

EEPSA as a core ontology for energy efficiency and thermal comfort in buildings

Iker Esnaola-Gonzalez [a,b], Jesús Bermúdez [b], Izaskun Fernandez [a], and Aitor Arnaiz [a]
[a] *IK4-TEKNIKER, Iñaki Goenaga 5, 20600 Eibar, Spain*
E-mail:{iker.esnaola, izaskun.fernandez, aitor.arnaiz}@tekniker.es
[b] *University of the Basque Country (UPV/EHU), Paseo Manuel Lardizabal 1, 20018 Donostia-San Sebastián, Spain*
E-mail: jesus.bermudez@ehu.eus

Abstract. Achieving a comfortable thermal situation within buildings with an efficient use of energy remains still an open challenge for most buildings. In this regard, IoT (Internet of Things) and KDD (Knowledge Discovery in Databases) processes may be combined to solve these problems, even though data analysts may feel overwhelmed by heterogeneity and volume of the data to be considered. Data analysts could benefit from an application assistant that supports them throughout the KDD process and aids them discovering which are the most relevant variables for the matter at hand, or informing about relationships among relevant data. In this article, the EEPSA (Energy Efficiency Prediction Semantic Assistant) ontology which supports such an assistant is presented. This ontology is developed on top of three ODPs (Ontology Design Patterns) which address weaknesses of existing proposals to represent features of interest and their respective qualities, as well as observations and actuations, the sensors and actuators that generate them, and the procedures used. The ontology is designed so that its customization to address similar problems in different types of buildings can be approached methodically. This feature is proved in a real-world poultry farm.

Keywords: Ontology, Ontology Design Patterns, Buildings, Energy Efficiency, Thermal Comfort

1. Introduction

In the early 2000s it was estimated that people used to spend around 90% of their time indoors [1], and this is a situation that may still apply nowadays. Thence, feeling comfortable while staying indoors is a must. User comfort can be influenced by different aspects such as visual, acoustic or thermal conditions. According to the ANSI/ASHRAE Standard 55-2017, thermal comfort is defined as follows: “that condition of mind that expresses satisfaction with the thermal environment and is assessed by subjective evaluation". Being a subjective perception, under the same thermal conditions different people can experience different levels of comfort.

Even though thermal comfort may be overlooked, conducted research proves its impact on humans. Some studies show the relation between indoor environment conditions and working efficiency or productivity [2,3]. There is also work demonstrating that indoor environment conditions can have a significant impact on occupants comfort, morale, health and well-being in commercial office buildings [4]. It is also proved that having an uncomfortable thermal situation involves many risks including clinical diseases, health impairments, and reduced human performance and work capacity [5]. Therefore, all these evidences reinforce the need of ensuring comfortable thermal conditions in buildings.

However, occupants' thermal comfort is not the only concern related to buildings. The building sector consumes more than 35% of global energy and it is responsible for nearly 40% of energy-related CO_2 emissions in the EU [6]. This is why, with views to meet the energy sustainability and minimize the climate change, this sector is addressed by the set of binding legislations agreed by the European Commission in the EU 2020 climate and energy package[1]. Therefore, the efficient management of building energy is becoming the trend for the future generation of buildings.

Fulfilling occupants' comfort whilst reducing energy consumption is still an unsolved problem in most buildings. Furthermore, it is important to note that certain type of buildings have specific features which may further hinder this problem. For instance, tertiary buildings normally contain spaces with big dimensions which are prone to have bigger thermal inertia [7]. This results in the need of longer periods of time to heat up or cool down spaces and consequently, they cannot be effectively climatized with rather simple solutions like thermostat-based reactive systems. Instead, heating or cooling systems need to be activated in advance in a specific mode to ensure a comfortable thermal condition in a given time. The expansion of the Internet of Things (IoT) [8] and Knowledge Discovery in Databases (KDD) [9] techniques may allow to improve matters in this regard. On the one hand, the IoT facilitates the monitoring of real-world qualities and events thanks to physical things equipped with electronic components and ubiquitous intelligence that allow them to connect, interact and exchange data. This led to the massive amount of data available nowadays, which has the potential to enable new discoveries and improve decision making processes. However, this data tends to be diverse and heterogeneous. Devices from different vendors may represent data in different formats, and even when a common format is used, the internal data model schema typically varies. Moreover, data may come from disparate external sources (often referred to as exogenous data), which further aggravates the data heterogeneity situation. Furthermore, the great variety of technical data hinders the human comprehension with regards to assessing which data is relevant for the matter at hand. These circumstances definitely pose a challenge for data analysts in charge of a KDD process which enables the extraction of useful knowledge from raw data.

In this context, data analysts have to deal with data related to the building where energy efficiency and user thermal comfort is aimed at. This data may describe building structural element properties including materials, heat transfer coefficients, and orientation of their boundaries (e.g. a room located in the second floor of a building which has a skilight with 2 m^2 of surface. A door with a U-factor of 2.61 that is opened by swinging to the left, and connects the hall with the southern outside part of the building). Data analysts also need to take into account information about sensors and actuators deployed in the building, their location, features and certainly their measurements and actuations (e.g. a temperature sensor located in the meeting room 03 that measured 23°C on 12th May 2018 at 16:35. A blind actuator that lowered blinds of window 121 on 26th November 2018 at 20:00). Likewise, data about weather conditions and weather forecasts for the building location are relevant (e.g. a forecast for Madrid made by the Spanish meteorology agency on 10th June 2018 at 10:00 forecasting a relative humidity of 53% on 12th June 2018 at 15:00. A weather report that described cloudy skies during the morning of 6th December in Amsterdam). Furthermore, there is other information to consider such as the space occupancy, work schedule or human related organization (e.g. the 29th November 2018 is a reduced working hours day. The occupancy value of the meeting room 06 at 11:00 is of 8 people). Under such circumstances where a deep energy efficiency, thermal comfort and building domain knowledge is required to efficiently handle all this information, having insufficient domain expertise could make data analysts feel overwhelmed. Consequently, they typically resort to a trial and error approach searching for variables and tasks that could be confidently used to make accurate predictions. Moreover, due to the plethora of possible combination of algorithms in each KDD phase, even expert data analysts may turn to the trial and error approach [10]. This is definitely an undesirable approach and it would be much more profitable to count with a KDD process assistant supported by technologies that enable the management of data semantics, data interrelationships, and knowledge representation. This means that it is necessary the representation of features of interest and their respective qualities, as well as observations and actuations, the sensors and actuators that generate them, and the procedures used. Furthermore, observations and actuations have to be described with respect to their values, in addition to their spatial and temporal context. In this paper, the development of a core ontology that supports such a KDD assistant is described.

The rest of this article is structured as follows. Section 2 makes a thorough review of ontologies related to the domains of discourse. Section 3 presents the development of the proposed core ontology. Section 4

[1] https://ec.europa.eu/clima/policies/strategies/2020_en

shows the customization of the ontology to apply it in a real-world use case. Finally the conclusions of this work are shown in section 5.

2. Related Work

Linking or mapping raw data to existing ontologies or vocabularies, allows a better representation of the data, structuring it and setting formal types, relations, properties and restrictions that hold among them. In addition, it allows representing data coming from multiple sources in a unified way, thereby supporting data integration [11]. Another benefit of the semantic annotation lies in the additional background knowledge about a domain that can be added to the dataset. This leads to the enrichment of the dataset, as well as enabling the application of indexing techniques, which are based on resource URIs and ensure the retrieval and navigation through related resources [12]. Last but not least, after a semantic annotation process, data is more domain-oriented than the original source and allows more application-independent solutions. Consequently, there is no need for the user to be aware of raw data's underlying structure.

Due to the aforementioned benefits, annotating data semantically can contribute improving the KDD process [13,14,15]. Next, the most relevant ontologies covering domains of discourse of this article are reviewed. Specifically, building domain ontologies are reviewed in section 2.1, ontologies addressing observations, actuations and related domains in section 2.2, and ontologies representing the context of these observations' and actuations' results in section 2.3. A discussion of reviewed ontologies is presented in section 2.4.

2.1. Building domain ontologies

BIM (Building Information Model) is a data exchange format used by different stakeholders involved in the construction process of a building, and deals with the representation of functional and physical characteristics of a building [16]. Each of these stakeholders adds domain knowledge to a common model which keeps information of the whole building life cycle. As a consequence, the model serves as a valuable source of information.

On the one hand, a BIM model may contain static information of a building element. For instance, in the case of a window, data about its location, the material it is made of, and even when it was installed is available and queryable. Nevertheless, it is not possible to know whether the window is opened or closed in a given moment. On the other hand, the use of IFC (Industry Foundation Classes) files has arisen several issues due to its complexity. A first approach for answering these issues comes in the form of processes such as IDMs (Information Delivery Manual) and the related MVDs (Model View Definition), which still remain complex and time-consuming [17]. Easy and intuitive ways to rapidly browse, query and use BIM information are not often available.

Semantic Technologies can be leveraged to remedy these issues, as they enable the data integration across several data sources and allow a more dynamic manipulation of the building information in RDF graphs via query and rule languages [18]. Furthermore, the ontology modelling paradigm for providing and implementing a BIM of a target building supports a variety of advantages such as reusability and automated reasoning upon the modelled entities. There exists a variety of technologies that offer conceptual modelling capabilities to describe a domain of interest, but only ontologies combine this feature with Web compliance, formality and reasoning capabilities [19].

There are many building domain ontologies, each designed to fulfil the specific information requirements of a certain use case within the AEC (Architecture, Engineering, and Construction) and FM (Facilities Management) domains. However, the lack of a common building model for representing data prevents interoperability between buildings and limits the scalability of applications. In this section, a set of the most relevant ontologies for modelling buildings are reviewed.

2.1.1. ifcOWL Ontology

The ifcOWL ontology[2] [20] provides an OWL representation of the EXPRESS schemas of IFC, which is the open standard developed by buildingSMART[3] for representing building and construction data. Using the ifcOWL ontology, IFC-based building models can be represented as directed labelled graphs. Furthermore, resulting RDF graphs can be linked to related data including material data, GIS (Geographic Information Systems) data or product manufacturer data.

The ifcOWL ontology aims at supporting the conversion of IFC instance files into equivalent RDF files.

[2] http://ifcowl.openbimstandards.org/IFC4_ADD2.owl
[3] https://www.buildingsmart.org/

This means that it is of secondary importance that an instance RDF file can be modelled from scratch using the ifcOWL ontology and an ontology editor. Furthermore, ifcOWL defines a faithful mapping of the IFC EXPRESS schema, replicating its conceptualization which has been found inconvenient for some practical engineering use cases [18]. For instance, the ifcOWL conceptualization of some relationships and properties as instances of classes (i.e. *ifc:IfcRelationship*, *ifc:IfcProperty*) is counterintuitive to semantic web principles, that would expect OWL properties to represent them. In this regard, a systematic transformation of this modelling issue has been proposed in the IfcWoD (IFC Web of Data) ontology[4] [21], which claims to simplify query writing, optimize execution of queries and maximize inference capabilities. Furthermore, other initiatives focus on addressing ifcOWL ontology weaknesses such as making IFC-based exchanged data more semantically robust [22] or making the ontology more flexible in terms its capability to deal with the real-world scenarios [23].

In summary, the ifcOWL ontology is a necessary tool to incorporate IFC models to the semantic web infrastructure but resulting graphs will be at least as large and complex as the original IFC models. This derives in models that may be too complicated and even inconvenient for some scenarios.

2.1.2. SAREF4BLDG

SAREF4BLDG[5] [24] is an extension of the SAREF ontology (explained in Section 2.2.3) based on the IFC standard. Since this extension is limited to devices and appliances, unlike in ifcOWL where the whole IFC is translated, only the corresponding part of the standard is transformed. In fact, it includes definitions from the IFC version 4-Addendum 1 to enable the representation of such devices and other physical objects in building spaces.

According to its representation, a building may have different spaces which may also have other sub spaces within themselves. These classes alongside with the class representing physical objects, are declared as subclasses of *geo:SpatialThing* in order to reuse the conceptualization for locations already proposed by the Basic Geo vocabulary (also known as WGS84 Geo Position vocabulary).

[4]At the moment of writing this article, the ontology is not publicly available.

[5]`https://w3id.org/def/saref4bldg`

2.1.3. DogOnt

The DogOnt ontology[6] [25] formalizes IDE (Intelligent Domotic Environment) aspects and it is designed with a particular focus on interoperation between domotic systems. Although primarily models devices, states and functionalities, it also supports the description of residential environments where devices are located.

Environment modelling in DogOnt is rather abstract and mainly aimed at locating indoor devices at room granularity. Reflecting this general design goal the available concepts permit to represent: (a) buildings, (b) storeys, as part of multi-storey buildings, (c) flats, either located on single or multiple storeys, (d) rooms inside flats and other indoor locations located outside flats (e.g. garages), (e) walls, ceilings, floors, partitions, doors and windows composing both rooms and building boundaries, and (f) objects contained in an indoor environment including furniture (e.g. chairs and desks) [26].

DogOnt authors claim that its ontology influenced the design principles of EEOnt, ThinkHome, and SAREF ontologies among others and that such common origin enables DogOnt to be used as a foundation towards a shared and unified schema for AEC/FM ontologies interoperability. The latest DogOnt version available at the moment of writing this article (version 4.0.1), counts with over 1,000 classes and over 70 properties, which may be rather large in some cases.

2.1.4. EEOnt

The Energy Efficiency Ontology [27] (EEOnt) is an ontology that defines the general structure of a building, the distribution and connectivity of its systems, objects, and spaces. Furthermore, the functionality and characteristics of the energy consuming devices and systems are also represented. The EEOnt describes EEI_B (Energy Efficiency Index for Buildings) and EEL_B (Energy Efficiency Landscape) corresponding to the building and its components, supplying useful information for the diagnosis and the correction of inefficiencies.

The principles of EEOnt are founded on DogOnt and its Energy Profile ontology (PowerOnt[7] [28]). Therefore, the modelling of the building environment, space and object topology is very similar to DogOnt,

[6]`http://elite.polito.it/ontologies/dogont.owl`

[7]`http://elite.polito.it/ontologies/poweront.owl`

as well as its abstraction level and focus on residential buildings. One of EEOnt's remarkable additions in this regard is that in the case of windows and doors, it includes two subclasses representing those that open to the outdoor and those that connect two inner spaces. Furthermore, fabrication materials (e.g. wood and steel) and the physical properties of those materials are specified following the IFC model.

It is worth noting that at the moment of writing this article, EEOnt is not publicly available.

2.1.5. *ThinkHome Ontology*

The ThinkHome ontology[8] [29] formalizes all relevant concepts needed to realize energy analysis in residential buildings. The knowledge captured in the ontology spans different domains, and it is logically segmented in different modules such as WeatherOntology[9] and EnergyResourceOntology[10].

The building information module (BuildingOntology[11]) describes knowledge that supports optimized control strategies striving for energy-efficient operation of smart homes. It consists of a set of basic classes, properties and customized datatypes that have been generated through XSLTs (Extensible Stylesheet Language Transformation) from gbXML (Green Building XML) Schema version 5.10. It focuses on the exchange of information for energy simulation and calculation, and therefore stores facts that are helpful for ThinkHome system's focal point.

There are enough concepts to model whole buildings including wall layers, window sizes and types, door sizes and positions, room areas and volumes as well as room purposes and orientation of buildings.

2.1.6. *BOT*

The Building Topology Ontology[12] [30] (BOT) is a minimal OWL DL ontology for covering core concepts of a building and for defining relationships between their subcomponents. A first design principle for the design of BOT has been to keep a light schema that could promote its reuse as a central ontology in the AEC domain.

BOT describes sites comprising buildings, composed of storeys which have spaces that can contain and be bounded by building elements. Sites, buildings, storeys and spaces are all non-physical objects defining a spatial zone [31]. These basic concepts and properties make the schema no more complex than necessary and this design makes the ontology a baseline extensible with concepts and properties from more domain specific ontologies. Therefore, BOT serves as an ontology to be shared.

Moreover, the W3C LBD community group[13] is aimed at producing more ontologies addressing geometry, products and other requirements across the life cycle of buildings that will extend from BOT concepts. The Building Product Ontology (PRODUCT[14]) is aimed at describing building elements (e.g. doors and windows), furnishings (e.g. chairs and tables), and MEP (Mechanical, Electrical and Plumbing) elements (e.g. humidifiers and energy meters) by means of different ontology modules. Furthermore, the iterative nature of a building design entails that information which is valid at one point in time might no longer be valid in the future. In order to manage that value variability and to keep track of property evolution history, the OPM (Ontology for Property Management) ontology[15] [32] is proposed. Finally, the emergence of a need for a standardized approach towards building-related properties derives in the future creation of the PROPS ontology[16].

It is worth mentioning that, BOT is aligned with related domain ontologies such as ifcOWL, DogOnt and Brick [33].

