

eman ta zabal zazu



Universidad del País Vasco      Euskal Herriko Unibertsitatea

# Semantic Technologies for supporting KDD Processes

**Iker Esnaola Gonzalez**

Supervised by:

**Jesús Bermúdez de Andrés**  
**Izaskun Fernández González**

A dissertation submitted to the Department of Computer Languages and Systems of the University of the Basque Country UPV/EHU for the degree of Doctor of Philosophy (Ph.D.) in Informatics Engineering

Donostia-San Sebastián, February 2019.



*A los que han confiado en mí*



# Acknowledgements

I would like to start this dissertation expressing my gratitude to people who were very important throughout this PhD thesis.

Firstly, I would like to thank my director Jesús for his constant help, advices and involvement in this thesis. I will miss the endless conversations we had, thanks to which I learnt believing in my abilities and having trust in the hard work. I would also like to thank my director Izaskun and my former director Aitor for their support and the trust they put in me when I needed most.

I thank IK4-Tekniker for giving me the opportunity of doing this PhD thesis in such a supportive environment. Furthermore, I also want to thank my colleagues for the lessons learnt from working with them.

Last but not least, I would like to thank my family and friends for dealing with my ups and downs through this process, and being always supportive, no matter what was going on. I wouldn't have got where I am today without them.



# Summary

Achieving a comfortable thermal situation with an efficient use of energy remains still an open challenge for most buildings. In this regard, the advent of the IoT (Internet of Things) and maturity of KDD (Knowledge Discovery in Databases) processes may contribute to the solution of these problems. However, the adequate combination of these two technologies is not straightforward, due to the heterogeneity and volume of the data to be considered. Therefore, data analysts could benefit from an application assistant that supports them throughout the KDD process.

This research work aims at supporting data analysts through the different KDD phases towards the achievement of energy efficiency and thermal comfort in tertiary buildings. To do so, the EEPsA (Energy Efficiency Prediction Semantic Assistant) is proposed, which aids data analysts discovering the most relevant variables for the matter at hand, and informing them about relationships among relevant data.

EEPSA leverages Semantic Technologies such as ontologies, ontology-driven rules and ontology-driven data access. More specifically, the EEPsA ontology is the cornerstone of the assistant. 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; 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.





# Resumen

Conseguir una situación térmica confortable con un uso de energía eficiente sigue siendo un desafío para la mayoría de edificios. La llegada del IoT (Internet of Things) y la madurez de los procesos KDD (Knowledge Discovery in Databases) pueden contribuir para solucionar este tipo de problemas. Sin embargo, la combinación de estas tecnologías no es directa, debido a la heterogeneidad y el volumen de los datos a considerar. En estos casos, los analistas de datos podrían beneficiarse de una aplicación de asistencia que les diera soporte a lo largo del proceso KDD.

Este trabajo de investigación pretende dar soporte a los analistas de datos a lo largo de las distintas fases del KDD, con miras a conseguir la eficiencia energética y el confort térmico en edificios terciarios. Para ello, se propone el EEP SA (Energy Efficiency Prediction Semantic Assistant), que pretende ayudar a los analistas de datos a descubrir las variables más relevantes del problema en cuestión, e informarlos acerca de las relaciones existentes entre estos datos.

EEPSA hace uso de Tecnologías Semánticas como ontologías, reglas basadas en ontologías, y acceso a datos basado en ontologías. Más concretamente, la ontología EEP SA es el pilar de dicho asistente. Esta ontología está desarrollada basándose en tres ODPs (Ontology Design Patterns), que abordan las debilidades identificadas en las propuestas existentes para representar: características de interés y sus respectivas cualidades; observaciones y actuaciones; los sensores y actuadores que las generan; y los procedimientos utilizados. La ontología está diseñada para que su customización para abordar problemas similares en distintos tipos de edificios pueda realizarse metódicamente.



# Laburpena

Erosotasun termikoa eta aldi berean energiaren erabilera eraginkor bat bermatzea, erronka bat da gaur egun eraikin gehienetan. IoT-aren (Internet of Things) iritsiera eta KDD (Knowledge Discovery in Databases) prozesuen heldutasunak arazo honi konponbide bat bilatzen lagundu dezakete. Hala ere, teknologia hauen konbinazioa ez da erraza, batez ere datuen heterogeneitate eta bolumenaren ondorioz. Kasu hauetan, datu analistek, KDD prozesu hauetan zehar laguntza eskaintzen duen asistentzia aplikazio batez baliatu litezke.

