the ontology about?

It deals with the management of CO2 emissions in cities.

2.) Why is this ontology being built?

The aim is to link any device to the IoT, regardless of its communication protocol (MQTT, CoAP, Rest, etc.), and to regulate its energy consumption.

3.) Who is it aimed at or who will be the users of the ontology?

The community of device developers (consumer electronics for home automation, smart buildings, automotive, instrumentation, etc.) and Machine to Machine (M2M) communication protocols.

5.2. Domain delimitation

As formulated in [39], the restrictions or terms intrinsic to the universe of ontological discourse must be considered, answering the triad of questions:

Q. N° 1) Define the scope; that is, what elements can be modeled with the available resources and which cannot be achieved?

Ans.: The control of the machine is not modeled, but rather monitored to reduce its energy consumption. This is because each manufacturer designs devices (machines) based on their properties (type, autonomy, energy consumption, etc.) and relationships (communication protocols, subscription and/or publication, among others).

Q. N° 2) Try to visualize; i.e., what data sources that allow populating the set of instances according to the model?

Ans.: Data are obtained directly from the devices when they connect to the network and report their energy consumption according to their performance.

Q. N° 3) Finally, identify the types of questions that can be answered with the ontology.

Ans.: Using the ontology, it is possible to answer the following questions:

- Which devices emit the greatest amount of CO2 according to their type and energy autonomy?
- What is the average energy consumption per unit time for a specific device?
- How much excess CO2 does a particular device emit?

5.3. Concepts Identifications

According to the statement in [39], the various notions or abstractions concerning the field of ontology (IoT, M2M, energy, etc.) must be distinguished from each other based on the following proposition: elaboration of a list of concepts present in the domain of ontology, such as individuals or entities (subjects or classes) and actions or relationships (predicates or behaviors).

- It should be noted that the properties or attributes of individuals correspond to monadic and/or polyadic first-order predicates when individuals are quantified. Likewise, the actions or relationships (or events) carried out, existing (or occurring) by, or between (or to) entities, concern higher-order predicates –both monadic and polyadic–, when estimating their quantity (i.e., subjects).

- List of concepts: device (machine), protocol (communication), consumption or spending (energy), autonomy (energy), subscription (to a service), publication (of a service), and CO2.

- The Internet of Things (IoT) is not a concept of ontology, since the IoT is the domain of the latter, that is, its environment, its order, its scope, its ecosystem.

- The energy consumption of a device is equivalent to the amount of carbon dioxide emitted into the environment by a machine during its operation (i.e., to perform its tasks).

5.4. Concepts Classification

In accordance with the formulation in [39], the abstractions determined in Section 5.3, must be catalogued, with their respective descriptions:

1.) Categorization of concepts by means of subsets based on common aspects. One concept can appear in more than one subset.

- Device: mobile or cell phone, tablet, phablet, laptop, PC, mp3 or mp4 player, etc.
- Service: publication, subscription, etc.
- Protocol: MQTT, CoAP, Rest, etc.
- GHG (Greenhouse Gas): CO2 (Carbon Dioxide), CH4 (Methane Gas), etc.
- Spending energy.
- Energy autonomy.

2.) Description of each subset.

- Device: Machine or consumer electronic equipment capable of connecting to the internet.
- Service: provision or assistance offered or requested by a device.
- Protocol: A set of communication rules agreed upon between devices.
- GHG: gaseous fluids that contribute to global warming when released into the Earth's atmosphere in large quantities.
- Spending energy: consumption of electrical energy by a device to perform its functions.
- Energy autonomy: amount of electrical energy that a device can store to perform its functions without requiring recharging.

3.) Obtaining an initial list of concepts with their descriptions (regardless of whether or not they match the subsets).

- Device: Machine or consumer electronic equipment capable of connecting to the Internet, such as mobile devices, tablets, laptops, desktop PC, mp3 or mp4 players, etc.
- Service: provision or assistance offered or requested by a device; for example, publication, subscription, etc.
- Protocol: set of communication rules agreed between devices, for example, MQTT, CoAP, Rest, etc.
- GHG (Greenhouse Gas): gaseous fluids that, when released into the Earth's atmosphere in large quantities, contribute to global warming; e.g. CO2 (Carbon Dioxide), CH4 (Methane Gas), etc.
- Spending Energy: consumption of electrical energy by a device to perform its functions. This is assumed to be a property of the device.
- Energy autonomy: amount of electrical energy that a device can store to perform its functions without requiring recharging. This is considered to be an attribute of machines.

5.5. Relationships Identification

As set out in [39], the following is the recognition of the links or connections existing between pairs of entities or individuals:

1.) One or more devices offer or demand one or more services.

2.) One or more devices establish or agree on one or more protocols.

3.) One or more devices emit or throw out one or more greenhouse gases.

4.) One or more devices possess or have one spending energy.

5.) One or more devices have or have an energy autonomy.

5.6. Model Outline

Continuing with what was expressed in [39], once the conceptual interrelations have been determined, the nature and derivations of said abstractions are outlined as follows:

1.) Definition of Attributes or characteristics (Data Properties).

2.) Determination of hierarchical relationships (Multiple and simple inheritance).

3.) Establishment of associations or behaviors (Object Properties).

5.7. Compilation and Evaluation

The last aspect to be covered in the preparation of input data[39] concerns grouping all the identified data (Sections 5.1 to 5.6) and estimating their relevance, coherence, and viability according to:

1.) Integration or combination of concepts, relationships or attributes.

2.) Verification and correction of the designed model.

The application of Steps 6 to 8 of [38], together with their respective iterative and incremental refinements, resulted in an ontological model, as shown in Figure 10.

6. Translation model

Step 7 of the system simulation process [38] corresponds to the translation model, which consists of the formulation of the model in an appropriate simulation language or failing that in a unified modeling language, such as UML.

As discussed in Section 5.7, it is advisable to structure the ontological model in UML, given its simplicity for building the system and its corresponding diagramming, since the fact that the ontology to be proposed and simulated is composed of classes or concepts related to or associated with each other, both at the level of properties and attributes in a specific domain. Likewise, UML allows mapping of the ontologies oneM2M, OM, SAREF4ENER, and SAREF, which are integrated to form the final ontology.