2.1.7. *FIEMSER Ontology*

The FIEMSER ontology[17] describes an energy-focused BIM model and WSN (Wireless Sensor Network) related data for residential buildings. With regards to the building-related concepts, it takes into account other building-related approaches such as IFC. The ontology describes buildings which consist of some building spaces representing flats or common areas. Likewise, these spaces consist of some other physical spaces. Furthermore, a building zone defines

[8] `https://www.auto.tuwien.ac.at/downloads/thinkhome/ontology/`
[9] `https://www.auto.tuwien.ac.at/downloads/thinkhome/ontology/WeatherOntology.owl`
[10] `https://www.auto.tuwien.ac.at/downloads/thinkhome/ontology/EnergyResourceOntology.owl`
[11] `https://www.auto.tuwien.ac.at/downloads/thinkhome/ontology/BuildingOntology.owl`
[12] `https://w3id.org/bot`
[13] `https://www.w3.org/community/lbd/`
[14] `https://github.com/w3c-lbd-cg/product`
[15] `https://github.com/w3c-lbd-cg/opm/blob/master/opm.ttl`
[16] `https://github.com/w3c-lbd-cg/props`
[17] `https://sites.google.com/site/smartappliancesproject/ontologies/fiemser-ontology`

a functional area in the building that will be controlled as a unique zone and which can be an aggregation of one or more building spaces. The source used to create the FIEMSER ontology is a secured pdf from which the information could not be automatically copied. As a consequence, comments that could better explain the ontology may be missing.

The FIEMSER data model represents one of the main trends identified in the context of the Smart Appliances study of the SAREF ontology and it is therefore linked to it.

2.1.8. Brick Ontology

Brick[18] [34] is a uniform schema for representing metadata in buildings and defines a concrete ontology for sensors, their subsystems and relationships among them. While other ontologies focus on BIM which is more oriented towards design and construction efforts, Brick has a specific emphasis on BMS (Building Management Systems) focused on building operation. The ontology captures hierarchies, relationships and properties for describing building metadata and has a clear focus on commercial buildings.

The design of Brick follows a methodology that combines tagging (like in the Project Haystack[19]) and semantic models. The resulting terminology allows describing real buildings but at the cost of a counterintuitive hierarchy of classes and a biased set of properties. Moreover, explanatory annotations accompanying term definitions are very scarce.

2.2. Observations, Actuations and other related domain ontologies

The rapid adoption of the IoT leads to an exponential growth of the number of existing devices worlwide. The IoT technology allows connecting the physical world with virtual representations in various domains including transportation, health and manufacturing. One of the most highlighted drawbacks of the IoT lies in the data level heterogeneity originated from different data models and formats supported by various device manufacturers. Such a diversity derives in semantic interoperability problems, where each system can represent the same thing in different ways, hindering the integration and understanding between these systems. In fact, a study estimated that nearly US$80 billion per year could be yield by implementing an effective semantic interoperability standard in the healthcare domain [35].

It has been proved that ontology-based approaches could contribute in achieving semantic interoperability [36], for instance by linking each data element to ontology terms thus providing them with semantics [37,38,39]. Furthermore, these approaches enable the discovery of IoT services, data and resources [40]. However, defining a comprehensive unified ontology for the domain of IoT may be challenging as there are more than 200 ontologies available [41].

There are some concepts that are common to the majority of IoT platforms [42], such as sensors, actuators and their corresponding observations[20] and actuations. In fact, these concepts comprise an important area of discourse of the problem tackled in this article. Next, a set of relevant ontologies covering these concepts are reviewed.

2.2.1. SSN Ontology

The initial Semantic Sensor Network (SSN) ontology[21] [43] was developed by the W3C Semantic Sensor Networks Incubator Group (SSN-XG) and it propsoed a conceptual schema for describing sensors, accuracy and capabilities of such sensors, their observations and methods used for sensing. Concepts for operating and survival ranges were also included, as well as sensors' performance within those ranges. Finally, a structure for field deployment was defined to describe deployment lifetime and sensing purposes. The initial SSN ontology was aligned with DOLCE ultralite (DUL) ontology and built on top of the Stimulus-Sensor-Observation (SSO) [44] Ontology Design Pattern (ODP) describing the relationships between sensors, stimulus, and observations.

The W3C Spatial Data on the Web Working Group (SDWWG[22]) proposed an update of the SSN ontology [45] (from now on referred to as SOSA/SSN ontology) that became a W3C recommendation. This new ontology follows a horizontal and vertical modularization architecture by including a lightweight but self-contained core ontology called SOSA[23] (Sensor, Ob-

[18]https://brickschema.org/

[19]http://project-haystack.org/

[20]The observation term is already used in different ways in different communities. The O&M (Observations and Measurements) model described in ISO 19156:2011 resolved this issue describing an observation as an event or activity, the result of which is an estimate of the value of a property of the feature of interest, obtained using a specific procedure.

[21]http://www.w3.org/ns/ssn/

[22]https://www.w3.org/2015/spatial

[23]http://www.w3.org/ns/sosa/

servation, Sample, and Actuator) for its elementary classes and properties. Furthermore, the SOSA/SSN ontology's scope is not limited to observations, but it is extended to cover actuations and samplings. In line with the changes implemented in the SOSA/SSN ontology, SOSA drops the direct DUL alignment although it can still be optionally achieved via the SSN-DUL alignment module[24]. Moreover, similar to the original SSO pattern, SOSA acts as a central building block for the new SOSA/SSN ontology but puts more emphasis on its lightweight expressivity and the ability to be used standalone. Then, constraint axioms are added to the vertical module extension named SSN.

Neither the previous SSN ontology nor the new SOSA/SSN ontology describe the different qualities which can be measured by sensors or acted on by actuators. Neither are covered related concepts such as units of measurements of these qualities, hierarchies of sensor/actuator/sampler types, or spatio-temporal terms. All this knowledge has to be modelled by the user, or preferably imported from other existing vocabularies.

2.2.2. om-lite Ontology

The om-lite ontology[25] [46] is an OWL representation of the Observation Schema described in clauses 7 and 8 of ISO 19156:2011 Geographic Information - Observations and Measurements (O&M)[26]. O&M defines a conceptual schema for observations, and for features involved when observations are produced. This schema separates concerns with classes for the feature of interest, the procedure, the observed property, the result, and the act of observation itself. This allows places and times associated with each of them to be distinct. An observation is defined as an act that results in the estimation of the value of a feature property, and it involves the application of a specified procedure, such as a sensor, instrument, algorithm or process chain. Specializations of the observation class are classified by the result-type. This way, the class *oml:Observation* has subclasses such as *oml:CountObservation* for observations whose results are integer, *oml:Measurement* for scaled numbers and *oml:TruthObservation* for booleans.

The om-lite ontology allows combining data unambiguously and referring to observations made in-situ, remotely, or ex-situ with respect to the location. These observation details are also important for data discovery and for data quality estimation. Furthermore, the om-lite ontology removes dependencies with preexisting ontologies and frameworks, and can therefore be used with minimal ontologies commitment beyond the O&M conceptual model. Additionally, it provides stub classes for time, geometry and measure (scaled number), which are expected to be substituted at runtime by a suitable concrete representation of the concept. Finally, it is aligned with PROV-O, as well as some other domain ontologies (e.g. the previous version of the SSN ontology).

2.2.3. SAREF Ontology

The Smart Appliances REFerence (SAREF) ontology[27] [47] is a shared model of consensus that facilitates the matching of existing assets in the smart appliances domain. The ontology provides building blocks that allow the separation and recombination of different parts of the ontology depending on specific needs. The central concept of the ontology is the *saref:Device* class, which is modelled in terms of functions, associated commands, states and provided services. The ontology describes types of devices such as sensors and actuators, white goods, HVAC (Heating, Ventilation and Air Conditioning) systems, lighting and micro renewable home solutions. A device makes an observation (which in SAREF is represented as *saref:Measurement*) which represents the value and timestamp and it is associated with a property (*saref:Property*) and a unit of measurement (*saref:UnitOfMeasure*). The description of these concepts is focused on the residential sector.

The modular conception of the ontology allows the definition of any new device based on building blocks describing functions that devices perform. As previously stated, for the building-related concepts SAREF provides the link to the FIEMSER data model. Furthermore, SAREF can be specialized to refine the general semantics captured in the ontology and create new concepts. The only requirement is that any extension/specialization may comply with SAREF. There are three extensions of the ontology: SAREF4BLDG which presents an extension of SAREF for the building domain, SAREF4ENVI[28] for the environment domain, and SAREF4ENER[29] for the energy domain.

[24] https://www.w3.org/ns/ssn/dul
[25] http://def.seegrid.csiro.au/ontology/om/om-lite
[26] https://www.iso.org/standard/32574.html
[27] http://ontology.tno.nl/saref
[28] https://w3id.org/def/saref4envi
[29] https://w3id.org/saref4ener

Furthermore, at the moment of writing this article there are three new planned extensions: SAREF4CITY for smart cities, SAREF4INMA for industry and manufacturing, and SAREF4AGRI for the agricultural domain.

2.2.4. SEAS Ontology

The SEAS Ontology[30] [48] is an ontology designed as a set of simple core ODPs that can be instantiated for multiple engineering related verticals. It is planned to be consolidated with the SAREF ontology as part of ETSI's Special Task Force 556[31]. The SEAS ontology modules are developed based on the following three core modules: the SEAS Feature of Interest ontology[32] which defines features of interest (*seas:FeatureOfInterest*) and their qualities (*seas:Property*), the SEAS Evaluation ontology[33] describing evaluation of these qualities, and the SEAS System ontology[34] representing virtually isolated systems connected with other systems. The Procedure Execution (PEP) ontology[35], which is not strictly a SEAS ontology module but it is contained under the same SEAS project, defines procedure executors that implement procedure methods, and generate procedure execution activities. Furthermore, PEP defines an ODP as a generalization of SOSA's sensor-procedure-observation and actuator-procedure-actuation models.

On top of these core modules, several vertical SEAS ontology modules are defined, which are dependent of a specific domain. For instance, the SEAS Electric Power System ontology[36] defines (i) systems that consume, produce or store electricity, (ii) connections between electric systems, and (iii) connection points of electric systems, through which electricity flows.

The SEAS ontology offers a set of alignments to ontologies like SOSA/SSN and QUDT.

2.2.5. IoT-O Ontology

The IoT-O ontology[37] [49] is a IoT domain modular ontology describing connected devices and their relation with the environment. It is intended to model knowledge about IoT systems and to be extended with application specific knowledge. It has been designed in five separated modules to facilitate its reuse and/or extension:

1. A sensing module, based on the previous version of the SSN ontology.
2. An acting module, based on the SAN (Semantic Actuator Network) ontology[38].
3. A service module, based on MSM[39] (Minimal Service Model) and hRESTS ontology[40].
4. A lifecycle module[41], based on a lifecycle vocabulary (a lightweight vocabulary defining state machines) and an IoT-specific extension.
5. An energy module, based on PowerOnt[42].

Furthermore, to maximize extensibility and reusability, IoT-O imports DUL and aligns all its concepts and imported modules with it.

The Observation representation proposed by the IoT-O ontology follows the same SSO pattern as its sensing module is based on the previous version of the SSN ontology. The representation of actuators, follows SAN ontology's AAE (the Actuation-Actuator-Effect) pattern, which intends to model the relationship between an actuator and the effect it has on its environment through actuations.

2.2.6. FIESTA-IoT Ontology

The FIESTA-IoT Ontology[43] [50] aims at creating a lightweight ontology that achieves semantic interoperability among heterogeneous testbeds. The ontology is focused on the description of the underlying testbeds' resource descriptions and the observations gathered from their physical devices. Furthermore, the design of the ontology is guided by the methodologies of ontology reuse and mapping. Some of the reused ontologies and taxonomies are the previous version of the SSN ontology, M3-lite taxonomy (a lite version of M3 ontology), Basic Geo WGS84 vocabulary, IoT-Lite ontology, OWL-Time ontology, and DUL ontology.

The previous version of the SSN ontology has a strong influence in FIESTA-IoT when describ-

[30] `https://w3id.org/seas/`
[31] `https://portal.etsi.org/STF/STFs/STFHomePages/STF556`
[32] `https://w3id.org/seas/FeatureOfInterestOntology`
[33] `https://w3id.org/seas/EvaluationOntology`
[34] `https://w3id.org/seas/SystemOntology`
[35] `https://w3id.org/pep/`
[36] `https://w3id.org/seas/ElectricPowerSystemOntology`
[37] `https://www.irit.fr/recherches/MELODI/ontologies/IoT-O`
[38] `https://www.irit.fr/recherches/MELODI/ontologies/SAN`
[39] `http://iserve.kmi.open.ac.uk/ns/msm`
[40] `http://www.wsmo.org/ns/hrests/`
[41] `https://www.irit.fr/recherches/MELODI/ontologies/IoT-Lifecycle`
[42] `ttp://elite.polito.it/ontologies/poweront.owl`
[43] `http://ontology.fiesta-iot.eu/ontologyDocs/fiesta-iot/doc`

ing sensors and observations. The central class is *ssnx:Observation*, which is related with *ssnx:Sensor* who made it, the property it observes (*qu:Quantity-Kind*) and the temporal and location context. Furthermore, the sensor is related with the unit of measurement (*qu:Unit*) used to represent the observation value.

The IoT-Lite Ontology[44] [51] is a lightweight ontology planned to be used by other independent platforms in the open calls of H2020 project FIESTA-IoT. It is an specialization of the previous SSN ontology designed with a clear purpose of defining only the most used terms when searching for IoT concepts in the context of data analytics such as sensory data, location and type. The ontology's lightweight allows the representation and use of IoT platforms without consuming excessive processing time when querying the ontology. However it is also an ontology that can be extended in order to represent IoT concepts in a more detailed way in different domains. The ontology is aimed to be simple, as it is considered as one of its requirements, and it is linked with other well-known and widely used ontologies such as SWEET[45] (Semantic Web for Earth and Environmental Terminology) and the previous version of the SSN.

IoT-Lite is built around the main three concepts which, according to IoT-Lite authors, are necessary in any ontology describing IoT: objects/entities, resources/devices, and services. However, the coverage of the ontology is limited to upper-level concepts, rather than representing types of devices as subclasses of *ssnx:SensingDevice* (e.g. thermometer) or units of measurements as subclasses of *qu:Unit* (e.g. degrees celsius).

Although the vocabularies used in IoT-Lite are aligned with their generalized counterparts, the representation of the key concepts in sensor-related environments (e.g. sensor, action and observation) is limited.

The M3-lite taxonomy[46] is a light version of the M3 ontology [52], designed to meet FIESTA-IoT ontology's requirements. M3-lite follows a modular design and provides links with other IoT-related ontologies to facilitate interoperability. These links are represented with the *rdfs:seeAlso* utility property.

The main purpose of the M3-lite taxonomy is to extend the representation of concepts that are not covered by the SSN ontology in a rather detailed way. In fact, M3-lite defines over 30 types of actuators (as subclasses of *iot-lite:ActuatingDevice*), over 100 types of sensors (as subclasses of *ssnx:SensingDevice*), over 170 types of quantities (as subclasses of *qu:Quantity-Kind*) and over 90 classes of units of measure (as subclasses of *qu:Unit*). Furthermore, the scope of the taxonomy is not limited to a single domain. In fact, it covers 12 different IoT application domains.

2.2.7. SmartEnv Ontology

The SmartEnv ontology[47] [53] proposes a generic ontology for sensorized environments with at least one inhabitant or user. The ontology is a network of 8 different ontology modules. Each module is represented in the form of a pattern to modularize the proposed solution, and it is represented as general as possible avoiding strong dependencies between the modules to manage the representational complexity of the ontology. Furthermore, the modularization allows the update of concepts with the minimum change propagation on the entire ontology, and individual patterns can also be used in isolation for some specific reasoning tasks (e.g., in order to avoid issues with reasoning complexity or clashes in the relations to foundational ontologies). The basis of these ontology modules are extracted from the SOSA/SSN ontology and DUL ontology, however with a number of specializations, either in the form of extension of class hierarchies or updating links between concepts.

2.2.8. The S3N Ontology

The Semantic Smart Sensor Network (S3N) ontology[48] [54] is an extension of the SOSA/SSN ontology to model the adaptation capabilities of Smart-Sensors to different contexts of use. The concept of Smart-Sensor is based on a sensor's ability to acquire data thanks to its embedded sensors, to process this data thanks to the algorithms implemented by its microcontroller, to communicate indicator values, and to be reprogrammable and reconfigurable. The ontology describes Smart-Sensors, their different computation and communication profiles, and the manner in which different algorithms are selected and loaded. The three main classes introduced in the S3N are the following:

- *s3n:MicroController*: Representing compact integrated circuits that consist of a processor, some

[44] http://www.w3.org/Submission/iot-lite/
[45] https://sweet.jpl.nasa.gov/
[46] http://ontology.fiesta-iot.eu/ontologyDocs/fiesta-iot/doc
[47] https://w3id.org/smartenvironment/smartenv.owl
[48] https://github.com/s3n-ontology/s3n/blob/master/s3n.ttl

memory, and input/output peripheral on a single chip, and it is designed to control a certain operation in an embedded system.

- *s3n:CommunicatingSystem*: Representing systems that enable the information exchange with other systems on some network.
- *s3n:SmartSensor*: Representing Smart-Sensors, which are composed of one or more basic sensors with a microcontroller.

2.2.9. PROV-O

PROV-O[49] [55] (PROVenance Ontology) is a lightweight ontology that provides a set of classes, properties, and restrictions that can be used to represent and interchange provenance information generated in different systems and under different contexts. These classes and properties are defined such that not only they directly represent provenance information, but they can also be specialized for modelling application-specific provenance details in a variety of domains.

The following three classes represent the core of PROV-O:

- An individual of *prov:Entity* is a physical, digital, conceptual, or other kind of thing with some fixed aspects; entities may be real or imaginary.
- An individual of *prov:Activity* is something that occurs over a period of time and acts upon or with entities; it may include consuming, processing, transforming, modifying, relocating, using, or generating entities.
- An individual of *prov:Agent* is something that bears some form of responsibility for an activity taking place, for the existence of an entity, or for another agent's activity.