Ikerketa lan honetan, datu analistak KDD fase desberdinen zehar lagundu nahi dira, eraikin tertziarioetan efizientzia energetikoa eta erosotasun termikoa lortzeko helburuarekin. Horretarako, EEP SA (Energy Efficiency Prediction Semantic Assistant) proposatzen da. Honek, datu analistei laguntza ematen die, alde batetik, arazoaren aldagai garrantzitsuenak aurkitzen, eta bestetik, datuen arteko erlazioei buruzko informazioa emanez.

EEPSA-k Teknologia Semantikoak erabiltzen ditu, hala nola ontologiak, ontologietan oinarritutako arauak, eta ontologietan oinarritutako datuen atzipena. Zehazki, proposatutako EEP SA-ren oinarria EEP SA ontologia da zein, era berean, 3 ODP-tan (Ontology Design Patterns) oinarritzen den. ODP hauek, gaur egun existitzen diren proposamenen ahuleziak konpontzen saiatzen dira. Zehazki, ondorengo kontzeptuak adierazten dituzten proposamenak: ezaugarri interesgarriak eta beraiek nolakotasunak; behaketa eta aktuazioak; hauek sortzen dituzten sentore eta aktuatzailak; eta erabilitako prozedurak. Ontologiaren diseinuak bere aldaketa era metodiko batean egitea ahalbidetzen du, era horretan datu analistak beste eraikin mota batzuetako arazo berdintsuetan laguntzeko.



# Contents

<b>List of Figures</b>	<b>xv</b>
<b>List of Tables</b>	<b>xix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Thesis objectives and contributions . . . . .	4
1.2 Thesis structure . . . . .	5
1.3 Published work . . . . .	5
<b>2 Fundamental Technologies</b>	<b>7</b>
2.1 KDD . . . . .	7
2.2 The Semantic Web and Semantic Technologies . . . . .	8
<b>3 KDD with Semantic Technologies: Related Work</b>	<b>17</b>
3.1 Semantic Technologies in Annotation . . . . .	17
3.2 Semantic Technologies in Data Selection . . . . .	35
3.3 Semantic Technologies in Preprocessing . . . . .	36
3.4 Semantic Technologies in Transformation . . . . .	40
3.5 Semantic Technologies in Data Mining . . . . .	41
3.6 Semantic Technologies in Interpretation . . . . .	41
<b>4 The EEP SA Ontology</b>	<b>45</b>
4.1 Ontology Development Methodology . . . . .	45

4.2	The EEP SA Ontology Scope . . . . .	48
4.3	Developing the EEP SA Ontology on top of ODPs . . . . .	48
4.4	Ontology Reuse Discussion . . . . .	62
4.5	The EEP SA Ontology Modules . . . . .	64
4.6	The EEP SA Ontology Customization . . . . .	74
4.7	Documentation . . . . .	76
4.8	Ontology Evaluation . . . . .	79
4.9	Ontology Versioning . . . . .	83
<b>5</b>	<b>The EEP SA</b>	<b>87</b>
5.1	Semantic Annotation . . . . .	87
5.2	Data Selection . . . . .	88
5.3	Preprocessing . . . . .	91
5.4	Transformation . . . . .	99
5.5	Data Mining . . . . .	103
5.6	Interpretation . . . . .	104
<b>6</b>	<b>The EEP SA in an Office</b>	<b>111</b>
6.1	Experiments, Evaluation and Results . . . . .	112
<b>7</b>	<b>The EEP SA in a Poultry Farm</b>	<b>135</b>
7.1	Requirements . . . . .	137
7.2	The EEP SA Customization . . . . .	139
7.3	Experiments . . . . .	144
7.4	Evaluation and Results discussion . . . . .	146
7.5	The PFEEP SA in Production . . . . .	146
<b>8</b>	<b>Conclusions</b>	<b>149</b>
8.1	Contributions . . . . .	150

<i>Contents</i>	xiii
8.2 Future work . . . . .	153
<b>Bibliography</b>	<b>173</b>
<b>A List of Abbreviations</b>	<b>175</b>
<b>B Ontology Requirements Specification Document</b>	<b>179</b>
B.1 EEPISA ontology Requirements . . . . .	184
<b>C Evaluation of the Ontology</b>	<b>189</b>
C.1 Design Correctness Metrics . . . . .	189
C.2 Structural Metrics . . . . .	194
C.3 Ontology Module Quality Metrics . . . . .	203