Everything related to the preparation of the input data detailed in Section 5 (step 6) is mapped into an ontology designed as a result and called “OntologyModelingIoTCo2”, which in turn associates its components with the ontologies oneM2M, SAREF4ENER, OM, and SAREF simultaneously. The resulting mapping of this macrostructure is shown below under UML in Figure 10.

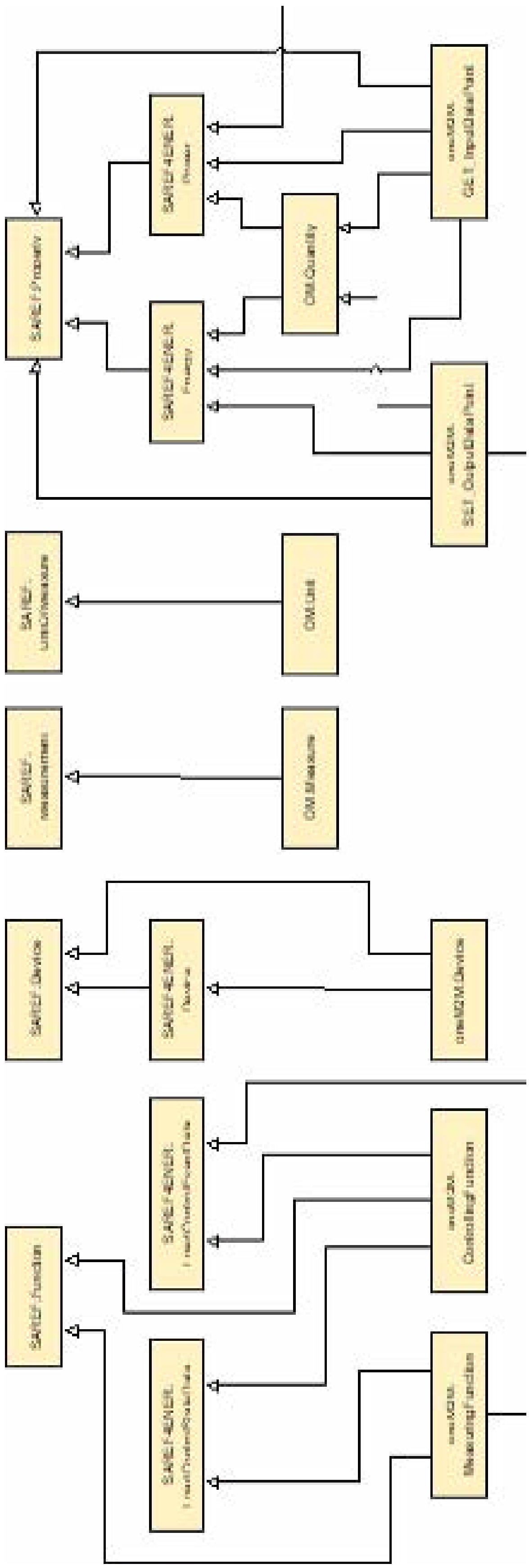

Figure 10. Basic conceptual structure of the proposed ontological mapping (Own elaboration), which becomes the class hierarchy of the mapping in question, based on the analysis of results (Section 10).

7. Verification and validation

Step 8 of the system simulation process [38] concerns verification and validation, that is, confirming that the model works in the way the analyst or researcher intended (debugging), and that the final model is credible and representative of the output of the real system.

The modeling of an IoT architecture to manage carbon emissions in cities concerns the initiative or proposal (or specification) to implement the oneM2M, SAREF, SAREF4ENER, and OM ontologies, integrating them in such a way that it is feasible to optimize the energy efficiency of electronic devices capable of communicating with the global computer network, given that the energy needed for said devices to perform the tasks for which they were designed has already been generated in advance in the respective power plants and mostly from non-renewable sources, that is, fossil fuels, which have in advance released carbon gas into the environment at the time of ignition, thus increasing the global greenhouse effect owing to its inherent climate change.

Based on the reference framework (Item 2) regarding the ontologies mentioned above, the proposed model answers three essential questions:
1.) What device should be managed in terms of energy consumption? for example, a switch, a light fixture, a thermostat, etc.
2.) What task does the device perform? that is, measurement (using a sensor) or control (through an actuator).

3.) What data is the device going to generate or retrieve?

- If the task is to be measured, then the device will generate (produce) data and deliver it (either to another device or a user).
- If the task is to be controlled, the machine retrieves (obtains) data and executes the programmed task.

Since every ontology is made up of concepts or classes and they are related by properties, these associations between classes allow establishing a hierarchy in which the notion of inheritance – whether simple or multiple – facilitates the identification of levels in terms of superclasses and subclasses within said categorization. Therefore, at the time of instantiation (individuals or objects) of the classes –that is, the creation of the devices per se with their peculiarities–, it is most appropriate to create said instances from the subclasses that are at the lowest level of the hierarchy.

Given that the class hierarchy that models the ontologies combined with each other must satisfy the questions formulated above, it has been identified that the oneM2M ontology presents the classes that serve as answers to the three questions as follows:

- Question 1 (Which device?):
  • Answered by the "oneM2M:Device" class.
- Question 2 (Which device task):
  • Answered by the "oneM2M:ControllingFunction" and "oneM2M:MeasuringFunction" classes.
- Question 3 (Which data to generate or retrieve?):
  • Answered by the "oneM2M:GET_InputDataPoint" and "oneM2M:SET_OutputDataPoint" classes.

The classes indicated and belonging to the oneM2M ontology are located at the lowest level of the respective hierarchy. Therefore, from instances of such classes, all the characteristics and properties of each of the superclasses that make up the hierarchy are accessed, regardless of the level at which they are located and the type of inheritance, whether simple or multiple.

Likewise, by integrating the oneM2M ontology with the other ontologies (SAREF, SAREF4ENER and OM), through the association with the subclasses that have equivalence in terms of their role and that are also at the lowest level of their respective hierarchies, the attributes and behaviors of the super-classes that constitute such hierarchies will also be inevitably accessed and therefore, it will be possible to manage the energy efficiency of any electronic device linked to the global computer network, by being able to record its functions, its profiles, its magnitudes, its measurements, its states, its services, its events, its power consumption according to time windows, its operations, etc., all of this immediately, without any delay; i.e., similar to the "hot swapping" of hardware components.

The associations between the various ontologies from the subclasses of the lowest level of each hierarchy are listed below as "ontology_name:class_name", indicating the hierarchy levels with the symbol "**==>**" or the opposite, indicating a superclass hierarchy level with the sign "**<==**", which can be corroborated by observing Figure 10.