2.3. Spatio-Temporal and Unit context ontologies

Observations and actuations are the central elements of the problem tackled in this article, and their values and result representation play an important role. Spatial, temporal, and units of measurements of these values are a context information that may differ in nature and granularity levels. Next, ontologies representing such context of observations and actuations are reviewed.

[49]https://www.w3.org/TR/prov-o/

2.3.1. Time

Since nearly everything is liable to undergo change, the notion of time features in the discourse about any subject. Many ontologies defining temporal context exist [56,57,58,59,60], even though the most commonly used ontology is the Time Ontology in OWL[50] [61] (OWL-Time).

OWL-Time is a W3C recommendation representing temporal concepts for describing the temporal properties of resources. The vocabulary expresses facts about topological relations among instants and intervals, together with information about durations and temporal position including date-time information. Time positions and durations may be expressed using either the conventional (Gregorian) calendar and clock, or using another temporal reference system such as Unix-time, geologic time, or different calendars.

2.3.2. Location

Together with time, spatial location is the other primary aspect that may help specifying a context. The WGS84 Geo Position[51] is a vocabulary for representing latitude, longitude and altitude information in the WGS84 geodetic reference datum. Another approach proposes a more detailed ontology to describe the location of device-based services that occur in ubiquitous computing environments [62]. GeoSPARQL [63] is the OGC (Open Geospatial Consortium) standard that not only defines an extension to the SPARQL query language, but also defines a vocabulary for representing geospatial data in RDF.

2.3.3. Units of measurements

Units of measurement play a key role in many engineering and scientific applications, and the correct handling of the scale is of utmost importance in most fields. Therefore, nowadays there are numerous ontologies describing units of measurement and their relations. Keil et al. [64] evaluate and compare different ontologies for modelling units of measurements and one of the main findings is that reviewed ontologies use different terms to refer to the same concepts. For instance, the concept “kind of quantity", is denoted as “physical quality" by MUO[52] (Measurement Units Ontology), and as “quantity kind" by QU[53] (Ontology for Quantity Kinds and Units) and QUDT[54] (Quanti-

[50]https://www.w3.org/TR/owl-time/
[51]https://www.w3.org/2003/01/geo/
[52]http://idi.fundacionctic.org/muo/
[53]https://www.w3.org/2005/Incubator/ssn/ssnx/qu/qu.owl
[54]http://www.qudt.org/

ties, Units, Dimensions and Data Types Ontologies). OBOE[55] (Extensible Observation Ontology), OM[56] (Ontology of Units of Measure) and SWEET do not provide an explicit class for this concept, but they model the respective notions as subclasses of “physical characteristic" (OBOE), “quantity" (OM), and “property" (SWEET).

The use of any of the aforementioned ontologies for representing observation results, means that quantity values are usually represented as OWL individuals linked to numeric values and a unit of measure. Next, QUDT and another approach (which is not covered in the aforementioned survey) are reviewed.

QUDT. QUDT[57] is an initiative sponsored by the NASA to formalize Quantities, Units of Measure, Dimensions and Types using ontologies. QUDT is organized as a catalogue of quantity kinds and units of different disciplines (e.g. acoustics or climatology). A quantity (*qudt:Quantity*) is the central element which represents a measurement of an observable property of a particular object, event or physical system. The quantity is related with the context of the measurement, and the underlying quantity kind remains independent of any particular measurement. A quantity kind is distinguished from a quantity in that the former is a type specifier, while the latter carries a value.

The dimensional approach of QUDT relates each unit to a system of base units using numeric factors and a vector of exponents defined over a set of fundamental dimensions. By this means, each base unit’s role is precisely defined in the derived unit. Furthermore, this allows reasoning over quantities as well as units.

Although at the moment of writing this article there are efforts towards the development of a second version of QUDT, these ontologies have only been partly published.

The following triples would represent a 29°C quantity value in QUDT:

```
:temp01 rdf:type qudt:QuantityValue;
   qudt:unit unit:DegreeCelsius;
   qudt:numericValue "29"^^xsd:double.
```

[55]https://code.ecoinformatics.org/code/semtools/trunk/dev/oboe/
[56]http://www.ontology-of-units-of-measure.org/page/om-2
[57]http://www.qudt.org/

UCUM Datatypes. The work presented by Lefrançois et al. [65] leverages UCUM (Unified Code of Units of Measure), a code system which aims at including units of measures currently used in international sciences, engineering, and business.

This proposal is different to the rest of the aforementioned ontologies representing units of measurements and related concepts. The proposed lexical space is the concatenation of a *xsd:decimal* value, at least one space, and a unit chosen from the case sensitive version of the UCUM code system. The value space corresponds to the set of measures, or quantity values as defined by the International Systems of Quantities. Using the UCUM datatypes requires only one triple to link a quantity to a fully qualified value, which is a reduction from the at least three triples needed in the aforementioned proposals.

```
:temp01 sosa:hasSimpleResult
         "29 Cel"^^cdt:temperature.
```

Furthermore, custom mechanisms to canonicalize literals based on external descriptions of units of measurements are not required. Therefore, one of the main advantages of the use of UCUM Datatypes lies in the lighter datasets and simpler queries achieved. However, at the time of writing this article, this work has not yet been implemented in the main RDF stores.

2.4. Discussion

Ontologies like ifcOWL are necessary to convey data registered in standard formats (like IFC files) to the semantic realm (like RDF files). These ontologies enable the automatic conversion of big quantities of data to leverage capabilities offered by the semantic technologies. However, such ontologies may be inadequate for a direct use in some scenarios due to their inconvenient, complex and often counter-intuitive conceptualization of data for the task at hand.

The documentation of ontologies is an often overlooked aspect, although potential users may be tempted to design their own ontologies rather than reusing/re-engineering an existing one when doubts about the meaning of terms arise. As a matter of fact, it is of utmost importance to provide proper descriptions of the ontology itself (e.g. authors or licenses) as well as of the classes and properties (e.g. labels and textual definitions) defined in the ontology if its reuse is aimed. Specially in ontologies with a high number of classes and/or properties a lack of careful documentation with explanatory descriptions of the intended meanings of their terms becomes a hurdle to their reuse. This situation may be present in ontologies such as DogOnt, ThinkHome, ifcOWL and Brick. Worse still, the lack of public access to ontologies, as it happens with EEOnt, makes them impossible to analize or reuse.

A trend towards a pattern-based design tends to produce modular ontologies that are more understandable and more easily extended or reengineered when necessary. The initial SSN ontology may be an example of this pattern-based design, and IoT-O and FIESTA-IoT ontologies may be considered extensions of such initial SSN. Moreover, when some undesirable design decisions on the original SSN were spotted, its reengineering to the new SOSA/SSN ontology was clearly affordable. ODPs promote the conceptualization of concise and simple ideas that may ease the usage, reuse and extension of ontologies. For instance, SmartEnv and S3N were developed as SOSA/SSN extensions. SEAS and BOT are other representative ontologies of this pattern-based design. Furthermore, SOSA/SSN, SEAS, and BOT are presented with a nice documentation.

Sometimes vocabularies play a similar role to catalogues. In such cases, a clear definition of the desired scope, a well explained criteria for the term hierarchy and classification, and a comprehensive coverage of the needed concepts makes a difference. The M3-lite taxonomy can be considered an example of these vocabularies.

Finally, the explicit alignment of terms from different ontologies as well as the mapping to upper-level ontologies promotes interoperability. More comprehensive alignments are favoured between clearly conceptualized and well documented ontologies. BOT offers a set of mappings to other domain ontologies such as ifcOWL, Brick, and DogOnt. Both SOSA/SSN and SEAS publish collections of precise mapping files to other related ontologies. As for SAREF, it is claimed to be aligned with other ontologies, even though these alignments are a set of concept pairings in an Excel sheet without an explicit indication of the precise relationship between each pair of concepts.

Summarizing, a concise representation of appropriate concepts, covering an adequately limited scope, accompanied by a well explained documentation, and augmented with the proper and most complete alignment with other related and upper level ontologies, definitely contribute to the reuse of an ontology.

3. The EEPSA Ontology

Towards the incorporation of the Semantic Technologies in the assistant that supports data analysts through the KDD process, it is of utmost importance to rely on proper ontologies and vocabularies that codify the required knowledge and enables a proper annotation of the data. Previous sections introduced the main areas of discourse of the problem at hand and motivated the need of an ontology that may be the cornerstone of such an assistant.

This section describes the EEPSA (Energy Efficiency Prediction Semantic Assistant) ontology[58] which focuses on energy efficiency and thermal comfort in buildings but it is aimed at being reusable and easily customizable for other use cases in similar domains.

3.1. Ontology Development Methodology

Ontologies must be carefully designed and implemented, as these tasks have a direct impact on their final quality. Therefore, the use of well-founded ontology development methodologies (e.g. On-To-Knowledge [66] and DILIGENT [67]) is advised. For the development of the EEPSA ontology, the NeOn Methodology [68] was followed mainly because it does not prescribe a rigid workflow but instead it suggests a variety of paths. The NeOn Methodology comprises 9 scenarios supporting different aspects of the ontology development process, and for the EEPSA ontology the following scenarios were applied:

- Scenario 1: From specification to implementation. In this scenario, the OSRD (Ontology Requirement Specification Document [69]) was created, which collected the ontology purpose, its intended users, and the set of ontology requirements in the form of Competency Questions (CQ). For building and maintaining the ontology, the Protégé[59] [70] tool version 5.1.0 was used and for managing the different versions of the ontology, the Git version-control system.
- Scenario 7: Reusing ontology design patterns. In this scenario, existing ODPs were reviewed (e.g. from the ODP repository OntologyDesignPatterns.org[60]). Since existing ODPs could not

[58] `https://w3id.org/eepsa`
[59] `https://protege.stanford.edu/`
[60] `http://ontologydesignpatterns.org`

satisfy the requirements captured in the OSRD, a set of basic ODPs were developed. These ODPs were used as building blocks on top of whom the rest of the EEPSA ontology was developed.

- Scenario 3: Reusing ontological resources and Scenario 4: Reusing and re-engineering ontological resources. The reuse of ontological resources built by others that have already reached some degree of consensus is a good practice in ontology development processes [71]. In this scenario, a set of ontologies were reviewed, assessed and compared to decide whether they were suitable for reuse. In some cases, these ontologies were re-engineered prior to their reuse.

The following competency questions summarize basic requirements for assisting data analysts through a KDD process for solving energy efficiency and thermal comfort problems in buildings. Therefore, the development of a core ontology that satisfies them is a prime task.

- CQ01: What are the qualities that influence a feature of interest?
- CQ02: What are the qualities that affect a given quality of a feature of interest?
- CQ03: Which feature of interest does a given quality belong to?
- CQ04: What are the observations/actuations performed by a given procedure?
- CQ05: What are the observations/actuations performed by a given sensor/actuator?
- CQ06: What are the procedures implemented by a given sensor/actuator?
- CQ07: What are the features of interest on a given observation/actuation?
- CQ08: What are the qualities sensed/actuated by a given observations/actuations?
- CQ09: What are the features of interest of a given sensor/actuator?
- CQ10: What are the qualities sensed/actuated by a given sensor/actuator?
- CQ11: Which is the value of an observation?
- CQ12: When was an actuation generated?
- CQ13: For what time interval or instant is valid an observation?
- CQ14: For what spatial location is valid an observation?

For each competency question CQn, a twin competency question CQn^{i} can be considered, which consists in rephrasing the question in the opposite direction. For instance, $CQ01^{i}$ would be defined as “What is the feature of interest influenced by a given quality?". In terms of a SPARQL query, it means that the query variable is moved from the subject position to the object position, or the other way round, in the triple pattern.

3.2. Developing the EEPSA ontology on top of ODPs

In ontology development processes, recurrent design problems may arise. Indeed, these problems may happen during the ontology conceptualization activity, the ontology formalization activity, or during the ontology implementation activity. An ODP is a modelling solution to solve this kind of problems [72]. Ideally, ODPs should be extensible but self-contained, minimize ontological commitments to foster reuse, address one or more explicit requirements (such as use cases or competency questions), be associatable to an ontology unit test, be the representation of a core notion in a domain of expertise, be alignable to other patterns, span more than one application area or domain, address a single invariant instead of targeting multiple reocurring issues at the same time, follow established modelling best practices, and so forth [73].

Developing the EEPSA ontology on top of ODPs was found convenient due to the great flexibility provided by this modelling solution, which allows a proper segmentation of the intended conceptualization. In this case, the considered CQs were divided in three subsets: {CQ01, CQ02, CQ03}, {CQ04, CQ05, CQ06, CQ07, CQ08, CQ09, CQ10} and {CQ11, CQ12, CQ13, CQ14}. In order to solve each of those subsets, an ODP was defined. The proposed ODPs are inspired by existing ontologies and ODPs which address similar CQs but they do not fully satisfy those previously enumerated.

Even though these ODPs are motivated by energy efficiency and thermal comfort problems in buildings, they are designed to be applicable to similar problems in different use cases. Therefore, for each ODP a set of alignments or mappings are developed. These alignments target domain ontologies as well as upper-level ontologies, as setting mappings to a common upper ontology alleviates integration problems [11], helps to ensure clarity in modelling and avoids errors that have unintended reasoning implications [46]. These alignments are kept in separate files and are available online in each ODP’s documentation page. Furthermore, an instantiation example of each ODP is shown in Appendix A, along with SPARQL queries that solve some CQs.

Next, the three proposed ODPs are presented: the AffectedBy ODP[61], the EEP (Execution-Executor-Procedure) ODP[62] and the RC (Result-Context) ODP[63].

3.2.1. The AffectedBy ODP

Data analysts dealing with energy efficiency problems in buildings would benefit from a resource that supports the discovery of relevant variables that affect the environment of a given space or another feature of interest. Any of these variables will be represented as qualities of a feature of interest. Specifically, the competency questions CQ01, CQ02 and CQ03 must be considered. Therefore, the conceptualization must include classes representing features of interest (*aff:FeatureOfInterest*) and their qualities (*aff:Quality*).

The SOSA/SSN ontology contains a building block that may be useful for this matter. However, an inadequacy was spotted. The *ssn:Property* class is textually defined as “a quality of an entity. An aspect of an entity that is intrinsic to and cannot exist without the entity". Furthermore, the *ssn:Property* class is linked to the *sosa:FeatureOfInterest* class with the *ssn:isPropertyOf* object property. Nevertheless, this object property is not functional, so the following triples can be found in a triple set annotated with SOSA/SSN terms:

```
:temperature rdf:type ssn:Property;
        ssn:isPropertyOf :room03.
:room03 rdf:type sosa:FeatureOfInterest.

:temperature ssn:isPropertyOf :room07.
:room07 rdf:type sosa:FeatureOfInterest.

:room03 owl:differentFrom :room07.
```

According to the aforementioned *ssn:Property*'s class textual definition, individual *:temperature* is intrinsic to and cannot exist without the existence of individual *:room03*. However, the triples shown contradict such definition because the individual *:temperature* is a quality of different entities (namely a quality of individual *:room03* and individual *:room07*).

A recent publication about the SOSA/SSN ontology [45] is aware of this possibility and explicitly expresses that “multiple observations across different features of interest or by different sensors or both can measure the same generic feature". The publication also recognizes the choice to represent observable properties as inherent characteristics specific to a feature of interest. Therefore, the SOSA/SSN ontology allow different ways of modelling observable properties and it is expected that “communities and applications to develop their own approaches to building catalogues of observable properties and choosing appropriate levels of specificity". However, the fact that different stakeholders adopt different modelling options may derive in interoperability problems.

This issue is tackled in the SEAS Feature of Interest ontology, where an ODP to describe features of interest and their qualities is defined. In this pattern, the *seas:isPropertyOf* object property links a *seas:Property* (which is equivalent to the class *ssn:Property*) to a *seas:FeatureOfInterest* (which is equivalent to the class *sosa:FeatureOfInterest*), and it is declared as subproperty of *ssn:isPropertyOf*. However, *seas:isPropertyOf* is functional. Therefore, it represents more faithfully the textual definition of *ssn:Property*.

The AffectedBy ODP defines the *aff:belongsTo* object property as functional to support the notion that a quality is intrinsic to the feature of interest to which it belongs. It is defined with *aff:Quality* as domain and *aff:FeatureOfInterest* as range, and it solves CQ03. Furthermore, the following axiom formalizes that every quality belongs to a feature of interest:

$$\textit{aff:Quality} \sqsubseteq \exists \textit{aff:belongsTo.aff:FeatureOfInterest}\ .$$

The SEAS Feature of Interest ontology also defines the *seas:derivesFrom* object property which links a *seas:Property* to another *seas:Property* it derives from. This object property is defined as a symmetric property. However, this constraint is unnecessary for the use case considered in this article and sometimes even inappropriate. For instance, the temperature of individual *:room03* may derive from the occupancy of the room, but the occupancy does not necessarily derive from the temperature of the room.

In order to tackle this specific issue and to solve CQ02, the *aff:affectedBy* object property is introduced. This property has class *aff:Quality* both as its domain and its range, and plays a slightly different role compared with *seas:derivesFrom*. In fact, *aff:affectedBy* is declared to be transitive.