# List of Figures

2.1	An overview of the steps that compose the KDD process proposed by Fayyad et al. . . . .	8
2.2	An RDF Triple example. . . . .	9
2.3	Venn diagram showing the relation between OWL 2 profiles. . . . .	13
2.4	The Ontology spectrum as defined by Lassila and McGuinness. . . . .	16
4.1	The AffectedBy ODP. . . . .	53
4.2	Triples using the AffectedBy ODP vocabulary. . . . .	54
4.3	A SOSA/SSN annotated set of triples. . . . .	56
4.4	The Execution-Executor-Procedure (EEP) ODP. . . . .	57
4.5	Triples using the EEP ODP vocabulary. . . . .	58
4.6	The Result-Context (RC) ODP. . . . .	60
4.7	Triples using the RC ODP vocabulary. . . . .	61
4.8	Overview of the main classes and properties defined in BOT. . . . .	65
4.9	Overview of the classes defined in FoI4EEPSA. . . . .	66
4.10	Overview of the classes defined in Q4EEPSA. . . . .	68
4.11	Overview of the classes defined in P4EEPSA. . . . .	69
4.12	Overview of the classes defined in EXR4EEPSA. . . . .	71
4.13	Overview of the classes defined in EXN4EEPSA. . . . .	73
4.14	Overview of the ontology modules replaced by the EEPSA ontology's customization for residential buildings domain. . . . .	76

4.15	An overview of relevant classes and properties in the EEPSA ontology version 1.2. . . . .	84
4.16	Overview of the Forecasting4eepsa ontology module. . . . .	85
4.17	Overview of Measurements4eepsa ontology module's extension. . .	86
5.1	EEPSA's ETL process for the Transformation phase. . . . .	100
5.2	Overview of the EROSO framework. . . . .	105
5.3	EROSO framework's interface. . . . .	109
6.1	IK4-Tekniker building's Open Space. . . . .	112
6.2	Overview of actual outliers measured by sensor T17 and their detection by different techniques. . . . .	119
6.3	Mean DTW distance between the original and the datasets with imputed values, for all the tested imputation methods and missing segments lengths. . . . .	123
6.4	Rapidminer process of the baseline predictive model. . . . .	131
6.5	OSCS's interface. . . . .	133
7.1	Use case poultry farm. . . . .	136
7.2	Optimal poultry farm temperatures through a breeding period. . .	137
7.3	Overview of the ontology modules replaced by the EEPSA ontology's customization for Poultry Farm domain. . . . .	141
7.4	Overview of the EEPSA's ETL process after its customization for AEMET weather stations. . . . .	142
7.5	Overview of the EROSO framework customization for the poultry farm domain. . . . .	144
7.6	Use case poultry farm's thermal zone division. . . . .	145
8.1	Summary of the major contributions of this thesis. . . . .	150
C.1	Design correctness metrics for the AffectedBy ODP. . . . .	189
C.2	Design correctness metrics for the EEP ODP. . . . .	190
C.3	Design correctness metrics for the RC ODP. . . . .	191

C.4	Design correctness metrics for the FoI4EEPSA ontology module. . .	192
C.5	Design correctness metrics for the Q4EEPSA ontology module. . .	192
C.6	Design correctness metrics for the P4EEPSA ontology module. . .	193
C.7	Design correctness metrics for the EXR4EEPSA ontology module.	193
C.8	Design correctness metrics for the EXN4EEPSA ontology module.	194
C.9	Design correctness metrics for the EK4EEPSA ontology module. . .	194
C.10	Structural metrics for the AffectedBy ODP. . . . .	195
C.11	Structural metrics for the EEP ODP. . . . .	196
C.12	Structural metrics for the RC ODP. . . . .	197
C.13	Structural metrics for the FoI4EEPSA ontology module. . . . .	198
C.14	Structural metrics for the Q4EEPSA ontology module. . . . .	199
C.15	Structural metrics for the P4EEPSA ontology module. . . . .	200
C.16	Structural metrics for the EXR4EEPSA ontology module. . . . .	201
C.17	Structural metrics for the EXN4EEPSA ontology module. . . . .	202
C.18	Structural metrics for the EK4EEPSA ontology module. . . . .	203