1.) "oneM2M:Device" **==>** "SAREF4ENER:Device"
2.) "oneM2M:Device" **==>** "SAREF:Device"
3.) "oneM2M:ControllingFunction" **==>** "SAREF4ENER:LoadControlEventData"
4.) "oneM2M:ControllingFunction" **==>** "SAREF:Function"
5.) "oneM2M:ControllingFunction" **==>** "SAREF4ENER:LoadControlStateData"
6.) "oneM2M:MeasuringFunction" **==>** "SAREF4ENER:LoadControlEventData"
7.) "oneM2M:MeasuringFunction" **==>**

"SAREF:Function"
8.) "oneM2M:MeasuringFunction" **==>** "SAREF4ENER:LoadControlStateData"
9.) "oneM2M:SET_OutputDataPoint" **==>** "SAREF:Property"
10.) "oneM2M:SET_OutputDataPoint" **==>** "SAREF4ENER:Energy"
11.) "oneM2M:SET_OutputDataPoint" **==>** "SAREF4ENER:Power"
12.) "oneM2M:SET_OutputDataPoint" **==>** "OM:Quantity"
13.) "oneM2M:GET_InputDataPoint" **==>** "SAREF:Property"
14.) "oneM2M:GET_InputDataPoint" **==>** "SAREF4ENER:Energy"
15.) "oneM2M:GET_InputDataPoint" **==>** "SAREF4ENER:Power"
16.) "oneM2M:GET_InputDataPoint" **==>** "OM:Quantity"
17.) "OM:Unit" **==>** "SAREF:UnitOfMeasure"
18.) "OM:Measure" **==>** "SAREF:Measurement"
19.) "SAREF4ENER:Energy" **<==** "OM:Quantity"
20.) "SAREF4ENER:Power" **<==** "OM:Quantity"
21.) "SAREF4ENER:Energy" **==>** "SAREF:Property"
22.) "SAREF4ENER:Power" **==>** "SAREF:Property"

8. Final experimental design

Step 9 of the system simulation process [38] concerns the final experimental design, that is, developing an experiment that provides the desired information and determines how each of the test executions specified in the experimental design will be carried out.

To access all the concepts of the oneM2M ontology, the classes that are at the lower level of its own hierarchy are identified, which refer to:

- "Class:MeasuringFunction"
- "Class:ControllingFunction"
- "Class:Device"
- "Class:SET_OutputDataPoint"
- "Class:GET_InputDataPoint"

From the previous classes of the oneM2M ontology and by associating them as subclasses to the SAREF4ENER, OM, and SAREF ontologies, through the classes shown in Table 7 –the latter being configured as superclasses–, absolute access is guaranteed to the properties and characteristics (i.e., behaviors and attributes) of all the classes that make up each of the hierarchies belonging to each ontology.

In the oneM2M ontology, the Class:	**It acts as a subclass of the superclass… in the ontology:**		
	SAREF-4ENER	**SAREF**	**OM**
Device	Device	Device	------
Controlling Function	LoadControl EventData and LoadControl StateData	Function	------
Measuring Function			
SET_ Output DataPoint	Energy and Power	Property	Quan-tity
GET_Input DataPoint			

Table 7. Classes chosen to be associated from the oneM2M ontology with the SAREF4ENER, OM and SAREF ontologies, according to their respective hierarchies (Own elaboration).

Therefore, any instance, individual or object whose energy efficiency is to be managed, is associated with one of the classes of the oneM2M ontology in terms of entity (Device), whether in relation to controlling or measuring (Device Function), or finally, in relation to

generating or recovering data (Input or Output thereof), in accordance with the three aspects mentioned –in Section 7–, as questions of the proposed model.
However, to confirm the versatility and robustness of the ontology that integrates those already mentioned, Table 8 maps thoroughly the classes of the oneM2M Base Ontology and the classes of the SAREF4ENER, OM, and SAREF ontologies with their corresponding arguments:

Base Ontology Class	Mapped ontology class	Mapping argumentation
oneM2M: Device (Sub Class)	SAREF4 ENER: Device (Super Class)	In all three ontologies, a Device is conceived as an artifact that performs a specific task –known as a function–, based on generated or retrieved data. This makes it easier to associate, through multiple inheritance, the “Device” subclass of the oneM2M ontology with the two “Device” super classes, both belonging to the SAREF4ENER and SAREF ontologies respectively.
	SAREF: Device (Super Class)	
oneM2M: Controlling Function (Sub Class)	SAREF4 ENER: LoadControlEventData (Super Class)	The control work that a device must perform allows it to be activated and to influence the environment. At that moment, the power produced by said device must be recorded, by controlling the load of one of the events mentioned with its respective data, as well as the state or condition in which said machine is found, which depends on the data that enters it at the given time. This is achieved by associating through multiple inheritance, from the “Controlling Function” subclass of the oneM2M ontology, to the “Load Control State Data” and “Load Control Event Data” superclasses, which are part of the SAREF4ENER ontology.
	SAREF4 ENER: LoadControlStateData (Super Class)	
	SAREF: Function (Super Class)	The control function is conceived as the ability of the device to perform a task, in this case to control, operate or act in an environment in which said artifact is immersed. Therefore, associating through multiple inheritance from the “Controlling Function” subclass of the oneM2M ontology to the “Function” superclass of the SAREF ontology allows access to the characteristics and properties of the latter, related to said faculty of the device.

oneM2M: Measuring Function (Sub Class)	SAREF4 ENER: LoadControlEventData (Super Class)	The measurement task that a device must perform allows it to be activated to perceive the environment. At that moment, the energy consumed by said device must be recorded by controlling the load of one of the events mentioned with its respective data, as well as the state or condition of said machine, which is subject to the data generated at the same given moment. This is achieved through the multiple inheritance association, from the “Measuring Function” subclass of the oneM2M ontology, to the “Load Control State Data” and “Load Control Event Data” superclasses, which are part of the SAREF4ENER ontology.
	SAREF4 ENER: LoadControlStateData (Super Class)	
	SAREF: Function (Super Class)	The measurement function is conceived as the ability of the device to perform a task, in this case to perceive, measure or capture in an environment in which said machine is immersed. Therefore, associating through multiple inheritance from the “Measuring Function” subclass of the oneM2M ontology to the “Function” superclass of the SAREF ontology facilitates access to the attributes and behaviors of the latter, related to said competence of the artifact.