In addition, the SEAS Feature of Interest ontology contains a textual comment that, although relevant, it is not materialized as an axiom. It is intended that:

[61] https://w3id.org/affectedBy
[62] https://w3id.org/eep
[63] https://w3id.org/rc

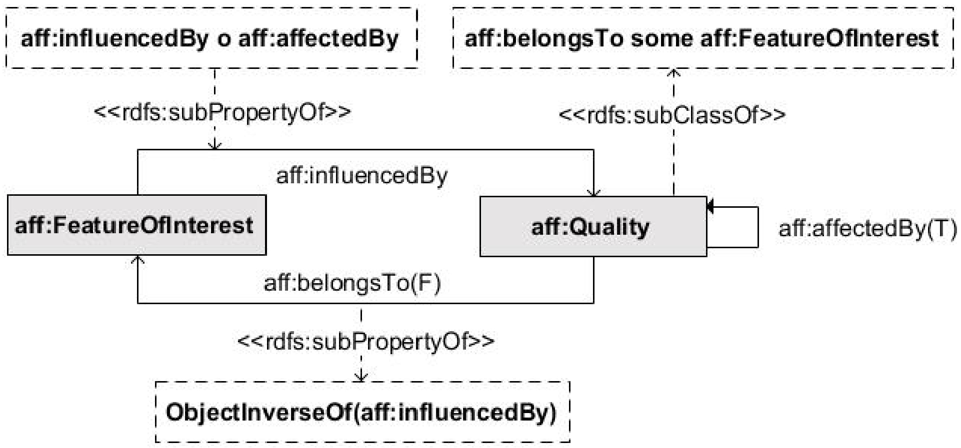

Fig. 1. The AffectedBy ODP. (F) represents functional and (T) transitive properties.

seas:hasProperty ∘ *seas:derivesFrom* ⊑ *seas:hasProperty* .

The inconvenience of adding such a property chain axiom is that *seas:hasProperty* and its inverse become non-simple object properties and therefore they cannot be used in cardinality constraint expressions due to undecidability issues.

In order to solve CQ01, the object property *aff:influencedBy*[64] with *aff:FeatureOfInterest* as its domain and *aff:Quality* as its range is introduced, alongside with the next property chain axiom:

aff:influencedBy ∘ *aff:affectedBy* ⊑ *aff:influencedBy* .

In contrast to the aforementioned SEAS case, the selected set of axioms in the AffectedBy ODP do not cause any undecidability problem.

Finally, the property axiom representing that *aff:belongsTo* is subproperty of the inverse of *aff:influencedBy* is introduced in the AffectedBy ODP.

A diagram of the AffectedBy ODP is shown in Figure 1.

AffectedBy ODP Alignments. The AffectedBy ODP is aligned with the SOSA/SSN ontology and the SEAS Feature of Interest ontology. Furthermore, it is mapped with the upper-level DUL ontology. These alignments are kept in separate files and are available online in the AffectedBy ODP's documentation page `https://w3id.org/affectedBy`.

[64]In the previous version of the AffectedBy ODP [74] this object property was named *aff:hasQuality*. However, it was renamed after *aff:influencedBy* to avoid misleading interpretations.

3.2.2. The EEP ODP

Another interesting information for data analysts working on energy efficiency and thermal comfort problems in buildings could addressed by CQ04, CQ05, CQ06, CQ07, CQ08, CQ09 and CQ10. These CQs are the requirements considered for the EEP ODP.

It may be questionable why competency questions related to results of observations or actuations are disregarded in this ODP, specially because it is common to include this information as parameters of observations or actuations. However, there are some modelling alternatives such as the SEAS Evaluation ontology, where the qualification of the value of a *seas:Property* is preferred. Moreover, different conceptualizations of the result and their spatio-temporal context may be conceived depending on the application. This is the rationale behind designing a separate ODP (presented in the next subsection) to represent result-related matters. Such a design intends to improve the reusability of the proposal, allowing users to easily replace such ODP if they are not satisfied with its modelling decision.

The aforementioned CQs (CQ04 to CQ10) have been tackled by the SOSA/SSN ontology. However, a set of triples annotated with SOSA/SSN (for instance the set shown in Figure 2) cannot properly solve a question like CQ10[i]: which is the sensor that observes the temperature of *:room07*?

```
:sensor1  sosa:madeObservation  :obs1;
        sosa:observes  :temperature.
:temperature  ssn:isPropertyOf  :room03.
:obs1  sosa:hasFeatureOfInterest  :room03.

:sensor2  sosa:madeObservation  :obs2;
        sosa:observes  :temperature.
```

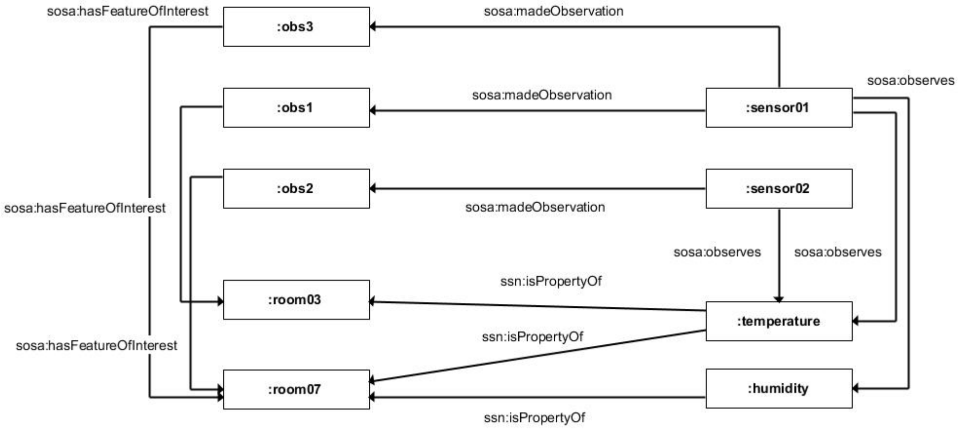

Fig. 2. A SOSA/SSN annotated set of triples.

```
:temperature ssn:isPropertyOf :room07.
:obs2 sosa:hasFeatureOfInterest :room07.

:sensor1 sosa:madeObservation :obs3;
         sosa:observes :humidity.
:humidity ssn:isPropertyOf :room07.
:obs3 sosa:hasFeatureOfInterest :room07.
```

The rationale behind this issue is that there is no property directly linking sensors to features of interest, and moreover, composition of properties that link them through the *sosa:Observation* class are not sufficiently constrained.

The SEAS PEP ontology generalizes the core concepts of SOSA/SSN (i.e. Observation, Actuation, Sensor, Actuator, and Procedure). The proposed Execution-Executor-Procedure (EEP) ODP is an adaptation of the PEP ontology to fully satisfy the required competency questions, overcoming the indicated weaknesses about SOSA/SSN.

The EEP ODP imports the AffectedBy ODP alongside with its notion that a quality is intrinsic to the feature of interest it belongs to. Apart from the two classes imported from the AffectedBy ODP (i.e. *aff:FeatureOfInterest* and *aff:Quality*), the EEP ODP consists of three more classes: *eep:Execution*, *eep:Executor*, and *eep:Procedure* (see Figure 3). An individual of *eep:Execution* is an event upon a quality of a feature of interest, produced by an agent by performing a procedure. As for an individual of *eep:Executor*, it is an agent capable of performing tasks by following procedures. Lastly, an individual of *eep:Procedure* is a description of some actions to be executed by agents.

Note that individuals of class *eep:Execution* can be abstractly represented by a ternary relationship of its executor, the procedure used to produce the execution, and the quality of the feature of interest being considered. Accordingly, the class *eep:Execution* is the domain of the three functional object properties: *eep:madeBy*, *eep:usedProcedure*, and *eep:onQuality*. Moreover the following axioms are introduced:

$eep{:}Execution \sqsubseteq \exists eep{:}madeBy.epp{:}Executor$,
$eep{:}Execution \sqsubseteq \exists eep{:}onQuality.eep{:}Quality$, and
$eep{:}Execution \sqsubseteq \exists eep{:}usedProcedure.eep{:}Procedure$

The object property *eep:madeBy* links an execution to the agent that performs the action; the object property *eep:usedProcedure* links an execution to the procedure that describes the task to be performed; and the object property *eep:onQuality* links an execution to the quality concerned by the execution. These three functional object properties jointly with the functional *aff:belongsTo* form the backbone of the EEP ODP.

The remaining object properties are: *eep:implements*, linking executors to procedures; *eep:hasFeatureOfInterest*, linking executions to features of interest; *eep:forQuality*, linking executors to qualities; and *eep:forFeatureOfInterest*, linking executors to features of interest. The values of all of them are inferred by the values of the four functional properties that form the backbone, due to the corresponding property chain axioms included in the EEP ODP:

$eep{:}madeBy^{-1} \circ eep{:}usedProcedure \sqsubseteq eep{:}implements$,
$eep{:}onQuality \circ eep{:}belongsTo \sqsubseteq eep{:}hasFeatureOfInterest$,
$eep{:}madeBy^{-1} \circ eep{:}onQuality \sqsubseteq eep{:}forQuality$, and
$eep{:}forQuality \circ eep{:}belongsTo \sqsubseteq eep{:}forFeature$-

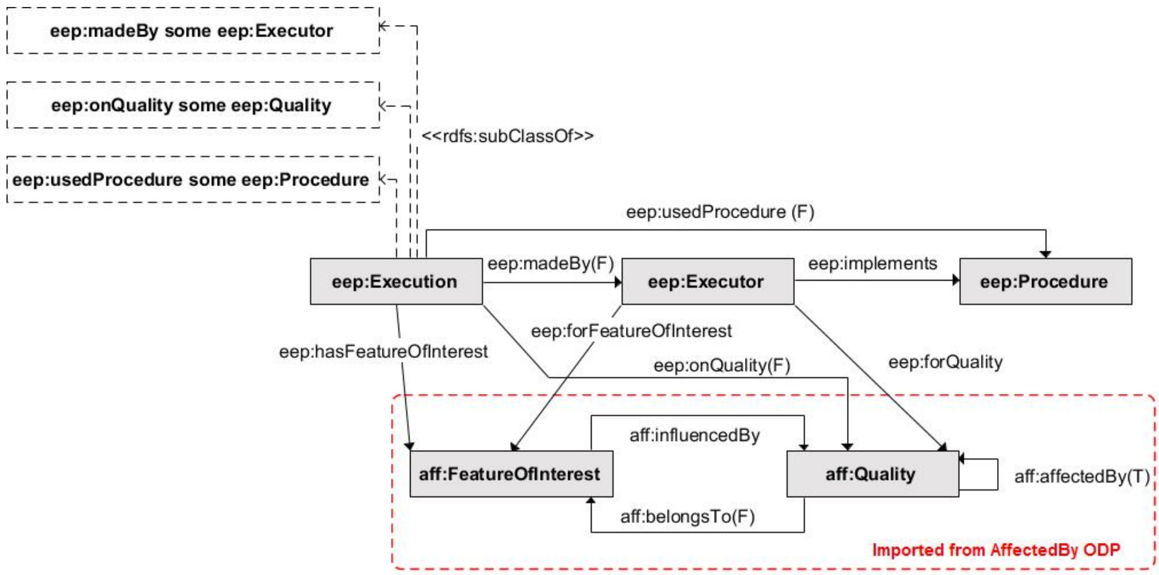

Fig. 3. The Execution-Executor-Procedure (EEP) ODP. (F) represents functional and (T) transitive properties.

OfInterest .

EEP ODP Alignments. The EEP ODP is aligned with the SOSA/SSN ontology, the PEP ontology and PROV-O. Furthermore, it is mapped to the upper-level DUL ontology. These alignments are kept in separate files and are available online in the EEP ODP's documentation page `https://w3id.org/eep`.

3.2.3. *The RC ODP*

Although the AffectedBy and EEP ODPs alleviate much of the data analysts' information needs, they may still require from data representing the results of the executions and their contexts. For example: which is the value of an observation? Or when was an actuation performed? This information may be collected answering the competency questions CQ11, CQ12, CQ13 and CQ14.

Every ontology or ontology network covering observations or actuations need to take into account the representation of these actions' results. For instance, the SOSA/SSN ontology uses the *sosa:hasResult* object property, the IoT Application Profile (IoT-AP) ontology[65] [75] uses *iotap:hasObservationValue* and om-lite uses *om-lite:result* object property. Values of these properties can be complex objects that usually include units of measurement, the measurement value, and some other optional parameters. However, sometimes a simple representation with a literal type value may suffice. In order to tackle these situations SOSA/SSN proposes the *sosa:hasSimpleResult* datatype property. Furthermore, properties representing results are typically associated to observations and actuations, even though there are alternative modelling options. For instance, in the SEAS ontology network, the SEAS Evaluation ontology associates *seas:value* and *seas:simpleValue* properties to the *seas:Property* class.

With respect to the proposed Result-Context (RC) ODP (shown in Figure 4), the representation of both complex and simple results is modelled with the object property *rc:hasResult* and the datatype property *rc:hasSimpleResult* respectively. This way, CQ11 is solved.

There are occasions in which parameters referring to temporal and spatial aspects may be necessary to qualify a result. Regarding the representation of temporal aspects, the SOSA/SSN ontology distinguishes between the time when the result of an observation, actuation, or sampling applies to the feature of interest (with the object property *sosa:phenomenonTime*) and the instant of time when such an observation, actuation or sampling was completed (with the datatype property *sosa:resultTime*). The phenomenon time is specified with an individual of OWL-Time ontology's *time:TemporalEntity* class as it may be either an instant, an interval of time, or even a temporal complex.

[65]`http://stlab.istc.cnr.it/IoT-AP/IoT-AP.rdf`, not available at the moment of writing this article.

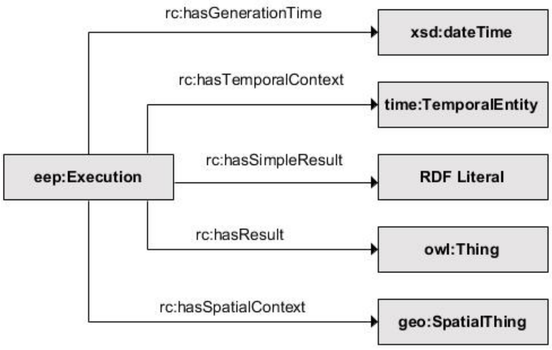

Fig. 4. The Result-Context (RC) ODP.

Meanwhile, the result time describes an instant represented with *xsd:dateTime*. As for the SEAS Evaluation ontology, the temporal context is modelled with the *seas:hasTemporalContext* object property that links an evaluation with its temporal entity modelled as an individual of *time:TemporalEntity*. Furthermore, PROV-O also enables the representation of temporal context. Specifically, the *prov:generatedAtTime* datatype property allows representing the completion of production of a new entity, which would be similar to the *sosa:resultTime* datatype property.

With respect to the RC ODP, it defines two properties: *rc:hasGenerationTime* which is equivalent to *sosa:resultTime*, and *rc:hasTemporalContext* which is equivalent to *sosa:PhenomenonTime*. These definitions solve CQ12 and CQ13 respectively.

When using the SOSA/SSN ontology, spatial aspects of an observation/actuation/sampling are expected to be associated with the feature of interest, the sensor/actuator/sampler or the platform on which they are mounted. However, the representation of this association is not covered by the ontology itself, and has to be made by deferring to external ontologies. By contrast, the SEAS Evaluation ontology leans towards a modelling option which is similar to the temporal aspect. Namely, it defines the *seas:hasSpatialContext* that links an evaluation to its spatial validity context represented as an individual of *geo:SpatialThing* class.

In the RC ODP, the *rc:hasSpatialContext* object property has been defined. It plays *seas:hasSpatialContext* property's same role, but it has *eep:Execution* class as domain, and *geo:SpatialThing* as range. This object property solves CQ14.

RC ODP Alignments. The RC ODP is aligned with the SOSA/SSN and PROV-O ontologies. These alignments are kept in separate files and are available online in the RC ODP's documentation page `https://w3id.org/rc`.

The RC ODP is designed as an horizontal extension of the EEP ODP. But, there are cases where data analysts may require from both ODPs so they need to be used jointly. For example:

– CQ15: Which is the temperature value of room 03 on 2018-11-20 at 16:00?

These three ODPs are the cornerstone of the EEPSA ontology. As a matter of fact, the classes defined by the AffectedBy and EEP ODPs act as stub classes, and for each of them an ontology module is developed. The EEPSA ontology is the addition of the following ontological resources: the three ODPs presented (AffectedBy, EEP and RC), five ontology modules specializing the stub classes defined by these ODPs (FoI4EEPSA, Q4EEPSA, P4EEPSA, EXR4EEPSA and EXN4EEPSA), and an ontology module containing expert knowledge (EK4EEPSA).

3.3. Ontology Modules

The modularization of ontologies consists in partitioning them into independent self-contained knowledge components. A modular approach brings benefits such as the flexibility for component reuse [76], the support for more efficient query answering [77], and the enhancement of components change and evolution [78].

When an already existing ontology is large and monolithic, it needs to be splitted up in order to benefit from the mentioned advantages. There are different techniques that perform ontology partitioning by dividing an ontology into a set of significant modules that together form the original ontology. However, there is no universal way to modularize an ontology and the choice of a particular technique or approach should be guided by the requirements of the application or use case [79].

In order to avoid performing ontology modularization techniques in the future, modularization is advised to be implemented from an early ontology development stage. The EEPSA ontology is modularized by design. Next, the EEPSA ontology modules are presented.

3.3.1. FoI4EEPSA (Feature of Interest for EEPSA ontology module)

This ontology module covers the knowledge specializing the *aff:FeatureOfInterest* class for the EEPSA

Ontology. In the context of this article, a feature of interest is understood as an abstraction of a real world phenomena (object, event, etc). A feature of interest is then described in terms of its qualities, which are qualifiable, quantifiable, observable or operable.