# List of Tables

4.1	Summary of ontology design correctness evaluation by OOPS! . . .	80
4.2	Summary of ontology structural metrics by Protégé’s Ontology Metrics tab. . . . .	81
6.1	Tibucon sensors features and observations summary. . . . .	116
6.2	Summary of obtained results after applying baseline and SemOD outlier detection techniques. . . . .	118
6.3	Dataset benchmark summary. . . . .	121
6.4	Closest Euskalmet weather stations to IK4-Tekniker building measuring outdoor temperature (results obtained after executing SPARQL query shown in Listing 6.5 the 28/01/2019). . . . .	127
6.5	Predictive models and the variables used to build them. . . . .	130
6.6	MAE and RMSE obtained with different predictive models enabled by the EEP SA (best results were obtained with EEP SA #4). . . .	131
6.7	Comparison of mean discomfort duration per day (according to RITE regulation for winter days and INSHT regulation) suffered if HVAC control strategies proposed by EROSO and the OSCS were applied. . . . .	134
7.1	MAE and RMSE obtained with predictive models developed for different thermal zones of the use case farm. . . . .	146
B.1	Requirements addressed by the AffectedBy ODP. . . . .	184
B.2	Requirements addressed by the EEP ODP. . . . .	185
B.3	Requirements addressed by the RC ODP. . . . .	185
B.4	Requirements addressed by the FoI4EEP SA ontology module. . . .	186

B.5	Requirements addressed by the Q4EEPSA ontology module. . . . .	186
B.6	Requirements addressed by the P4EEPSA ontology module. . . . .	187
B.7	Requirements addressed by the EXR4EEPSA ontology module. . . . .	187
B.8	Requirements addressed by the EXN4EEPSA ontology module. . . . .	188
B.9	Requirements addressed by the EK4EEPSA ontology module. . . . .	188

# Chapter 1

## Introduction

Concerns over changing climatic conditions, energy security, and adverse environmental effects are growing among governments, researchers, policy makers, and scientists in developed as well as developing countries [1]. In order to meet the energy sustainability and minimize the climate change, the European Commission agreed a set of binding legislations inside the EU 2020 climate and energy package<sup>1</sup>. One of the spotlighted sectors regarding this package is the building sector, which consumes more than 35% of global energy and is responsible for nearly 40% of energy-related CO<sub>2</sub> emissions in the EU [2]. Therefore, efficient management of building energy plays a vital role and is becoming the trend for the future generation of buildings.

However, this is not the only concern related to buildings. In the early 2000s it was estimated that people spent around 90% of their time indoors [3], 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<sup>2</sup>, 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 sense, under the same thermal conditions a person may be shivering while another person may be sweating.

Although many times being an overlooked factor, extensive research has been conducted proving the impact of thermal comfort on humans. Some studies show the relation between indoor environment conditions and working efficiency or productivity [4, 5], which have a direct effect on company revenues. There is also work demonstrating that indoor environment conditions can have a significant impact on occupants comfort, morale, health and wellbeing in commercial office buildings [6]. 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 [7]. Therefore, all these evidences rein-

---

<sup>1</sup>[https://ec.europa.eu/clima/policies/strategies/2020\\_en](https://ec.europa.eu/clima/policies/strategies/2020_en)

<sup>2</sup><https://www.ashrae.org/technical-resources/bookstore/standard-55-thermal-environmental-conditions-for-human-occupancy>

force the need of ensuring comfortable thermal conditions in buildings.

Fulfilling occupants' comfort whilst reducing energy consumption is still an unsolved problem in most buildings. Furthermore, it is important to note that tertiary buildings have specific features which may further hinder this problem. For example, they normally contain spaces with bigger dimensions compared with the residential rooms which typically are rather small. These bigger spaces are prone to have bigger thermal inertia, which means that they require longer periods of time to heat up or cool down [8]. Therefore, 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. However, an efficient activation in advance of these systems has been historically full of intricacies due to the immaturity of technologies enabling the observation of environmental conditions or the prediction of future outcomes.

The expansion of the Internet of Things (IoT) [9] and Knowledge Discovery in Databases (KDD) [10] techniques may allow to improve matters in this regard. The IoT facilitates the monitoring of real-world qualities and events thanks to the devices equipped with electronic components and ubiquitous intelligence. This led to the massive amount of data available nowadays, which has the potential to enable new discoveries and improve decision-making processes. Certainly, KDD processes could also contribute to achieve the same goals as they enable the extraction of useful knowledge from raw data by means of five steps: data selection, preprocessing, transformation, data mining and interpretation. These KDD processes are performed by data analysts who develop predictive models to be exploited by the stakeholders.