oneM2M: GET_Input DataPoint (Sub Class)	SAREF4ENER: Energy (Super Class)	The process of reading or retrieving data entering an activating device involves recording both the energy that said device consumes and the power that it produces, and simultaneously executing the operation of entering such data at a specific time or window. This routine is achieved through the multiple inheritance association, from the "Get Input Data Point" subclass of the oneM2M ontology, to the "Energy" and "Power" super classes belonging to the SAREF4ENER ontology.
	SAREF4ENER: Power (Super Class)	
	OM: Quantity (Super Class)	The data that is read when entering an actuator type artifact, concerns a value, whether numeric, text, boolean, etc., that results from a measurement or operation with respect to a magnitude, in a pre-established universe of discourse. Said value is typified by means of the association of multiple inheritance type from the subclass "Get Input Data Point" of the oneM2M ontology, to the superclass "Quantity" concerning the OM ontology.
	SAREF: Property (Super Class)	The magnitude used by the activating device corresponds to a physical property that can be measured, whether it is the energy consumed and/or the power produced by said machine when reading or retrieving the data and performing the programmed task. Therefore, the association through multiple inheritance, from the "Get Input Data Point" subclass of the oneM2M ontology, to the "Property" superclass of the SAREF ontology, facilitates the quantification of the magnitude used in the entry of data to the device with the respective triggering of its action.

oneM2M: SET_ Output Data Point (Sub Class)	SAREF4 ENER: Energy (Super Class)	The process of updating or generating the data delivered by a sensor device involves recording both the energy that said device consumes and the power that it produces, and simultaneously executing the operation of outputting said data at a specific time or window. This routine is achieved through the multiple inheritance association, from the “Set Output Data Point” subclass of the oneM2M ontology, to the “Energy” and “Power” super classes belonging to the SAREF4ENER ontology.
	SAREF4 ENER: Power (Super Class)	
	OM: Quantity (Super Class)	The data that is updated at the time of delivery by a sensor-type artifact, concerns a value, whether numeric, text, boolean, etc., that results from a measurement or operation with respect to a magnitude, in a pre-established universe of discourse. Said value is typified by means of the association of multiple inheritance type from the subclass “Set Output Data Point” of the oneM2M ontology, to the superclass “Quantity” concerning the OM ontology.
	SAREF: Property (Super Class)	The magnitude used by the sensor device corresponds to a physical property that can be measured, whether it is the energy consumed and/or the power produced by said machine at the time of updating or generating the data and performing the programmed task. Therefore, the association through multiple inheritance, from the “Set Output Data Point” subclass of the oneM2M ontology, to the “Property” superclass of the SAREF ontology, facilitates the quantification of the magnitude used in the delivery of data by the artifact with the corresponding triggering of its perception task.

OM: Unit (Sub Class)	SAREF: UnitOf Measure (Super Class)	The expression corresponding to the measurement of a physical property or magnitude is susceptible to classification in some fundamental or derived unit of the International System of Units SI. This categorization is specified from the simple inheritance type association, from the "Unit" subclass of the OM ontology, to the "Unit of Measure" superclass of the SAREF class. Likewise, within the OM ontology, the simple inheritance type association is presented between the "Quantity" subclass and the "Unit" superclass. Similarly, in the SAREF ontology there is the relationship or "Object Property" between the "Measurement" and "Unit of Measure" classes (Measurement). Both of these peculiarities –although not shown in Figure 10 for reasons of clarity– facilitate access to all the characteristics and properties of such ontologies, since they are available from the "Set Output Data Point" and "Get Input Data Point" subclasses of the oneM2M ontology, when associated with the "Quantity" subclass of the OM ontology.
OM: Measure (Sub Class)	SAREF: Measure ment (Super Class)	The comparison of a quantity with its respective unit, in order to determine a magnitude, facilitates the measurement of the same, i.e., its dimensioning. This estimation work is formalized from the simple inheritance type association, from the "Measure" subclass of the OM ontology, to the "Measurement" superclass of the SAREF class. Additionally, within the OM ontology, the relationship or Object Property between the "Measure" and "Unit" classes is presented. Likewise, in the SAREF ontology there is the relationship or Object Property between the "Measurement" and "Unit of Measure" classes. Both of these peculiarities –although not shown in Figure 10 for reasons of clarity– facilitate access to all the characteristics and properties of such ontologies, since they are available from the "Set Output Data Point" and "Get Input Data Point" subclasses of the oneM2M ontology, when associated with the "Quantity" subclass of the OM ontology.

<table>
<tr><td rowspan="2">OM: Quantity (Sub Class)</td><td>SAREF4 ENER: Energy (Super Class)</td><td rowspan="4">The electricity consumption of an electronic device connected to the computer network must be recorded in order to achieve its energy efficiency, by keeping track of its energy and/or power required to perform its task. This is achieved by complementing and strengthening the associations already detailed in the oneM2M ontology, with the following links:

- Through multiple inheritance, from the “Quantity” subclass of the OM ontology, to the “Energy” and “Power” super classes, which are part of the SAREF4ENER ontology.

- Through simple inheritance, from the “Energy” and “Power” subclasses belonging to the SAREF4ENER ontology, to the “Property” superclass, which concerns the SAREF ontology.</td></tr>
<tr><td>SAREF4 ENER: Power (Super Class)</td></tr>
<tr><td>SAREF4 ENER: Energy (Sub Class)</td><td rowspan="2">SAREF: Property (Super Class)</td></tr>
<tr><td>SAREF4 ENER: Power (Sub Class)</td></tr>
</table>

Table 8. Details of the mapping and association of classes between the ontologies oneM2M, SAREF4ENER, OM and SAREF, choosing oneM2M as the ontological interface (Own elaboration).

9. Simulation

Step 10 of the system simulation process [38] concerns experimentation, that is, running the simulation to generate the expected data or the result of the modeling and thus, performing a sensitivity study that facilitates understanding the uncertainties, scope, and limitations of the proposed architectural model, which is developed as follows:

The following is indicated in OWL (Ontology Web Language): the postulated flexible functional architectural model ontology, whose denomination is “OntologyModelingIoTCo2.owl”, being compatible with RFD/XML (Resource Framework Description and eXtensible Markup Language) and its web address is https://gitlab.com/mbermudez.amaya/ModelingIoTCo2/-/raw/main/OntologyModelingIoTCo2.owl.