In particular, the FoI4EEPSA ontology module[66] tries to tackle CQs such as the following:

- CQ16: Which building does a given space belong to?
- CQ17: How many spaces does a building have?
- CQ18: In which storey is a given space located?

Different ontologies that cover the representation of the building domain were reviewed in section 2.1, and finally BOT was considered to be reused for basic building topology descriptions.

As for representing building elements, which are also an important part of the domain at hand, the FoI4EEPSA ontology module needs to solve the following CQs:

- CQ19: Which space does a given door belong to?
- CQ20: How many windows does a given space have?
- CQ21: Is a window adjacent to outdoors?

To this end, the PRODUCT ontology was considered. PRODUCT (which at the moment of writing this article is still under development) has a much wider coverage scope than the needed, so its importation would result in increasing EEPSA ontology's size with unnecessary concepts. Therefore, following the simplicity goal of the EEPSA ontology, importing PRODUCT was discarded. Instead, a set of building elements identified in the EEPSA ontology requirements were defined, such as doors (*foi4eepsa:Door*) and windows (*foi4eepsa:Window*). Furthermore, a class *foi4eepsa:ExternalBuildingElement* was defined to represent building elements that face outdoors. This representation mimics the approach followed by EEOnt, and allows the representation of doors and windows that open to the outdoor (via *foi4eepsa:ExternalDoor* and *foi4eepsa:ExternalWindow* classes), as well as external walls (*foi4eepsa:ExternalWall*). These new terms defined in FoI4EEPSA are mapped to the related PRODUCT ontology terms. PRODUCT is in turn aligned with the IFC4 Addendum 2 standard, making the FoI4EEPSA ontology module interoperable.

[66]https://w3id.org/eepsa/foi4eepsa

Last but not least, information related to the building context is also an important aspect. Namely, FoI4EEPSA has to solve the following CQs:

- CQ22: Which is the intended use of the building?
- CQ23: When was the building built?
- CQ24: Which is the gross floor area of the building?

IFC presents a comprehensive collection of property sets (known as PSETs) for describing different aspects of building and building-related context. However, the conceptualization of these properties in ifcOWL as instances of classes (e.g. *ifc:IfcIdentifier* or *ifc:IfcLabel*) is counterintuitive to semantic web principles that would expect OWL properties to represent them. Therefore, inspired by the semantic transformations proposed by Mendes de Farias et al. [21], FoI4EEPSA defines a re-engineering of the relevant properties contained in IFC PSET Building Common and IFC PSET Building collections. Namely, concepts such as *foi4eepsa:hasYearOfConstruction* are used to represent the construction year of a building, and *foi4eepsa:hasMarketCategory* to define the type of use of the building (e.g. residential or commercial).

3.3.2. Q4EEPSA (Quality for EEPSA ontology module)

This ontology module covers the knowledge specializing the *aff:Quality* class, which refers to qualities or aspects of a feature of interest that are intrinsic to and cannot exist without the feature of interest.

In particular, the Q4EEPSA ontology module[67] tries to tackle CQs such as the following:

- CQ25: Which are the actuatable qualities?
- CQ26: Which are the predictable qualities?
- CQ27: Which are the thermal comfort qualities?

In Q4EEPSA two categories of qualities are differentiated. On the one hand, observable qualities of a feature of interest defined by the class *q4eepsa:ObservableQuality*. Bearing in mind the conceptualization of observation proposed by the O&M model followed by the EEPSA ontology, this class comprises qualities that can be observed, estimated and even forecasted. On the other hand, qualities of a feature of interest that can be acted on, defined by the class *q4eepsa:ActuatableQuality*. Qualities that are relevant for the EEPSA's domain of discourse are classified at least in one of the aforementioned classes. Likewise,

[67]https://w3id.org/eepsa/q4eepsa

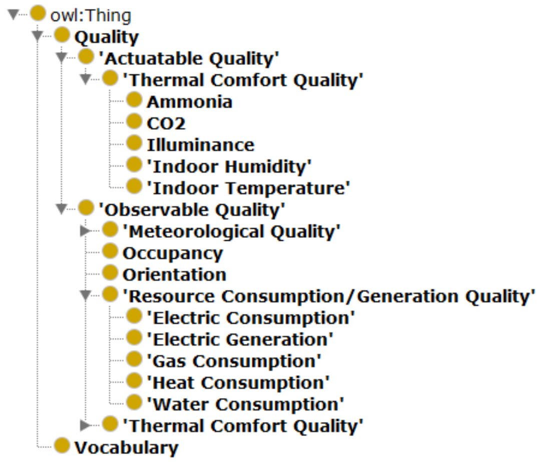

Fig. 5. Overview of the classes defined in Q4EEPSA (visualized in Protégé).

qualities that belong to these categories are also classified into orthogonal groups according to dimensions like their area of interest.

Meteorological qualities such as the solar radiation (*q4eepsa:SolarRadiation*) or the cloud coverage (*q4eepsa:CloudCover*), are defined as subclasses of *q4eepsa:MeteorologicalQuality*, which are observable but not actuatable as defined with the following axiom:

$$q4eepsa{:}MeteorologicalQuality \sqsubseteq q4eepsa{:}ObservableQuality \sqcap \neg q4eepsa{:}ActuatableQuality\ .$$

Qualities related with the thermal comfort within a space such as indoor temperature (*q4eepsa:IndoorTemperature*) and indoor humidity (*q4eepsa:IndoorHumidity*) are represented as subclasses of *q4eepsa:ThermalComfortQuality*. These qualities can be observed and acted on. Furthermore, qualities related to the resource consumption such as water consumption (*q4eepsa:WaterConsumption*) or electric generation (*q4eepsa:ElectricGeneration*), are also defined. These concepts are described as subclasses of *q4eepsa:ResourceConsumptionQuality*, which is observable. However, even though it can be indirectly actuated on (for example with consumption restriction strategies), a consumption is not directly actuatable, so that it is not categorised as a subclass of *q4eepsa:ActuatableQuality*. Some of the mentioned classes are reengineered and reused from the M3-lite taxonomy because it contains a great set of well-organized quality classes.

The Q4EEPSA ontology module is aligned with related ontologies such as SAREF and the SEAS Generic Property ontology[68].

Figure 5 shows an overview of the main Q4EEPSA classes.

3.3.3. P4EEPSA (Procedure for EEPSA ontology module)

This ontology module covers the knowledge specializing the *eep:Procedure* class, which represents workflows, protocols, plans, algorithms, or computational methods specifying how to produce an event.

In particular, the P4EEPSA ontology module[69] tries to tackle CQs such as the following:

- CQ28: What are the actuating procedures?
- CQ29: What are the predictive procedures?
- CQ30: What are the imputation procedures?

P4EEPSA represents four types of procedures: actuating procedures (*p4eepsa:ActuatingProecedure*) specifying how to act on an event; sensing procedures (*p4eepsa:SensingProcedure*) specifying how to sense an event; imputation procedures (*p4eepsa:ImputationProcedure*) specifying how to impute an event; and predictive procedures (*p4eepsa:PredictiveProcedure*) specifying how to predict an event. Such a classification of procedures is a requirement of the data analyst assistant.

3.3.4. EXR4EEPSA (Executor for EEPSA ontology module)

This ontology module covers the knowledge specializing the *eep:Executor* class, which represents agents that produce an event by implementing a procedure.

The EXR4EEPSA ontology module [70] tries to tackle CQs such as the following:

- CQ31: Which type of sensor is a given sensor?
- CQ32: Is a given executor a window actuator?
- CQ33: Is a given executor a predictive model?

EXR4EEPSA concepts are categorised in four different classes: sensors, actuators, predictive models and imputation methods. *exr4eepsa:Sensor* represents agents that implement a procedure to sense a change in a real world's quality. Following SOSA/SSN's conceptualization, a sensor is not necessarily a physical

[68] `https://w3id.org/seas/GenericPropertyOntology`
[69] `https://w3id.org/eepsa/p4eepsa`
[70] `https:/w3id.org/eepsa/exr4eepsa`

device, and it can also be virtual, or even a human being. Sensors are classified in two main classes: meters and environment sensors. On the one hand, the class *exr4eepsa:UtilityMeter* defines a set of meters observing the water, heat, gas or electricity consumption, as well as meters for observing the energy generated (e.g. from photovoltaic panels). On the other hand, sensors observing environment conditions include anenometers (*exr4eepsa:Anenometer*) for sensing wind speed and humidity sensors (*exr4eepsa:HumiditySensor*). Furthermore, these environment sensors include the *exr4eepsa:AirQualitySensor* subclass comprising agents sensing air pollution and gases in the surrounding area (e.g. *exr4eepsa:CO2Sensor*).

exr4eepsa:Actuator represents agents that implement a procedure to act on a real world quality. This concept is more general than *seas:Actuator*, *iot-lite:ActuatingDevice* or *sosa:Actuator* since, similarly to sensors, the agent does not necessarily need to be a device or a physical element. It can be for example a software that switches on or off a light bulb. This class includes a set of common actuators for an energy efficiency problem in tertiary buildings, such as door actuators (*exr4eepsa:DoorActuator*) and window actuators (*exr4eepsa:WindowActuator*).

EXR4EEPSA is not aimed at making an exhaustive representation of different types of sensors and actuators. Instead, it focuses on describing sensors and actuators that are recurrent to energy efficiency and thermal comfort problems in buildings. Furthermore, two additional high-level class of executors are defined in EXR4EEPSA. The first one is *exr:PredictiveModel*, representing agents that implement a predictive modelling procedure to forecast unknown or future outcomes. The second one, the class *exr:ImputationMethod*, describes agents that implement a procedure to compute an estimation of missing values.

Some of these classes are inspired by the M3-lite taxonomy. However, they are not reused because they do not represent the same sensors/actuators (e.g. M3-lite represents only physical sensors, while in the context of EXR4EEPSA sensors are not necessarily physical objects). Some other classes are reengineered and reused from the SEAS Smart Meter ontology[71]. Furthermore, the EXR4EEPSA ontology module is aligned with these two related domain ontologies.

3.3.5. EXN4EEPSA (Execution for EEPSA ontology module)

This ontology module covers the knowledge specializing the *eep:Execution* class. This class represents events or actions made by an agent executing a task implemented by a procedure with respect to a quality of a feature of interest.

In particular, the EXN4EEPSA ontology module[72] tries to tackle CQs such as the following:

- CQ34: Is a given execution an actuation?
- CQ35: Is a given execution an observation?
- CQ36: What are the imputed observations?

To that end, this ontology module defines three main concepts: an observation (*exn4eepsa:Observation*), which is an execution made by an executor to estimate or calculate a quality of a feature of interest; an actuation (*exn4eepsa:Actuation*) which is an execution made by an executor to act upon a quality of a feature of interest; and a missing value (*exn4eepsa:MissingValue*), which happens when executions are empty or null in attributes where a value should have been recorded. Likewise, an observation can be predicted or forecasted (*exn4eepsa:Forecast*), obtained after using an imputation method (*exn4eepsa:-Imputation*), or it can even be an outlier (*exn4eepsa:-Outlier*) when it does not conform to the expected behaviour. Furthermore, EXN4EEPSA also defines the class *exn4eepsa:CollectionOfExecutions*. This class represents a set of executions, such as a sequence of missing values, or the collection of observations forecasted by a predictive model. Furthermore, object properties *exn4eepsa:hasMember* and its inverse *exn4eepsa:isMemberOf* are defined to associate individuals of class *eep:Execution* that belong to a collection of executions, and viceversa.

Such a detailed hierarchy of concepts is motivated by the relevance these concepts may have in data analysis problems. Furthermore, the EXN4EEPSA ontology module is aligned with a set of domain ontologies such as the SOSA/SSN ontology, the SEAS Device ontology, SAREF and om-lite ontology. It is important to note that other ontologies such as SmartEnv and S3N can be indirectly aligned with EXN4EEPSA since they are based on the SOSA/SSN ontology.

[71] `https://w3id.org/seas/SmartMeterOntology`

[72] `https://w3id.org/eepsa/exn4eepsa`

3.3.6. EK4EEPSA (Expert Knowledge for EEPSA ontology module)

This ontology module covers the necessary expert knowledge to provide inferencing capabilities that can be exploited by the data analyst assistant. This module is defined under the supervision of experts in the domain at hand in order to capture task-based knowledge.

In particular, EK4EEPSA[73] tries to tackle CQs such as the following:

- CQ37: What is a naturally enlightened space?
- CQ38: Which types of spaces are in a building?
- CQ39: Which are the qualities affecting a bad insulated space's temperature?

On the one hand, EK4EEPSA defines a classification of types of spaces in buildings. These space definitions are based on their structural features, such as spaces in contact with outdoor environment (*ek4eepsa:-AdjacentToOutdoorSpace*) or spaces located below the ground floor (*ek4eepsa:BelowGroundLevelSpace*). However, other space definitions such as the proposed by the HBC (Human Comfort in Building) ontology[74] [80] may be incorporated, where spaces are mainly characterized by the equipment contained or not within themselves (e.g. *hbc:SpaceWithHeater* or *hbc:SpaceWithoutHeater*). Note that in the scenario tackled in this article, it may be convenient to make heavy usage of axioms expressing sufficient conditions to infer the recognition of individuals in appropriate classes. That is, it may be suitable to use equivalent class axioms with appropriate right hand class expressions, rather than being dependent on explicit assertions only. For instance, the *ek4eepsa:AdjacentToOutdoorSpace* is defined as follows:

ek4eepsa:AdjacentToOutdoorSpace $\equiv$
bot:Space $\sqcap$
$\exists$*bot:hasElement.foi4eepsa:ExternalBuildingElement*

On the other hand, for each space type, qualities that affect their indoor temperature are captured. Such a modelling relies on qualities represented in Q4EEPSA and the axioms defined in the AffectedBy ODP. It is worth noting that this is the only EEPSA ontology module that has dependencies with other EEPSA ontology modules. However, the data analyst assistant has a requisite that needs for the ability to ask for inter-relationships of entities coming from any other modules. For instance, the temperature of an adjacent to outdoor space may be affected by qualities such as the indoor humidity, and the occupancy of the room, as represented in the following axioms:

ek4eepsa:AdjacentToOutdoorSpaceIndoorTemperature
$\sqsubseteq$ $\exists$*aff:affectedBy.q4eepsa:IndoorHumidity*
$\sqcap$ $\exists$ *aff:affectedBy.q4eepsa:Occupancy*
$\sqcap$ $\exists$*aff:affectedBy.q4eepsa:SolarRadiation*
$\sqcap$ $\exists$*aff:affectedBy.q4eepsa:WindSpeed* .

This knowledge modelling can be exploited by application programs and to support data analysts in a proper manner. After knowing which is the type of space at hand, data analysts get to know which are the qualities that are relevant to solve the energy efficiency or thermal comfort problem.

At the moment of writing this article, the EK4EEPSA ontology module solves the presented CQs. However, being an ontology module containing expert knowledge, it is extendible as more requisites are demanded.

3.4. Documentation

A good ontology documentation increases its understandability and potential usability, both by experts in semantics and by people who are not necessarily experts [81]. The documentation of the EEPSA ontology and its ontology modules is generated with WIDOCO (a WIzard for DOCumenting Ontologies) [82] which creates a set of linked enriched HTML pages. These HTML pages are extended with hand-made sections such as the alignments to other ontologies or with ontology usage examples.

W3C's Data on the Web Best Practices [83] states that providing metadata is a fundamental requirement that helps human users and computer applications to understand the data as well as other important aspects that describes a dataset. All the ontological resources presented in this article were annotated following guidelines described by Garijo and Poveda-Villalón [84] as it was considered the most complete guideline among the ones reviewed. As a matter of fact, for each EEPSA ontology module or ODP, both the ontology itself and the classes and properties are annotated with all the recommended terms and some optional terms.

[73] `https://w3id.org/eepsa/ek4eepsa`
[74] `https://w3id.org/ibp/hbc`

Next, the canonical URIs for the different ontology modules documentation are shown[75]:

- EEPSA ontology: `https://w3id.org/eepsa`
- AffectedBy ODP: `https://w3id.org/affectedBy`
- EEP ODP: `https://w3id.org/eep`
- RC ODP: `https://w3id.org/rc`
- FoI4EEPSA ontology module: `https://w3id.org/eepsa/foi4eepsa`
- Q4EEPSA ontology module: `https://w3id.org/eepsa/q4eepsa`
- P4EEPSA ontology module: `https://w3id.org/eepsa/p4eepsa`
- EXR4EEPSA ontology module: `https://w3id.org/eepsa/exr4eepsa`
- EXN4EEPSA ontology module: `https://w3id.org/eepsa/exn4eepsa`
- EK4EEPSA ontology module: `https://w3id.org/eepsa/ek4eepsa`

With regards to the ODPs, they are also available in the ODP repository[76], which collects and makes ODPs available on the web, allowing users to download, propose, and discuss them.

3.5. *Evaluation*

There are many evaluation metrics for assessing ontologies in existing literature [85,86]. Most of them, focus on structural notions without taking into account the semantics, leading to incomparable measurement results [87]. And even though these are valid metrics, they may not be enough to determine the quality of an ontology. In order to avoid biased evaluations, next, the EEPSA ontology and the modules that comprise it are assessed from three perspectives: design correctness, structural metrics, and modularity quality.