The complete source code of the proposed ontology developed by the author of this research (OntologyModelingIoTCo2.owl) is displayed in the hyperlink mentioned above. A fraction of this can be seen below because of space limitations.

Example fragment for the concept “ControllingFunction”:

<Class rdf:about=”https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#ControllingFunction”>

<rdfs:subClassOf rdf:resource=”https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#LoadControlEventData”/>

<rdfs:subClassOf rdf:resource=”https://saref.etsi.org/core/v3.1.1/saref.rdf#Function”/>

<rdfs:subClassOf rdf:resource=”https://saref.etsi.org/saref4ener/v1.1.2/saref4ener.rdf#LoadControlStateData”/>…

</Class>

When performing the simulation in the W3C validator (http://www.w3.org/RDF/Validator) of the proposed ontology in OWL format, the result is the graph (https://gitlab.com/mbermudez.amaya/ModelingIoTCo2/-/blob/main/GraphOntologyModelingIoTCo2.png) and the RDF/XML triplets of the model (https://gitlab.com/mbermudez.amaya/ModelingIoTCo2/-/blob/main/TripletsOntologyModelingIoTCo2.pdf).

Owing to space reasons in this document, the graph and the triplets mentioned are not included; instead, the relevant links are provided

10. Analysis of results

Step 11 of the system simulation process [38], which consists of the analysis and interpretation of results, leads to the extraction of inferences from the data generated by the execution of the simulation or experimentation.

The application of said simulation in reference to the validation [42] of the proposed ontology called “OntologyModelingIoTCo2.owl”, was carried out by applying the World Wide Web Consortium W3C validator (http://www.w3.org/RDF/Validator), which yielded the results detailed below regarding the compilation of its instructions, its graphical representation and its semantic units:

1.) In reference to the source code file called “OntologyModelingIoTCo2” and with the OWL extension, the following was obtained:

- All instructions are exact and precise, both syntactically and grammatically.
- All namespaces are well defined.
- Every web resource associated with each namespace was verified to be available on a computer network.
- The proposed ontology correctly integrates other ontologies, namely, oneM2M, SAREF4ENER, OM, and SAREF.
- Within the proposed ontology, the subclasses belonging to oneM2M and which are at the lower level of its hierarchy were also chosen so that they themselves function as subclasses of the proposed ontology “OntologyModelingIoTCo2”, remaining without defects when compiled.
- The multiple inheritance set up between the subclasses of oneM2M and the superclasses of the ontologies SAREF4ENER, OM, and SAREF is correct.
- The definition of classes that function as domains within each of the various ontologies is correct.

2.) Regarding the graph or diagram of the proposed ontology “OntologyModelingIoTCo2.owl”, the following was achieved:

- All nodes match their respective classes and data in the source code.
- All edges match the inheritance and dependency associations in the source code.
- The Multiple inheritance associations observed in the graph are consistent with the source code file.

- Access to each web resource listed in the nodes and edges was checked, and all were found to be in compliance.
- Each node and edge of the graph was compared with its respective class, data, and association in the source code in OWL, without any defects being present.

3.) Regarding the RDF triplets (semantic units) of the proposed ontology "OntologyModelingIoTCo2.owl", the following was achieved:

- Seventy (70) RDF triplets were coded in total.
- Each triplet is consistent with the "Subject-Predicate-Object" data expression model of the RDF Resource Description Framework.
- No empty node (unknown Subject or Object) was presented; therefore, it was not necessary to replace it with a generated ID "genid", which is known as "skolemizing".
- Access to each web resource that appears in each triplet was verified, both at the Subject and Object levels, as well as at the predicate level, and all were in compliance.
- Each triplet in the graph was compared with its corresponding node (Subject, Object) and edge (Predicate), with no missing or surplus nodes.
- Each triplet was verified with its respective class (Subject, Object), association (Predicate), and data (Object), without any imperfections occurring.

It is therefore inferred that the analysis of results has yielded a favorable corollary in the simulation carried out on the proposed ontological model named "OntologyModelingIoTCo2.owl," which is corroborated in the class hierarchy diagram in Figure 10 corresponding to the translation model (Item 6).

11. Conclusions

Step 12 of the system simulation process [38] involves implementation and documentation, that is, reporting the results, putting them into operation, recording their findings, and writing up the work done with the model and its use. These aspects are as follows.

The IoT flexible functional architectural model initiative applies in the first instance to smart appliances, but not to intelligent devices. Given that in Spanish-speaking countries, the words "smart" and "intelligent" are used interchangeably, this does not occur in the English language [43], establishing the difference as follows: the word "smart" refers to the device programmed to be capable of independent action; likewise, the term "intelligent" refers to the device capable of changing its state or its action in response to variable situations or past experiences in a rational manner –since the latter does not suggest understanding [44], as intelligence does–. In addition, the number of devices that respond to a given instruction (capture a signal/ move a mechanism, i.e., the "smart appliance") is much larger than those that make a decision regarding an event (regulate the behavior of the device according to the value of the measured magnitude, id est, the "intelligent device"). Therefore, for the purpose of this research, a smart appliance is defined as "a tangible object designed to perform a particular task in ordinary homes, offices, or public buildings [10] [11] [12] [13]. The use of the model developed for "intelligent device" devices will be part of a subsequent study.

Access to all the features and properties of the proposed ontology's class hierarchy, starting from the classes that are at the lower level of the same (i.e., the oneM2M subclasses), guarantees an efficient implementation of multiple inheritance; that is, an adequate and convenient use is made of aggregation- and composition-type associations. Therefore, the OWL construction for the architectural model supported by the proposed ontology allows overcoming dilemmas typical of this type of inheritance –for example, the diamond problem– starting from the use of namespaces (i.e., the name of each ontology), the application of intersections or unions (known as interfaces in OOP), and efficient data mapping.

The proposed ontology "OntologyModelingIoTCo2.owl" is consistent with the foundation, composition, conceptualization, and functionality of oneM2M technology, as well as a common standard platform for M2M/IoT, which allows the interworking of devices, whether smart or not. It has a set of technical specifications in the form of reports regarding architecture, requirements, domain models, protocols, abstraction, semantics, security, and tests. It also provides a middleware solution that links both network and application resources, and provides application programming interface APIs (functions, routines, methods, etc.) in a service layer [45], facilitating the connection between devices so that they interact, regardless of their underlying technology. This is why the postulated ontology, like that of oneM2M, is suitable to act as a point interface for all other ontologies with which it is combined, because of its universality, architecture, and syntactical and semantic interoperability to connect both oneM2M and non-oneM2M technology devices. This holistic vision of oneM2M as an ontological interface allows focusing on the artifact or machine, the data or values, and the function or task to be performed with them. It is not in vain that, at the lowest level of the respective class hierarchy of the modeled semantic corpus, there are the subclasses "Device", "GET_OutputDataPoint", "SET_InputDataPoint", "ControllingFunction", and "MeasuringFunction".

The proposed ontological model is an integration or combination of already consolidated ontologies; it does not intend to become a new semantic corpus of a certain universe of discourse, in addition to those that already exist. It only seeks to become an alternative resource for the management of CO2 emissions in cities, supported by both the Internet of Things and M2M technology, and thus, to optimize the consumption of electrical energy by digital devices (i.e., electronics). Therefore, "OntologyModelingIoTCo2" ontology postulates that oneM2M should not be implemented as a supra-ontology, nor should it be attempted to map the largest number of classes and properties. The homologation of conceptual frameworks is not intended; On the contrary, the aim is to integrate them and for oneM2M to function as an infra-ontology so that, based on the identification of the classes that are at the lower level of each ontology, it is possible to access all the characteristics and behaviors of the same combined, given that there is a tendency in the world of computing to implement oneM2M as a metamodel; that is, as a profile, configuration or theoretical scheme of a specific domain and from which other models are derived or, failing that, a textual and graphic conceptual framework that brings together various existing models, already tested and in use.

Given the versatility, robustness, suitability and support that distinguish the oneM2M, SAREF4ENER, OM and SAREF ontologies, these were more than enough reasons for their selection in the conformation of the proposed ontology for urban CO2 management through IoT "OntologyModelingIoTCo2". However, such adaptability must be handled with caution, given that the ontologies that have been chosen and integrated are designed with so much flexibility, which the late event of loss of clarity and conciseness could manifest itself, with the possibility of gradually leaving aside accuracy and precision. Therefore, their rigor should be prevented from dissipating, as their developers add object properties to them so that their usefulness is holistic. This circumstance could –subsequently–, harm them in terms of their understanding and, therefore, their application to overcome the problems of open and standardized communication. Therefore, it would be advisable to regularize such a contingency, since there would run the risk of making them impracticality.

The implementation of a Customer Energy Manager (CEM) as a service facilitates the control or administration of the energy efficiency of a device and, therefore, the management of CO2 emissions in cities by recording the consumption/production of energy/power on the fly. Which, in association with the inherent syntactic and semantic interoperability offered by the oneM2M technology (integrated in the proposed ontology) and on the way to optimization, will contribute to the mitigation of global warming. In other words, if the energy spending of smart devices supported by IoT is controlled, urban atmospheric carbon emissions are managed by default.

The validation of the proposed ontology in OWL format "OntologyModelingIoTCo2" showed that the semantic units or triplets are consistent in terms of Subject – Predicate – Object, denoting that the Subject and Object entities were correctly defined with regard to the web resources they link, whether they were classes or data. Likewise, regarding the Predicate entity, they were also determined with accuracy and precision with regard to the Object Property or relationship they are linking. Likewise, the graph of the proposed ontological model generated by the validation stands out for its schematic coherence with regard to the source nodes (Subject, Class), edges (Predicate, Property), and destination nodes (Object, Class or Data), which allow it to be verified through triplets. Finally, it is worth noting that there are no empty nodes in the graph of the proposed ontology when validating the source code in the corresponding tool; therefore, it is not necessary to replace these nodes with an IRI that has an ID generated –genid– by the validator. In other words, no "skolemization" is necessary.

The results obtained from the experimentation confirm that the ontological model designed in steps 6 and 9 of the simulation process of a system [38] are appropriate and convenient for managing the emission of urban atmospheric carbon from the modeling of an architecture of the IoT, by recording and controlling the consumption/production of energy/power by an electronic device with connectivity to the computer network, that is, a smart appliance type device. Therefore, Figure 10 confirms the diagram of the proposed ontology "OntologyModelingIoTCo2," whose navigation is carried out in bottom-up horizons for each mapped ontology; that is, oneM2M, OM, SAREF4ENER, and SAREF.

12. Future works

In the installation and configuration of M2M servers to access the oneM2M architecture, opportunities for future work are presented, as machine-to-machine technology [46] consists of a system "that allows computers, embedded processors, smart sensors, actuators, and mobile devices to communicate with each other, take measurements and make decisions, often without human intervention", as well as being a configuration [47] to acquire data "from a large-scale physical domain and control it in return, with the sensors and actuators shared and distributed as monitoring and control points across said infrastructure." Therefore, with a Client-Server and Service-Oriented Architecture SOA [48], a machine-to-machine system is established in which the computer equipment in charge of processing the requests and responses of the devices (smart or not, oneM2M or Non-oneM2M) through an M2M gateway becomes the application server to manage the business logic and data access.

Another possible future work concerns the preparation of such a gateway, implementing a Single Board Computer (SBC), whose architecture has been developed by several organizations (Arduino, Raspberry, BeagleBoard, Gumstix, etc.). Given its economic cost, good performance in demanding environments, open-source operating system, reduced energy consumption, and versatility of implementation (education, industry, prototypes, etc.), it has become a favorite within the world of computing. Likewise, its configuration as an "M2M Gateway" is carried out through an open source container, accompanied by an object-oriented data input and output API to manage said platform, with low-level native integration and interruption supervision, which makes it easier to concentrate on the gateway's objective: to act as a physical and logical connection between the machine-to-machine server and the electronic device for its corresponding management (monitoring and/or actuation).
A subsequent activity would consist of the adaptation of the energy-efficiency domain, understood as the appropriate consumption of electrical power to perform tasks –singular or diverse– by the device (i.e., energy spending). This configuration corresponds to the specification or standardized application of the ontological model (OntologyModelingIoTCo2) that would facilitate the management of CO2 emissions in cities through the IoT, which integrates the ontologies oneM2M, SAREF4ENER, OM, and SAREF.