3.5.1. *Design Correctness Metrics*

The design correctness is evaluated using OOPS! (OntOlogy Pitfall Scanner) [88], which detects some of the most common pitfalls appearing within ontology developments. OOPS! is available online[77] and evaluates an ontology against a catalogue of 41 potential pitfalls classified into three levels according to their severity: minor, important and critical. This tool was used during the ontology modules development phase, contributing to an early detection of pitfalls, and complementing the manual review of the ontology's correctness. Table 1 summarizes the number of pitfalls detected in the EEPSA ontology and its components.

Table 1
Summary of ontology design correctness evaluation by OOPS!

Ontology	Minor	Important	Critical
EEP	13	2	0
AffectedBy	4	1	0
RC	8	3	0
FoI4EEPSA	7	1	0
Q4EEPSA	4	1	0
P4EEPSA	4	1	0
EXR4EEPSA	4	1	0
EXN4EEPSA	5	1	0
EK4EEPSA	5	1	0

Overall, most ontology modules share the same minor pitfalls "P04: Creating unconnected elements" and "P08: Missing annotations". These pitfalls appear mainly in the stub classes that ontology modules extend (e.g. class *aff:FeatureOfInterest* for the case of the FoI4EEPSA ontology module) as well for the *voaf:Vocabulary* class used to describe the ontology itself. These concepts are adequately annotated and connected in their source ontology module so annotating them again would derive in having duplicated metadata when all ontology modules are imported by the EEPSA ontology. Therefore, these pitfalls are ignored.

Regarding the important pitfalls, the "P10: Missing disjointness" is repeated in all the ontology modules and ODPs. This pitfall arises when an ontology lacks from disjointness axioms between classes or between properties that should be defined as disjoint. However, in the EEPSA ontology modules case, those suggested disjointness axioms are an inconvenient conceptualization constraint, so it was decided not to add those constraints.

3.5.2. *Structural Metrics*

Structural metrics by themselves may not be enough to assess the quality of an ontology or an ontology module, but they may still be relevant to describe an ontology. Protégé has an Ontology Metrics tab[78] that displays entity and axiom counts for the axioms in

[75]All URIs are provided by the `https://w3id.org` redirection service.
[76]`http://ontologydesignpatterns.org`
[77]`http://oops.linkeddata.es/`
[78]`http://protegeproject.github.io/protege/views/ontology-metrics`

Table 2

Summary of ontology structural metrics by Protégé's Ontology Metrics tab (OP = Object Properties, DP = Datatype Properties, * = Imported axioms are not considered).

Ontology	Axioms	Class	OP	DP	Annotation	DL Expressivity
EEP(*)	80	6	8	0	40	ALERIF
AffectedBy	62	3	3	0	31	ALERIF+
RC	40	4	3	2	20	AL(D)
FoI4EEPSA(*)	128	17	0	5	64	AL(D)
Q4EEPSA	197	30	0	0	124	AL
P4EEPSA	40	6	0	0	16	AL
EXR4EEPSA	207	33	0	0	127	AL
EXN4EEPSA	72	9	2	0	36	ALI
EK4EEPSA	81	25	4	0	32	ALC

the active ontology. Table 2 summarizes the structural metrics for the different EEPSA ontology modules, ODPs and the EEPSA ontology itself.

Results show that ODPs are richer from a DL expressivity point of view. They define more constraints, while the rest of the ontology modules are more light weighted. As for the size, most EEPSA ontology modules are rather small (less than 17 classes). The only exception are the Q4EEPSA, EXR4EEPSA and EK4EEPSA ontology modules, which represent over 25 classes. The first two are in charge of representing qualities, sensors and actuators that are typical in problems addressed in the article, so it is understandable to contain a bigger number of classes. The latter, in turn, actually defines only 8 new classes. The rest of the classes are defined in other modules but are necessary to describe the expert knowledge contained in the module.

3.5.3. *Modularity Quality*

The EEPSA ontology's module quality is also assessed based on the guidelines proposed by Khan and Keet [89]. This work creates a comprehensive list of module evaluation metrics as well as a definition of 14 types of ontology modules. For each type of ontology module, it is described which metrics need to be measured and the expected values for a high quality ontology module. In the case of the EEPSA ontology, modules of type T1 (ODP modules: AffectedBy, EEP and RC) and T2 (Subject domain modules: FoI4EEPSA, Q4EEPSA, P4EEPSA, EXR4EEPSA, EXN4EEPSA and EK4EEPSA) are identified. The evaluation is performed with TOMM[79] (Tool for Ontology Module Metrics) and results are available online[80].

Regarding the ODPs, the guidelines suggest that a good quality module should have a small size compared to the original ontology size (i.e. relative size), a small cohesion (i.e. the extent to which entities in a module are related to each other), and be complete. The proposed three EEPSA ODPs satisfy the small relative size and cohesion requirements. However, EEP and RC are not logically complete, as they do not describe terms defined in other ontologies (e.g. *aff:affectedBy* object property in EEP and *eep:Execution* in RC) to avoid duplicated metadata in the final EEPSA ontology.

With regards to the rest of the ontology modules, which can be classified as of type "T2-subject domain modules", they are required to fulfil these criteria to be considered good quality modules: small cohesion, large encapsulation (i.e. "swappability" or ease to exchange a module for another without side effects), small coupling (i.e. the degree of interdependence of a module) and small redundancy (i.e. the duplication of axioms within a set of ontology modules). All the EEPSA ontology modules satisfy these criteria.

3.6. *Ontology Customization by Module Replacement*

Although the EEPSA ontology is aimed at supporting data analysts in energy efficiency and thermal comfort problems in buildings, it is designed to enable its customization to support data analysts in similar prob-

[79] `http://www.thezfiles.co.za/Modularity/TOMM.zip`

[80] `https://github.com/iesnaola/eepsa/tree/master/Evaluation/TOMM`

lems in different types of buildings. Being modularized by design, the EEPSA ontology is expected to be easily modified. Furthermore, as it has been demonstrated that the EEPSA ontology modules are loosely coupled and have a few dependencies between them, this ontology customization can be methodically approached.

The customization of the EEPSA ontology is recommended to be performed via ontology module replacement. That is, existing ontology modules should be replaced with other ontology modules, which can be new modules or extensions of existing ones. This way, the development of customized EEPSA ontologies is expected to be of bounded complexity. In the next section, this ontology customization process is illustrated with a real-world use case.

4. Real-world use case: Poultry farm

In the context of the Internet of Food & Farm 2020 European H2020 project[81] (from now on referred to as IoF2020), one of the trials is aimed at optimizing animal health, production chain transparency and traceability. Within this trial, there is a use case which consists in a poultry farm with a capacity for around 33,000 animals. The farm is equipped with monitoring sensors distributed across the entire building and an automatized ventilation and window system to control indoor climatic conditions. A typical poultry breeding period lasts around 42 days, which can be split in different stages such as chickling or adult stages. Each stage has its own comfort requirements that needs to be fulfilled in order to ensure poultry welfare. Furthermore, farm's building structure, thermal inertia, and outside climatic conditions have a direct effect on the indoor climatic conditions. Currently, farm's ventilation and windows system are set according to the current indoor climatic conditions. That is, no actuation is decided in advance. Figure 6 shows the poultry farm of the use case at hand.

With views to ensuring poultry welfare, farmers could benefit from a system that lets them know if future farm indoor conditions will not meet animals comfort conditions. Such a system could be based on a predictive model forecasting future farm indoor conditions. And data analysts developing such a predictive model could certainly take leverage of the assistant based on the EEPSA ontology.

[81] https://www.iof2020.eu/

Fig. 6. Use case poultry farm.

4.1. Requirements

The EEPSA ontology is designed to assist data analysts in energy efficiency and thermal comfort problems in buildings. Furthermore, it is designed with views to be applicable to similar problems in similar domains. However, each use case may have its own requirements, so every time a new use case from a different domain is faced, it is necessary to identify this use case's requirements and see to which extent the EEPSA ontology satisfies them.

Farms in general have specialized building structures and equipment, such as elements designed to feed animals. Similarly, rearing farms contain special spaces intended for raising animals, which have specific comfort requirements. Poultry farms have specialised monitoring devices installed in their facilities such as scales for birds weighting and moreover, variables that are not measured in common buildings are monitored (e.g. ammonia levels). Likewise, in these contexts there is a very specific domain knowledge that is not within the grasp of everyone and it is usually limited to poultry farming experts. For example, the comfort requirements that animals have in every growth stage or the variables that affect farm's indoor climatic conditions. Summarizing, the system developed for the use case at hand should support data analysts in answering competency questions like the following:

- CQ40: How many breeding spaces are in the farm?
- CQ41: How many troughs are in a given space?
- CQ42: What is the stocking density in a given space?
- CQ43: What is the CO2 level in a given space?

– CQ44: What are the devices installed within a farm?
– CQ45: What are the variables affecting the temperature of a breeding space?

This knowledge is important, and the EEPSA ontology needs to properly capture it to satisfy these requirements. The FoI4EEPSA ontology module describes buildings and building elements, but this knowledge is not specialized in farms, so that some terminology specific to poultry farms is not covered. The Q4EEPSA ontology module in charge of representing qualities of features of interest, does not describe qualities that can be typically found in poultry farms, such as the amount of kilos contained within a space. The EXR4EEPSA ontology module in charge of representing sensors and actuators lacks of the description of some agents monitoring or acting on farm conditions. EXN4EEPSA describing different types of observations and actuations does not represent animal thermal comfort requirements. Furthermore, spaces such as breeding spaces and variables affecting their comfort conditions are also missing in the EK4EEPSA ontology module, as they are far from a common building's casuistry. Therefore, the EEPSA ontology needs to be adequately extended and customized to tackle these issues that are currently uncovered.

4.2. The EEPSA ontology customization

Bearing in mind the existing gap between the current EEPSA ontology and the poultry farm use case requirements, an Ontological Resource Reuse Process was performed in order to find resources that could fill this gap.

The penetration of Semantic Technologies in agriculture is mostly focused on ontologies representing agricultural concepts. But there are also repositories hosting vocabularies for the agricultural domain, which were helpful for this reuse process. AgroPortal[82] [90] is an ontology repository for the agronomy domain which features ontology hosting, search, versioning, visualization, comment, and recommendation. Planteome[83] [91] is another repository of ontologies providing resources for plant traits, phenotypes, diseases, genomes, gene expression and genetic diversity data across a wide range of plant species. Agrisemantics[84] is a catalogue of data standards of different types and formats for the agri-food domain.

AGROVOC[85] is a thesaurus that organizes concepts related to the FAO[86] (Food and Agriculture Organization of the United Nations) including agriculture, food, nutrition, fisheries, forestry and environment. At the moment of writing this article, AGROVOC consists of over 35,000 concepts and it is available in 23 different languages. Furthermore, there are ontologies covering different aspects of the agricultural domain. The Food ontology[87] [92] contains specifications of ingredients, substances, and nutrition facts, and supports a system that assists in the menu planning task for different scenarios. The CROPont ontology [93] describes the crop production life cycle and the FTTO [94] (Food Track&Trace Ontology) is an ontology that aims at enabling information sharing among the different stakeholders along the supply chain. This information sharing supports the food traceability, which can be understood as a part of a complex system in which different business processes collaborates in sharing information. Taking these ontologies into account, the EEPSA ontology customization task is performed.

The representation of poultry farms and related elements is tackled in the new FoI4PFEEPSA ontology module[88]. This ontology is an extension of the original FoI4EEPSA for the poultry farm domain. AGROVOC defines two concepts related to buildings that may be of interest for the matter at hand: "farm buildings" and "poultry housing". AGROVOC defines the former as subclass of "buildings" while the latter is defined as subclass of "housing". Such a fine-grained distinction between buildings and housings does not fit with the conceptualization of the *bot:Building* class. Therefore, these two AGROVOC terms are not reused. Instead, a new *foi4pfeepsa:PoultryHousing* is created and defined as subclass of the *foi4pfeepsa:Farm* class. With regards to equipment related to animal activities, AGROVOC inspires the creation of the *foi4pfeepsa:AnimalHusbandryEquipment* class, which represents the equipment to breed animals. Furthermore, two subclasses of this concept are described: *foi4pfeepsa:Drinker* representing water dispensers, and *foi4pfeepsa:Trough* representing a specific type

[82] http://agroportal.lirmm.fr
[83] http://browser.planteome.org/amigo
[84] http://vest.agrisemantics.org
[85] http://aims.fao.org/en/agrovoc
[86] http://www.fao.org
[87] https://www.bbc.co.uk/ontologies/fo/1.1.ttl
[88] https://w3id.org/pfeepsa/foi4eepsa

of food dispenser. It is worth mentioning that the FoI4PFEEPSA ontology module is aligned with AGROVOC in a separate file available online[89].

The stocking density of a space represents the amount of kilos contained in that given space. This is a very specialized farming term which is not even covered by AGROVOC, and it is not within the scope of the original Q4EEPSA ontology module. Therefore, the use case required the creation of a Q4PFEEPSA ontology module[90] extending Q4EEPSA with the term *q4pfeepsa:StockingDensity*. Likewise, the representation of sensors estimating the stocking density are not covered in EXR4EEPSA, so the new EXR4PFEEPSA ontology module[91] was developed. This new ontology module describes classes such as *exr4pfeepsa:StockingDensitySensor* and *exr4pfeepsa:-WeightScale* for representing scales used to measure bird weight. It is worth mentioning that, due to the specificity of these concepts, none of them was available in AGROVOC or other agricultural ontologies. With regards to the EXN4EEPSA, it was extended specializing the *exn4eepsa:Observation* class. This derived in the new EXN4PFEEPSA ontology module[92] with the definition of classes representing different levels of poultry discomfort: moderate (*exn4pfeepsa:ModerateThermalDiscomfort*), high (*exn4pfeepsa:HighThermalDiscomfort*) and severe (*e-xn4pfeepsa:SevereThermalDiscomfort*). Furthermore, data analysts could also benefit from an ontology module capturing expert knowledge related with the variables affecting specific space types for farms. Therefore, a new EK4PFEEPSA[93] ontology module extended the original EK4EEPSA ontology module, including the description of a class *ek4pfeepsa:Breeding-Space* and variables affecting its indoor conditions.

Figure 7 depicts the ontology modules that conform the EEPSA ontology's customization for poultry farms, that is, the PFEEPSA (Poultry Farm Energy Efficiency Prediction Semantic Assistant) ontology[94].

[89]https://iesnaola.github.io/eepsa/PFEEPSA/FoI4PFEEPSA/alignments/foi4pfeepsa_Alignment_AGROVOC.owl

[90]https://w3id.org/pfeepsa/q4pfeepsa

[91]https://w3id.org/pfeepsa/exr4pfeepsa

[92]https://w3id.org/pfeepsa/q4pfeepsa

[93]https://w3id.org/pfeepsa/ek4pfeepsa

[94]https://w3id.org/pfeepsa

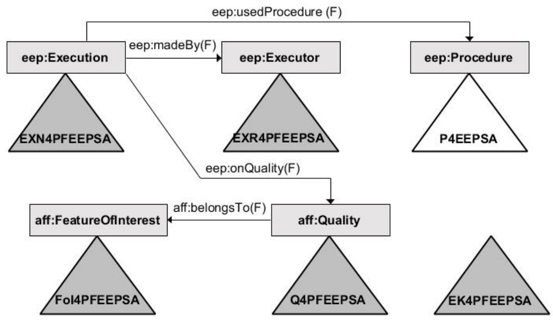

Fig. 7. An overview of the ontology modules replaced by the EEPSA ontology's customization for Poultry Farm domain.

An excerpt of the RDF model for the poultry farm at hand is available online[95].

The new PFEEPSA ontology and the proper customization of the rest of the assistant's components supported data analysts in the development of six different predictive models (one for each thermal zone in which the use case poultry farm is divided). Afterwards, this predictive models were used to forecast future indoor temperature, detect potential uncomfortable situations, and alert farmers so that proper actions could be taken in advance.

5. Conclusions

The proposed EEPSA ontology is a core ontology which aims at supporting a data analyst assistant in energy efficiency and thermal comfort problems in buildings. Towards the development of this ontology, a thorough review of existing ontologies related to the domains of discourse was made. Furthermore, a discussion of those reviewed ontologies led to emphasizing on the features that make ontologies more (re)usable.

Following a well-known ontology development methodology, the requirements of the ontology were collected first. Then, the backbone of the EEPSA ontology was discussed and defined as a combination of three ODPs which try to be minimal in the number of classes and properties offered but complete with respect to the considered CQs and including appropriate ontology axioms that allow proper inferences. Moreover, the careful design of property axioms overcome some weaknesses discovered in existing ontolo-

[95]https://raw.githubusercontent.com/iesnaola/pfeepsa/master/examples/poultryFarmExample.ttl

gies. On top of these ODPs, a set of ontology modules were developed, each of them specializing knowledge in the scope of the stub classes defined in the ODPs and reusing existing resources to the extent possible. Furthermore, in order to contribute to the interoperability of the solution, EEPSA ODPs and ontology modules are aligned with other related ontologies as well as upper-level ontologies. All these developments are properly documented, made available online, and evaluated from three different viewpoints.

Thanks to the design of the EEPSA ontology and the high encapsulation of its modules, the customization of the ontology to address similar problems in different use cases is feasible. Such a customization is based on ontology module replacement and it is tested in a real-world use case.

6. Acknowledgements

This work is partly supported by the project Internet of Food & Farms 2020 which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 731884. This work is also supported by PRE-2016-1-0303, IT1041-16-GBV and FEDER/TIN2016-78011-C4-2-R.