Recording the energy consumption of the smart device is another task that must be carried out in a typical (unrestricted) and efficient (restricted) manner and in various units, such as kilowatt hours (kWh), kilo-tonnes of oil equivalent (kToe), giga-tonnes of carbon dioxide equivalent (GtCO2e), etc., in coordination with the optimization of energy efficiency, by articulating the Client Energy Manager, the M2M server with the oneM2M service APIs, the gateway with both the proxy and oneM2M container APIs, and finally, the device (smart or not). All of this in the computer network, to manage the load control and the power series of said device when consuming/producing the energy/power, it requires to carry out its perception/activation work, thus resulting in the management of atmospheric CO2 in cities.

References

[1] Gruber, T.R. (1993). A translation approach to portable ontology specifications. In *Knowledge Acquisition (Incorporated into International Journal of Human-Computer Studies), Elsevier BV,* 5(2), 199-220.

[2] Liu, L. & Özsu, M.T. (2009). Encyclopedia of Database Systems. In *Springer-Verlag US, New York, NY. Ling Liu and M. Tamer* Özsu *(Eds.).* 1, 1–4355.

[3] Gruber, T.R. (1995). Toward principles for the design of ontologies used for knowledge sharing? In *International Journal of Human-Computer Studies. Elsevier BV, Amsterdam, Netherlands,* 43(5–6), 907–928.

[4] Rijgersberg, H., Wigham, M., & Top, J.L. (2011). How semantics can improve engineering processes: A case of units of measure and quantities. In *Advanced Engineering Informatics, the science of supporting knowledge-intensive activities. Elsevier BV, Amsterdam, Netherlands.* 25(2), 276-287.

[5] Rijgersberg, H., Assem, M.V., & Top, J.L. (2013). Ontology of units of measure and related concepts. In *Semantic Web – Interoperability, Usability, Applicability, Linked Data for Science and Education, IOS Press.* 4(1), 3–13.

[6] Antoniou, G., & Harmelen, F.V. (2009). Web Ontology Language: OWL. In *Handbook on Ontologies. International Handbooks on Information Systems (INFOSYS). Springer, Berlin, Heidelberg.* 91–110.

[7] Díez, G.F. (2006). El análisis lógico de la predicación colectiva. In *Crítica. Revista Hispanoamericana De Filosofía,* 38(114), 21–44.

[8] Hartog, F.D., Daniele, L., & Roes, J. (2015). Toward semantic interoperability of energy using and producing appliances in residential environments. In *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC). Las Vegas, NV, USA.* 170–175.

[9] Daniele, L., Hartog, F.D., & Roes, J. (2015). Created in Close Interaction with the Industry: The Smart Appliances REFerence (SAREF) Ontology. In *Formal Ontologies Meet Industry. 7th International Workshop, FOMI 2015. Springer, Cham.* 225, 100–112.

[10] Daniele, L., Hartog, F.D., & Roes, J. (2015). How to Keep a Reference Ontology Relevant to the Industry: A Case Study from the Smart Home. In *Ontology Engineering, 12th International Experiences and Directions Workshop on OWL, OWLED 2015. Springer, Cham.* 9557, 117–123.

[11] ETSI. (2015). SmartM2M; Smart Appliances; Reference Ontology and oneM2M Mapping. In *European Telecommunications Standards Institute, ETSI. TS 103 264 V1.1.1 (2015-11).* 1–26.

[12] Sahlmann, K., Scheffler, T., & Schnor, B. (2018). Ontology-driven Device Descriptions for IoT Network Management. In *IEEE 2018 Global Internet of Things Summit (GIoTS). Bilbao, Spain.* 1–6.

[13] Gyrard, A. (2015). Designing cross-domain semantic Web of things applications. In *Ubiquitous Computing. Télécom ParisTech. HAL open science.* 1–231.

[14] Saha, G.K. (2007). Web ontology language (OWL) and semantic web. In *Ubiquity. Association for Computing Machinery ACM, New York, NY, US.* 2007(September), 1–24.

[15] Berners-Lee, T., Fielding, R.T., & Masinter, L. (2005). Uniform Resource Identifier (URI): Generic Syntax. In *RFC 3986. STD 66. The Internet Engineering Task Force IETF, RFC Editor. January 2005.* 1–61.

[16] Freed, N., & Borenstein, N.S. (1996). Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. In *RFC 2046. The Internet Engineering Task Force IETF, RFC Editor. November 1996.* 1–44.

[17] Swartz, A. (2004). application/rdf+xml Media Type Registration. In *RFC 3870. The Internet Engineering Task Force IETF, RFC Editor. September 2004.* 1–8.

[18] Tomaszuk, D., & Hyland-Wood, D. (2020). RDF 1.1: Knowledge Representation and Data Integration Language for the Web. In *Symmetry. Multidisciplinary Digital Publishing Institute MDPI.* 12(1), 1–26.

[19] McBride, B. (2004). The Resource Description Framework (RDF) and its Vocabulary Description Language RDFS. In *Handbook on Ontologies. International Handbooks on Information Systems. Springer, Berlin, Heidelberg, January 2004.* 51–65.

[20] Brickley, D., & Guha, R.V. (2002). Resource Description Framework (RDF) Model and Syntax Specification. In *W3C Recommendation 22 February 1999. World Wide Web Consortium. O. Lassila & R.R. Swick (Eds.).* 1–47.

[21] Motik, B., et al. (2008). OWL 2 Web Ontology Language XML Serialization. In *W3C Recommendation 27 October 2009. World Wide Web Consortium. Boris Motik, et alii, (Eds.).* 1–35.

[22] Freed, N., & Klensin, J.C. (2005). Media Type Specifications and Registration Procedures. In *RFC 4288. The Internet Engineering Task Force IETF, RFC Editor. December 2005.* 1–24.

[23] Hitzler, P., et al. (2012). OWL 2 Web Ontology Language Primer (Second Edition). In *W3C Recommendation 11 December 2012. World Wide Web Consortium. Pascal Hitzler, et alii, (Eds.).* 1–65.

[24] Dürst, M.J., & Suignard, M. (2005). Internationalized Resource Identifiers (IRIs). In *RFC 3987. The Internet Engineering Task Force IETF, RFC Editor. January 2005.* 1–46.