This work was conducted using the Protégé resource, which is supported by grant GM10331601 from the National Institute of General Medical Sciences of the United States National Institutes of Health.

References

[1] N.E. Klepeis, W.C. Nelson, W.R. Ott, J.P. Robinson, A.M. Tsang, P. Switzer, J.V. Behar, S.C. Hern and W.H. Engelmann, The National Human Activity Pattern Survey (NHAPS): a resource for assessing exposure to environmental pollutants, *Journal of Exposure Science and Environmental Epidemiology* **11**(3) (2001), 231.

[2] B.P. Haynes, The impact of office comfort on productivity, *Journal of Facilities Management* **6**(1) (2008), 37–51.

[3] A. Hedge and D.E. Gaygen, Indoor environment conditions and computer work in an office, *Hvac&R Research* **16**(2) (2010), 123–138.

[4] M. Mulville, N. Callaghan and D. Isaac, The impact of the ambient environment and building configuration on occupant productivity in open-plan commercial offices, *Journal of Corporate Real Estate* **18**(3) (2016), 180–193.

[5] K. Parsons, *Human Thermal Environments: The Effects of Hot, Moderate, and Cold Environments on Human Health, Comfort, and Performance*, 3rd edn, CRC Press, Inc., Boca Raton, FL, USA, 2014. ISBN ISBN 146659599X, 9781466595996.

[6] T. Abergel, B. Dean and J. Dulac, Towards a zero-emission, efficient, and resilient buildings and construction sector: GLobal Status Report, *UN Environment and International Energy Agency (2017)* (2017). ISBN 978-92-807-3686-1.

[7] S. Verbeke and A. Audenaert, Thermal inertia in buildings: A review of impacts across climate and building use, *Renewable and Sustainable Energy Reviews* **82** (2018), 2300–2318, ISSN 1364-0321. doi:10.1016/j.rser.2017.08.083.

[8] J. Gubbi, R. Buyya, S. Marusic and M. Palaniswami, Internet of Things (IoT): A vision, architectural elements, and future directions, *Future generation computer systems* **29**(7) (2013), 1645–1660. doi:10.1016/j.future.2013.01.010.

[9] U. Fayyad, G. Piatetsky-Shapiro and P. Smyth, From data mining to knowledge discovery in databases, *AI magazine* **17**(3) (1996), 37. doi:10.1609/aimag.v17i3.1230.

[10] A. Bernstein, F. Provost and S. Hill, Toward intelligent assistance for a data mining process: An ontology-based approach for cost-sensitive classification, *IEEE Transactions on Knowledge and Data Engineering* **17**(4) (2005), 503–518. doi:10.1109/TKDE.2005.67.

[11] N.F. Noy, Semantic integration: a survey of ontology-based approaches, *ACM Sigmod Record* **33**(4) (2004), 65–70. doi:10.1145/1041410.1041421.

[12] P. Andrews, I. Zaihrayeu and J. Pane, A classification of semantic annotation systems, *Semantic Web* **3**(3) (2012), 223–248. doi:10.3233/SW-2011-0056.

[13] S.-C. Yoon, L.J. Henschen, E.K. Park and S. Makki, Using Domain Knowledge in Knowledge Discovery, in: *Proceedings of the Eighth International Conference on Information and Knowledge Management*, CIKM '99, ACM, New York, NY, USA, 1999, pp. 243–250. ISBN ISBN 1-58113-146-1. doi:10.1145/319950.320008.

[14] I. Kopanas, N.M. Avouris and S. Daskalaki, The Role of Domain Knowledge in a Large Scale Data Mining Project, in: *Methods and Applications of Artificial Intelligence*, I.P. Vlahavas and C.D. Spyropoulos, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, pp. 288–299. ISBN ISBN 978-3-540-46014-5. doi:10.1007/3-540-46014-4_26.

[15] F.M. Pinto and M.F. Santos, Considering Application Domain Ontologies for Data Mining, *WSEAS Trans. Info. Sci. and App.* **6**(9) (2009), 1478–1492, ISSN 1790-0832.

[16] C.M. Eastman, C. Eastman, P. Teicholz, R. Sacks and K. Liston, *BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors*, John Wiley & Sons, 2011. doi:10.6028/NIST.IR.7908.

[17] P. Pauwels, S. Zhang and Y.-C. Lee, Semantic web technologies in AEC industry: A literature overview, *Automation in Construction* (2016). doi:10.1016/j.autcon.2016.10.003.

[18] P. Pauwels and A. Roxin, SimpleBIM: From full ifcOWL graphs to simplified building graphs, in: *eWork and eBusiness in Architecture, Engineering and Construction: ECPPM 2016: Proceedings of the 11th European Conference on Product and Process Modelling (ECPPM 2016), Limassol, Cyprus, 7-9 September 2016*, S. Christodoulou and R. Scherer, eds, CRC Press, 2017, pp. 11–18.

[19] D. Oberle, How ontologies benefit enterprise applications **5**(6) (2014), 473–491. doi:10.3233/SW-130114.

[20] P. Pauwels and W. Terkaj, EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology, *Automation in Construction* **63** (2016), 100–133. doi:10.1016/j.autcon.2015.12.003.

[21] T.M. de Farias, A. Roxin and C. Nicolle, IfcWoD, semantically adapting IFC model relations into OWL properties, *Proceedings of the 32nd CIB W78 Conference on Information Technology in Construction* (2015).

[22] M. Venugopal, C.M. Eastman and J. Teizer, An ontology-based analysis of the industry foundation class schema for building information model exchanges, *Advanced Engineering Informatics* **29**(4) (2015), 940–957, Collective Intelligence Modeling, Analysis, and Synthesis for Innovative Engineering Decision Making Special Issue of the 1st International Conference on Civil and Building Engineering Informatics, ISSN 1474-0346. doi:10.1016/j.aei.2015.09.006.

[23] S. Borgo, E.M. Sanfilippo, A. Šojić and W. Terkaj, Ontological Analysis and Engineering Standards: An Initial Study of IFC, in: *Ontology Modeling in Physical Asset Integrity Management*, V. Ebrahimipour and S. Yacout, eds, Springer International Publishing, Cham, 2015, pp. 17–43. ISBN ISBN 978-3-319-15326-1. doi:10.1007/978-3-319-15326-1_2.

[24] M. Poveda-Villalón and R. García-Castro, Extending the SAREF ontology for building devices and topology, in: *Proceedings of the 6th Linked Data in Architecture and Construction Workshop (LDAC 2018)*, Vol. CEUR-WS 2159, 2018, pp. 16–23.

[25] D. Bonino and F. Corno, Dogont-ontology modeling for intelligent domotic environments, in: *International Semantic Web Conference*, Springer, 2008, pp. 790–803. doi:10.1007/978-3-540-88564-1_51.

[26] D. Bonino and L.D. Russis, DogOnt as a viable seed for semantic modeling of AEC/FM, *Semantic Web* **9**(6) (2018), 763–780. doi:10.3233/SW-180295.

[27] J.J.V. Díaz, M.R. Wilby, A.B.R. González and J.G.M. noz, EEOnt: An ontological model for a unified representation of energy efficiency in buildings, *Energy and Buildings* **60** (2013), 20–27, ISSN 0378-7788. doi:10.1016/j.enbuild.2013.01.012.

[28] D. Bonino, F. Corno and L.D. Russis, PowerOnt: An Ontology-Based Approach for Power Consumption Estimation in Smart Homes, in: *Internet of Things. User-Centric IoT*, R. Giaffreda, R.-L. Vieriu, E. Pasher, G. Bendersky, A.J. Jara, J.J.P.C. Rodrigues, E. Dekel and B. Mandler, eds, Springer International Publishing, Cham, 2015, pp. 3–8. ISBN ISBN 978-3-319-19656-5. doi:10.1007/978-3-319-19656-5_1.

[29] C. Reinisch, M. Kofler, F. Iglesias and W. Kastner, ThinkHome Energy Efficiency in Future Smart Homes, *EURASIP Journal on Embedded Systems* **2011** (2010), 1–1118, ISSN 1687-3955. doi:10.1155/2011/104617.

[30] M.H. Rasmussen, P. Pauwels, C.A. Hviid and J. Karlshøj, Proposing a Central AEC Ontology That Allows for Domain Specific Extensions, in: *Joint Conference on Computing in Construction*, Vol. 1, 2017, pp. 237–244. doi:10.24928/JC3-2017/0153..

[31] M.H. Rasmussen, P. Pauwels, M. Lefrançois, G. Schneider, C. Hviid and J. Karlshøj, Recent changes in the Building Topology Ontology, in: *Proceedings of the 5th Linked Data in Architecture and Construction Workshop (LDAC 2017)*, 2017. doi:10.13140/RG.2.2.32365.28647.

[32] M.H. Rasmussen, M. Lefrançois, M. Bonduel, C.A. Hviid and J. Karlshøj, OPM: An ontology for describing properties that evolve over time, in: *Proceedings of the 6th Linked Data in Architecture and Construction Workshop (LDAC 2018)*, Vol. CEUR-WS 2159, 2017, pp. 24–33.

[33] G. Schneider, Towards Aligning Domain Ontologies with the Building Topology Ontology, in: *Proceedings of the 5th Linked Data in Architecture and Construction Workshop (LDAC 2017)*, 2017. doi:10.13140/RG.2.2.21802.52169.

[34] B. Balaji, A. Bhattacharya, G. Fierro, J. Gao, J. Gluck, D. Hong, A. Johansen, J. Koh, J. Ploennigs, Y. Agarwal, M. Berges, D. Culler, R. Gupta, M.B. Kjærgaard, M. Srivastava and K. Whitehouse, Brick: Towards a Unified Metadata Schema For Buildings, in: *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments*, BuildSys '16, ACM, New York, NY, USA, 2016, pp. 41–50. ISBN ISBN 978-1-4503-4264-3. doi:10.1145/2993422.2993577.

[35] J.M. Hook, E. Pan, J. Adler-Milstein, D. Bu and J. Walker, The Value of Healthcare Information Exchange and Interoperability in New York State, in: *AMIA Annual Symposium Proceedings*, Vol. 2006, American Medical Informatics Association, 2006, p. 953, ISSN 1942-597X.

[36] S.N.A.U. Nambi, C. Sarkar, R.V. Prasad and A. Rahim, A unified semantic knowledge base for IoT, in: *2014 IEEE World Forum on Internet of Things (WF-IoT)*, 2014, pp. 575–580. doi:10.1109/WF-IoT.2014.6803232.

[37] Y. Liao, M. Lezoche, H. Panetto, N. Boudjlida and E.R. Loures, Formal Semantic Annotations for Models Interoperability in a PLM environment, *IFAC Proceedings Volumes* **47**(3) (2014), 2382–2393, 19th IFAC World Congress, ISSN 1474-6670. doi:10.3182/20140824-6-ZA-1003.02551.

[38] Y. Liao, M. Lezoche, H. Panetto and N. Boudjlida, Semantic annotations for semantic interoperability in a product lifecycle management context, *International Journal of Production Research* **54**(18) (2016), 5534–5553.

[39] Y. Lin and H. Ding, Ontology-based Semantic Annotation for Semantic Interoperability of Process Models, in: *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, Vol. 1, 2006, pp. 162–167. doi:10.1109/CIMCA.2005.1631259.

[40] H.N. Talantikite, D. Aissani and N. Boudjlida, Semantic annotations for web services discovery and composition, *Computer Standards & Interfaces* **31**(6) (2009), 1108–1117, ISSN 0920-5489. doi:10.1016/j.csi.2008.09.041.

[41] A. Gyrard, C. Bonnet, K. Boudaoud and M. Serrano, LOV4IoT: A second life for ontology-based domain knowledge to build Semantic Web of Things applications, in: *2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)*, IEEE, 2016, pp. 254–261.

[42] G. Bajaj, R. Agarwal, P. Singh, N. Georgantas and V. Issarny, A study of existing Ontologies in the IoT-domain, *arXiv preprint arXiv:1707.00112* (2017).

[43] M. Compton, P. Barnaghi, L. Bermudez, R. García-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson and A. Herzog, The SSN ontology of the W3C semantic sensor network incubator group, *Web Semantics: Science, Services and Agents on the World Wide Web* **17** (2012), 25–32. doi:10.1016/j.websem.2012.05.003.

[44] K. Janowicz and M. Compton, The Stimulus-Sensor-Observation Ontology Design Pattern and its Integration into the Semantic Sensor Network Ontology., K. Taylor, A. Ayyagari and D.D. Roure, eds, 2010, ISSN 1613-0073. `http://ceur-ws.org/Vol-668/paper12.pdf`.

[45] A. Haller, K. Janowicz, S. Cox, M. Lefrançois, K. Taylor,

D.L. Phuoc, J. Lieberman, R. García-Castro, R. Atkinson and C. Stadler, The modular SSN ontology: A joint W3C and OGC standard specifying the semantics of sensors, observations, sampling, and actuation, *Semantic Web* **To be published** (2018). doi:10.3233/SW-180320.

[46] S. Cox, Ontology for observations and sampling features, with alignments to existing models, *Semantic Web* **8**(3) (2016), 453–470. doi:10.3233/SW-160214.

[47] L. Daniele, F. den Hartog and J. Roes, Created in close interaction with the industry: the smart appliances reference (SAREF) ontology, in: *International Workshop Formal Ontologies Meet Industries*, Springer, 2015, pp. 100–112. doi:10.1007/978-3-319-21545-7_9.

[48] M. Lefrançois, Planned ETSI SAREF Extensions based on the W3C&OGC SOSA/SSN-compatible SEAS Ontology Patterns, in: *Proceedings of Workshop on Semantic Interoperability and Standardization in the IoT, SIS-IoT,*, 2017.

[49] N. Seydoux, K. Drira, N. Hernandez and T. Monteil, IoT-O, a Core-Domain IoT Ontology to Represent Connected Devices Networks, in: *Knowledge Engineering and Knowledge Management: 20th International Conference, EKAW 2016, Bologna, Italy, November 19-23, 2016, Proceedings 20*, Vol. 10024, Springer, 2016, pp. 561–576. doi:10.1007/978-3-319-49004-5_36.

[50] R. Agarwal, D.G. Fernandez, T. Elsaleh, A. Gyrard, J. Lanza, L. Sanchez, N. Georgantas and V. Issarny, Unified IoT Ontology to Enable Interoperability and Federation of Testbeds, in: *3rd IEEE World Forum on Internet of Things*, 2016. doi:10.1109/WF-IoT.2016.7845470.

[51] M. Bermudez-Edo, T. Elsaleh, P. Barnaghi and K. Taylor, IoT-Lite: A Lightweight Semantic Model for the Internet of Things and its use with dynamic semantics, *Personal and Ubiquitous Computing* **21**(3) (2017), 475–487, ISSN 1617-4909. doi:10.1007/s00779-017-1010-8.

[52] A.G.S.K. Datta, C. Bonnet and K. Boudaoud, Cross-Domain Internet of Things Application Development: M3 Framework and Evaluation, in: *2015 3rd International Conference on Future Internet of Things and Cloud*, IEEE, 2015, pp. 9–16. doi:10.1109/FiCloud.2015.10.

[53] M. Alirezaie, K. Hammar and E. Blomqvist, SmartEnv as a Network of Ontology Patterns, *Semantic Web* (2018), 1–16. http://www.semantic-web-journal.net/.

[54] S. Sagar, M. Lefrançois, I. Rebaï, M. Khemaja, S. Garlatti, J. Feki and L. Médini, Modeling Smart Sensors on top of SOSA/SSN and WoT TD with the Semantic Smart Sensor Network (S3N) modular Ontology, in: *9th International Semantic Sensor Networks Workshop*, 2018.

[55] T. Lebo, S. Sahoo and D. McGuinness, PROV-O: The PROV Ontology, W3C Recommendation, W3C, 2013, http://www.w3.org/TR/2013/REC-prov-o-20130430/.

[56] R. Fikes and Q. Zhou, A Reusable Time Ontology, in: *Proceeding of the AAAI Workshop on Ontologies for the Semantic Web*, 2002.

[57] J. Hobbs and J. Pustejovsky, Annotating and reasoning about time and events, in: *Proceedings of AAAI Spring Symposium on Logical Formalizations of Commonsense Reasoning*, Vol. 3, 2003.

[58] M.J. O'Connor and A.K. Das, A Method for Representing and Querying Temporal Information in OWL, in: *Biomedical Engineering Systems and Technologies*, A. Fred, J. Filipe and H. Gamboa, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 97–110. ISBN ISBN 978-3-642-18472-7. doi:10.1007/978-3-642-18472-7_8.

[59] C. Zhang, C. Cao, Y. Sui and X. Wu, A Chinese time ontology for the Semantic Web, *Knowledge-Based Systems* **24**(7) (2011), 1057–1074, ISSN 0950-7051. doi:10.1016/j.knosys.2011.04.021.

[60] A. Galton, The Treatment of Time in Upper Ontologies, in: *Formal Ontology in Information Systems: Proceedings of the 10th International Conference (FOIS 2018)*, Vol. 306, IOS Press, 2018, pp. 33–46. doi:10.3233/978-1-61499-910-2-33.

[61] S. Cox and C. Little, Time Ontology in OWL, W3C Recommendation, W3C, 2017, https://www.w3.org/TR/2017/REC-owl-time-20171019/.

[62] T. Flury, G. Privat and F. Ramparany, OWL-based location ontology for context-aware services, *Proceedings of the Artificial Intelligence in Mobile Systems (AIMS 2004)* (2004), 52–57.