[25] Haghgoo, M., Sychev, I., Monti, A., & Fitzek, F.H. (2020). SARGON – Smart energy domain ontology. In *IET Smart Cities. The Institution of Engineering and Technology,* 2(4), 191–198.

[26] Weerdt, R.V., Boer, V.D., Daniele, L., Nouwt, B., & Siebes, R. (2021). Making heterogeneous smart home data interoperable with the SAREF ontology. In *International Journal of Metadata, Semantics and Ontologies IJMSO,* 15(4), 280–293.

[27] Lefrançois, M. (2017). Planned ETSI SAREF Extensions based on the W3C&OGC SOSA/SSN-compatible SEAS Ontology Pattern. In *International Conference on Semantic Systems. Workshop on Semantic Interoperability and Standardization in the IoT, SIS-IoT'17, HAL open science,* 1–11.

[28] Daniele, L., Solanki, M., Hartog, F.D., & Roes, J. (2016). Interoperability for Smart Appliances in the IoT World. In *The Semantic Web – ISWC 2016. International Semantic Web Conference 2016. Lecture Notes in Computer Science. Springer, Cham,* 9982, 21–29.

[29] Gyrard, A., Kung, A., Genest, O., & Moreau, A. (2021). SAREF-Compliant Knowledge Discovery for Semantic Energy and Grid Interoperability. In *2021 IEEE 7th World Forum on Internet of Things (WF-IoT),* 71-76.

[30] Datta, S.K., Gyrard, A., Bonnet, C., & Boudaoud, K. (2015). oneM2M Architecture Based User Centric IoT Application Development. In *IEEE 2015 3rd International Conference on Future Internet of Things and Cloud (FiCloud),* 100-107.

[31] Gyrard, A., Datta, S.K., Bonnet, C., & Boudaoud, K. (2015). A Semantic Engine for Internet of Things: Cloud, Mobile Devices and Gateways. In *IEEE 2015 9th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS),* 336-341.

[32] Grønbæk, I. (2008). Architecture for the Internet of Things (IoT): API and Interconnect. In *IEEE 2008 Second International Conference on Sensor Technologies and Applications (sensorcomm 2008), Cap Esterel, France,* 802–807.

[33] Sun, J., et al. (2019). Ontology-Based Interworking Framework Among Heterogeneous IoT Protocols. In *Transdisciplinary Engineering for Complex Socio-technical Systems. Advances in Transdisciplinary Engineering. IOS Press,* 10, 360–369.

[34] Kang, S., & Chung, K. (2018). IoT Framework for Interworking and Autonomous Interaction Between Heterogeneous IoT Platforms. In *Smart Computing and Communication. SmartCom 2018. Lecture Notes in Computer Science. Springer, Cham,* 11344, 217–225.

[35] Ganzha, M., Paprzycki, M., Pawłowski, W., Szmeja, P., & Wasielewska, K. (2017). Towards Common Vocabulary for IoT Ecosystems - preliminary Considerations. In *Intelligent Information and Database Systems. ACIIDS 2017. Lecture Notes in Computer Science. Springer, Cham,* 10191, 35–45.

[36] Barchini, G.E., Alvarez, M.M., & Herrera, S. (2006). Sistemas de información: nuevos escenarios basados en ontologías. In *Jistem Journal of Information Systems and Technology Management,* 3(1), 2–18.

[37] Noll, R.P., & Ribeiro, M.B. (2007). Ontological Traceability over the Unified Process. In *14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'07), Tucson, AZ, USA,* 249–255.

[38] Shannon, R.E. (1998). Introduction to the art and science of simulation. In *WSC '98: Proceedings of the 30th conference on Winter Simulation. IEEE, Washington, DC, USA. December 1998.* 1, 7–14.

[39] Silva, F.A., Sifaqui, C., & Labra Gayo, J.E. (2011). Towards an architecture and adoption process for linked data technologies in open government contexts: a case study for the Library of Congress of Chile. International Conference on Semantic Systems. In *I-Semantics '11: Proceedings of the 7th International Conference on Semantic Systems. Association for Computing Machinery ACM,* 148–161.

[40] Cuenca, J., Larrinaga, F., & Curry, E. (2020). DABGEO: A Reusable and Usable Global Energy Ontology for the Energy Domain. In J*ournal of Web Semantics: Science, Services and Agents on the World Wide Web, March 2020.* Vol. 61–62, 100550. 1–12.

[41] Weerdt, R.V., Boer, V.D., Daniele, L., & Nouwt, B. (2021). Validating SAREF in a Smart Home Environment. In *Metadata and Semantic Research. MTSR 2020. Communications in Computer and Information Science. Springer, Cham,* 1355, 35–46.

[42] Labra Gayo, J.E., Prud'hommeaux, E., Boneva, I., & Kontokostas, D. (2018). Validating RDF Data. In *Validating RDF Data. Synthesis Lectures on Data, Semantics, and Knowledge. Springer, Cham,* 9–26.

[43] Taiwo, O., & Absalom, E.E. (2021). Internet of Things-Based Intelligent Smart Home Control System. In *Security and Communication Networks, Hindawi Ltd., John Wiley & Sons Inc.,* 2021, 1–17.

[44] Russell, S.J. (1997). Rationality and Intelligence. In *Artificial Intelligence. Elsevier BV, Amsterdam, Netherlands,* 94(1–2), 57–77.

[45] Bermúdez Amaya, M.O., Salcedo Parra, O.J., & Rodríguez Miranda, J.P. (2021). Ontological base models machine-to-machine M2M applied to the internet of things IOT. In *Revista Boletín Redipe, diciembre 2021,* 10(12), 148–161.

[46] Watson, D., Piette, M.A., Sezgen, O., Motegi, N., & Hope, L.T. (2004). Machine to Machine (M2M) Technology in Demand Responsive Commercial Buildings. In *Proceedings 2004 ACEEE Summer Study on Energy Efficiency in Buildings,* 3, 351–363.

[47] Grieco, L.A., Alaya, M.B., Monteil, T., & Drira, K. (2014). Architecting information centric ETSI-M2M systems. In *2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS),* 211–214.

[48] Bermúdez Amaya, M.O., Salcedo Parra, O.J., & Acosta López, A. (2020). Functional architectures for the Internet of Things: fundamental models. In *International Journal of Mechanical and Production Engineering Research and Development IJMPERD. TransStellar Journal Publications Research Consultancy TJPRC (Ed.)*. 10(6), 655–672.