[63] M. Perry and J. Herring, OGC GeoSPARQL-A geographic query language for RDF data, *OGC implementation standard* (2012).

[64] J.M. Keil and S. Schindler, Comparison and evaluation of ontologies for units of measurement, *Semantic Web* (2018), 1–19.

[65] M. Lefrançois and A. Zimmermann, The Unified Code for Units of Measure in RDF: cdt:ucum and other UCUM Datatypes, in: *The Semantic Web: ESWC 2018 Satellite Events*, A. Gangemi, A.L. Gentile, A.G. Nuzzolese, S. Rudolph, M. Maleshkova, H. Paulheim, J.Z. Pan and M. Alam, eds, Springer International Publishing, 2018, pp. 196–201. ISBN ISBN 978-3-319-98192-.

[66] Y. Sure, S. Staab and R. Studer, On-To-Knowledge Methodology (OTKM), in: *Handbook on Ontologies*, S. Staab and R. Studer, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 117–132. ISBN ISBN 978-3-540-24750-0. doi:10.1007/978-3-540-24750-0_6.

[67] H.S. Pinto, S. Staab and C. Tempich, DILIGENT: Towards a fine-grained methodology for DIstributed, Loosely-controlled and evolvInG Engineering of oNTologies, in: *In Proceedings of the 16th European Conference on Artificial Intelligence (ECAI*, IOS Press, 2004, pp. 393–397.

[68] M.C. Suárez-Figueroa, A. Gómez-Pérez and M. Fernández-López, The NeOn Methodology for Ontology Engineering, in: *Ontology Engineering in a Networked World*, M.C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta and A. Gangemi, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 9–34. ISBN ISBN 978-3-642-24794-1. doi:10.1007/978-3-642-24794-1_2.

[69] M.C. Suárez-Figueroa and A. Gómez-Pérez, Ontology Requirements Specification, in: *Ontology Engineering in a Networked World*, M.C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta and A. Gangemi, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 93–106. ISBN ISBN 978-3-642-24794-1. doi:10.1007/978-3-642-24794-1_5.

[70] M.A. Musen, The protégé project: a look back and a look forward, *AI matters* **1**(4) (2015), 4–12.

[71] E. Simperl, Reusing ontologies on the Semantic Web: A feasibility study, *Data & Knowledge Engineering* **68**(10) (2009), 905–925.

[72] A. Gangemi and V. Presutti, Ontology Design Patterns, in: *Handbook on Ontologies*, S. Staab and R. Studer, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 221–243. ISBN ISBN 978-3-540-92673-3. doi:10.1007/978-3-540-

92673-3_10.
[73] P. Hitzler, A. Gangemi and K. Janowicz, *Ontology Engineering with Ontology Design Patterns: Foundations and Applications*, Vol. 25, IOS Press, 2016.
[74] I. Esnaola-Gonzalez, J. Bermúdez, I. Fernandez and A. Arnaiz, Two Ontology Design Patterns toward Energy Efficiency in Buildings, in: *Proceedings of the 9th Workshop on Ontology Design and Patterns (WOP 2018) co-located with 17th International Semantic Web Conference (ISWC 2018)*, Vol. 2195, CEUR, 2018, pp. 14–28.
[75] A. Gangemi, R. Lillo, G. Lodi and A.G. Nuzzolese, A pattern-based ontology for the Internet of Things, *Proceedings of the 8th Workshop on Ontology Design and Patterns (WOP 2017)* **2043** (2017), ISSN 1613-0073. `http://ceur-ws.org/Vol-2043/paper-11.pdf`.
[76] B.C. Grau, I. Horrocks, Y. Kazakov and U. Sattler, Modular reuse of ontologies: Theory and practice, *Journal of Artificial Intelligence Research* **31** (2008), 273–318. doi:10.1613/jair.2375.
[77] H. Stuckenschmidt and M. Klein, Reasoning and change management in modular ontologies, *Data & Knowledge Engineering* **63**(2) (2007), 200–223, ISSN 0169-023X. doi:10.1016/j.datak.2007.02.001.
[78] F. Ensan and W. Du, A Semantic Metrics Suite for Evaluating Modular Ontologies, *Inf. Syst.* **38**(5) (2013), 745–770, ISSN 0306-4379. doi:10.1016/j.is.2012.11.012.
[79] M. d'Aquin, A. Schlicht, H. Stuckenschmidt and M. Sabou, Criteria and Evaluation for Ontology Modularization Techniques, in: *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, H. Stuckenschmidt, C. Parent and S. Spaccapietra, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 67–89. ISBN ISBN 978-3-642-01907-4. doi:10.1007/978-3-642-01907-4_4.
[80] H. Qiua, G. Schneider, T. Kauppinen, S. Rudolph and S. Steigerd, Reasoning on Human Experiences of Indoor Environments using Semantic Web Technologies, in: *Proceedings of the 35th International Symposium on Automation and Robotics in Construction (ISARC 2018), Berlin, Germany*, 2018.
[81] S. Peroni, D. Shotton and F. Vitali, Tools for the Automatic Generation of Ontology Documentation: A Task-Based Evaluation, *Int. J. Semant. Web Inf. Syst.* **9**(1) (2013), 21–44, ISSN 1552-6283. doi:10.4018/jswis.2013010102.
[82] D. Garijo, WIDOCO: A Wizard for Documenting Ontologies, in: *The Semantic Web – ISWC 2017*, C. d'Amato, M. Fernandez, V. Tamma, F. Lecue, P. Cudré-Mauroux, J. Sequeda, C. Lange and J. Heflin, eds, Springer International Publishing, Cham, 2017, pp. 94–102. ISBN ISBN 978-3-319-68204-4.
[83] N. Calegari, C. Burle and B.F. Loscio, Data on the Web Best Practices, W3C Recommendation, W3C, 2017, https://www.w3.org/TR/2017/REC-dwbp-20170131/.
[84] D. Garijo and M. Poveda-Villalón, A checklist for complete vocabulary metadata, Technical Report, 2017. `https://w3id.org/widoco/bestPractices`.
[85] J. Brank, M. Grobelnik and D. Mladenić, A Survey of Ontology Evaluation Techniques, in: *Proc. of 8th Int. multi-conf. Information Society*, 2005, pp. 166–169.
[86] L. Obrst, W. Ceusters, I. Mani, S. Ray and B. Smith, The Evaluation of Ontologies, in: *Semantic Web: Revolutionizing Knowledge Discovery in the Life Sciences*, C.J.O. Baker and K.-H. Cheung, eds, Springer US, Boston, MA, 2007, pp. 139–158. ISBN ISBN 978-0-387-48438-9. doi:10.1007/978-0-387-48438-9_8.
[87] D. Vrandečić and Y. Sure, How to Design Better Ontology Metrics, in: *The Semantic Web: Research and Applications*, E. Franconi, M. Kifer and W. May, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 311–325. ISBN ISBN 978-3-540-72667-8.
[88] M. Poveda-Villalón, A. Gómez-Pérez and M.C. Suárez-Figueroa, OOPS! (OntOlogy Pitfall Scanner!): An Online Tool for Ontology Evaluation, *Int. J. Semant. Web Inf. Syst.* **10**(2) (2014), 7–34, ISSN 1552-6283. doi:10.4018/ijswis.2014040102.
[89] Z.C. Khan and C.M. Keet, Dependencies Between Modularity Metrics Towards Improved Modules, in: *Knowledge Engineering and Knowledge Management*, E. Blomqvist, P. Ciancarini, F. Poggi and F. Vitali, eds, Springer International Publishing, Cham, 2016, pp. 400–415. ISBN ISBN 978-3-319-49004-5.
[90] C. Jonquet, A. Toulet, E. Arnaud, S. Aubin, E.D. Yeumo, V. Emonet, J. Graybeal, M.-A. Laporte, M.A. Musen, V. Pesce et al., AgroPortal: A vocabulary and ontology repository for agronomy, *Computers and Electronics in Agriculture* **144** (2018), 126–143.
[91] P. Jaiswal, L. Cooper, J. Elser, A. Meier, M.-A. Laporte, C. Mungall, B. Smith, E. Johnson, M. Seymour, J. Preece et al., Planteome: a resource for common reference ontologies and applications for plant biology, in: *XXIV Plant and Animal Genome Conference*, 2016, pp. 9–13.
[92] C. Snae and M. Bruckner, FOODS: a food-oriented ontology-driven system, in: *2nd IEEE International Conference on Digital Ecosystems and Technologies*, IEEE, 2008, pp. 168–176.
[93] N. Bansal and S.K. Malik, A framework for agriculture ontology development in semantic web, in: *Communication Systems and Network Technologies (CSNT), 2011 International Conference on*, IEEE, 2011, pp. 283–286.
[94] T. Pizzuti, G. Mirabelli, M.A. Sanz-Bobi and F. Goméz-Gonzaléz, Food Track & Trace ontology for helping the food traceability control, *Journal of Food Engineering* **120** (2014), 17–30.

Appendix

A. Application Examples of the ODPs

This appendix shows an example of the presented three ODPs.

A.1. AffectedBy ODP Example

Figure 8 shows a triple graph as an example for applying and answering some competency questions using the AffectedBy vocabulary.

With respect to this example, the following competency questions can be applied and answered:

- (CQ01): What are the properties that influence the feature of interest *:room03*?

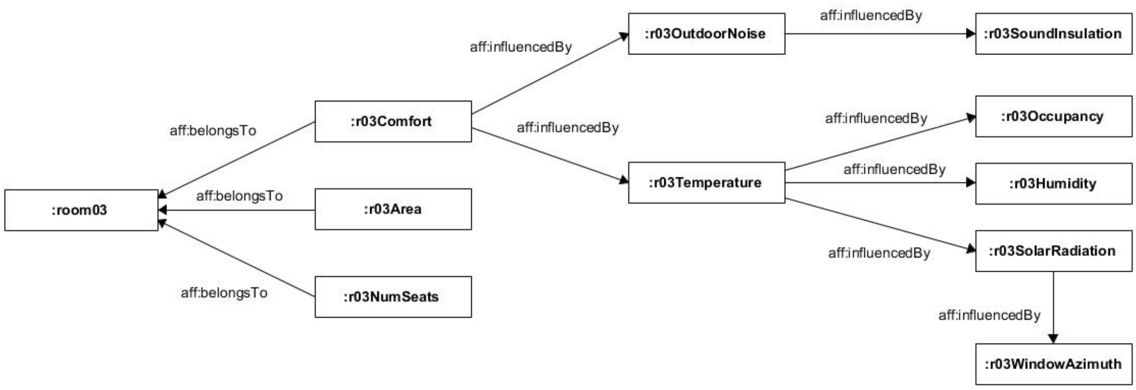

Fig. 8. Triples using the AffectedBy ODP vocabulary.

SELECT *?x*
WHERE {*:room03 aff:influencedBy ?x.*}
Answer: *:r03Area*, *:r03NumSeats :r03Comfort*, *:r03Temperature*, *:r03OutdoorNoise*, *:r03Occupancy*, *:r03Humidity*, *:r03SolarRadiation*, *:r03-SoundInsulation*, *:r03WindowAzimuth*.
(After inferences provided by axioms *aff:influencedBy* ∘ *aff:affectedBy* ⊑ *aff:influencedBy* and *aff:belongsTo* ⊑ *aff:influencedBy*$^{-1}$).

– (CQ01[i]): Which is the feature of interest influenced by the property *:r03SolarRadiation*
SELECT *?x*
WHERE {*?x aff:influencedBy ?r03SolarRadiation.*}
Answer: *:room03*.
(After inferences provided by the axioms *aff:influencedBy* ∘ *aff:affectedBy* ⊑ *aff:influencedBy* and *aff:belongsTo* ⊑ *aff:influencedBy*$^{-1}$).

– (CQ02): What are the properties that affect the property *:r03Temperature*?
SELECT *?x*
WHERE { *:r03Temperature aff:affectedBy ?x.*}
Answer: *:r03Occupancy*, *:r03Humidity*, *:r03SolarRadiation*, *:r03WindowAzimuth*.
(After inferences provided by the transitivity of *aff:affectedBy*).

– (CQ03): Which feature of interest does the property *:r03Area* belongs to?
SELECT *?x*
WHERE {*:r03Area aff:belongsTo ?x.*}
Answer: *:room03*.

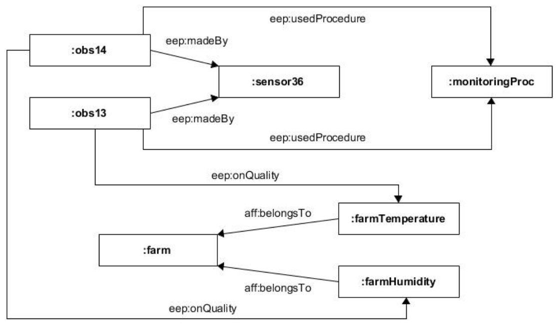

Fig. 9. Triples using the EEP ODP vocabulary.

A.2. EEP ODP Example

Figure 9 shows an instantiation of the EEP ODP in a farm scenario where poultry are reared. In this case, a sensor *:sensor36* deployed in the farm individual *:farm* is in charge of measuring both farm's temperature and humidity (i.e. *:farmTemperature* and *:farmHumidity*). Furthermore, this sensor implements a monitoring procedure (*:monitoringProc*) to make two observations *:obs13* and *:obs14*.

With respect to this example, the following competency questions can be applied and answered:

– (CQ04): What are the executions performed by procedure *:monitoringProc*?
SELECT *?x*
WHERE {*?x eep:usedProcedure :monitoringProc.*}
Answer: *:obs13*, *:obs14*.

– (CQ05): What are the observations performed by sensor *:sensor36*?
SELECT *?x*

WHERE {*?x eep:madeBy :sensor36.*}
Answer: *:obs13*, *:obs14*.

- (CQ06): Which are the procedures implemented by the sensor *:sensor36*?
SELECT *?x*
WHERE {*:sensor36 eep:implements ?x.*}
Answer: *:monitoringProc*
(After inferences provided by the axiom *eep:madeBy^{-1}* ∘ *eep:usedProcedure* ⊑ *eep:implements*).

- (CQ07[i]): What are the executions on the feature of interest *:farm*?
SELECT *?x*
WHERE {*?x eep:hasFeatureOfInterest :farm.*}
Answer: *:obs13*, *:obs14*.
(After inferences provided by the axiom *eep:onQuality* ∘ *eep:belongsTo* ⊑ *eep:hasFeatureOfInterest*).

- (CQ08): What are the qualities observed by the observation *:obs13*?
SELECT *?x*
WHERE {*:obs13 eep:onQuality ?x.*}
Answer: *:farmTemperature*.

- (CQ09[i]): What are the executors that observe/act on the feature of interest *:farm*?
SELECT *?x*
WHERE {*?x eep:forFeatureOfInterest :farm.*}
Answer: *:sensor36*.
(After inferences provided by the axioms *eep:forQuality* ∘ *eep:belongsTo* ⊑ *eep:forFeatureOfInterest* and *eep:madeBy^{-1}* ∘ *eep:onQuality* ⊑ *eep:forQuality*).

- (CQ10): What are the qualities observed by sensor *:sensor36*?
SELECT *?x*
WHERE {*:sensor36 eep:forQuality ?x.*}
Answer: *:farmTemperature*, *:farmHumidity*.
(After inferences provided by the axiom *eep:madeBy^{-1}* ∘ *eep:onQuality* ⊑ *eep:forQuality*).

A.3. RC ODP Example

The RC ODP is instantiated in a weather forecast report. In this case, an execution *:forecastReport* is generated on 2018-11-02 at 11:00 (with the data property *rc:hasGenerationTime*) and forecasts that there will be a temperature of 16°C (with the data property *rc:hasSimpleResult*) in Madrid (with the object property *rc:hasSpatialContext*) on 2018-11-03 at 16:00 (with the data property *rc:hasTemporalContext*). Figure 10 shows this instantiation example.

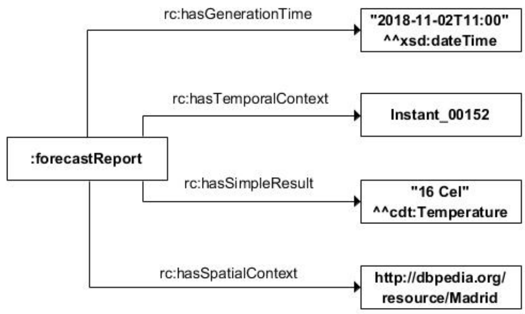

Fig. 10. Triples using the RC ODP vocabulary.

With respect to this example, the following competency questions can be applied and answered:

- (CQ11): Which is the simplified value of execution *:forecastReport*?
SELECT *?x*
WHERE {*forecastReport ec:hasSimpleResult ?x.*}
Answer: *"16 Cel"^^cdt:temperature*.

- (CQ12): When is the execution *:forecastReport* generated?
SELECT *?x*
WHERE {*:forecastReport ec:hasGenerationTime ?x.*}
Answer: *"2018-11-02T11:00:00"^^xsd:dateTime*.

- (CQ13): For what time interval or instant is valid the execution *:forecastReport*?
SELECT *?x*
WHERE {*:forecastReport ec:hasTemporalContext ?x.*}
Answer: *:Instant_00152*.

- (CQ14): For what spatial location is valid the execution *:forecastReport*?
SELECT *?x*
WHERE {*:forecastReport ec:hasSpatialContext ?x.*}
Answer: *http://dbpedia.org/resource/Madrid*.