However, the development of these predictive models is not straightforward as data coming from IoT 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, relevant data may also come from disparate external sources (often referred to as exogenous data), which further aggravates the data heterogeneity situation. This great variety of 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.

Data analysts facing energy efficiency and thermal comfort problems in tertiary buildings have to deal with the aforementioned data variety. This data encompasses description of building topology and 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 skylight with 2 m<sup>2</sup> 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) and other information related to buildings such as the space occupancy, work schedule or human related organization (e.g. the 29<sup>th</sup> November 2018 is a reduced working hours day; the occupancy value of the meeting room 06 at 11:00 is of 8 people). 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 12<sup>th</sup> May 2018 at 16:35; a blind actuator that lowered blinds of window 121 on 26<sup>th</sup> 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 10<sup>th</sup> June 2018 at 10:00 forecasting a relative humidity of 53% on 12<sup>th</sup> June 2018 at 15:00; a weather report that described cloudy skies during the morning of 6<sup>th</sup> December in Amsterdam).

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 have difficulties finding variables and tasks that could be confidently used to make accurate predictions. Furthermore, due to the plethora of possible combination of algorithms in each KDD phase, even expert data analysts may turn to a trial and error approach [11]. This is definitely an undesirable approach and it would be much more profitable to rely on a KDD process assistant supported by technologies that enable the management of the semantics and interrelationships of data, as well as the knowledge representation.

In this thesis Semantic Technologies such as ontologies, ontology-driven rules and ontology-driven data access are leveraged to support a KDD process assistant for the aforementioned problematic scenario in tertiary buildings, as Semantic Technologies enable the previously referred features (i.e. management of semantics and interrelationship of data, and knowledge representation). Specifically, the Energy Efficiency Prediction Semantic Assistant (EEPSA) is proposed, an approach that assists data analysts through the different KDD phases.

First of all, building related data needs to be semantically annotated with appropriate ontological terms. This comprises the annotation of features of interest (e.g. a room) and their respective qualities (e.g. a room's temperature), as well as observations and actuations (e.g. a temperature observation), the sensors and actuators that generate them (e.g. a temperature sensor), and the procedures used (e.g. a sensing procedure). Furthermore, observations and actuations have to be described with respect to their values, in addition to their spatial and temporal context. This semantic annotation is fundamental for enriching data, integrating heterogeneous data and representing it in a more domain-oriented way, as well as for enabling the improvement of the upcoming KDD phases.

In the data selection phase the data analyst is assisted to decide which might be the most relevant variables for the matter at hand (e.g. which structural properties influence an adequate warming of a given room? Does the season of the year have an effect in the interpretation of some sensor measurements? Is there any relation between the working calendar and the occupancy of specific rooms?). Ontology-driven queries and inferencing capabilities support this task.

The preprocessing phase intends to clean data from undesired noise, missing values or inconsistencies (e.g. is reliable the data measured under certain spatio-temporal context? Can data captured by a weather station replace the data captured by a sensor in a given context?). Ontology-driven rules help detecting

such problematic data and classifying them according to their potential cause, as well as in proposing possible methods to fix them according to the established goal.

The transformation phase generates additional knowledge in form of new attributes. External data sources are critical in this phase (e.g. which data sources may provide relevant data for a given scenario? Can data coming from a specific source be used to aggregate it to a sensor data?).

All the enhancements in these phases are aimed at improving the robustness and performance of machine learning algorithms applied in the data mining phase.

Afterwards, another set of ontology-driven rules and ontology-driven queries ease the interpretation of results obtained from the data mining phase (e.g. does a given room's forecasted temperature satisfy the workplace safety regulation?).

The EEPISA is focused on energy efficiency and thermal comfort problems in tertiary buildings. However, the proposed data analyst assistant is designed to be easily reused in similar use cases in different types of buildings. The main driver behind this feature is the EEPISA's foundation of Semantic Technologies. More specifically, the EEPISA ontology which is the cornerstone of the data analyst assistant and which, thanks to its high abstraction level and its modular design, can be easily customized. In this thesis, the EEPISA's reusability feature is evaluated in an animal welfare problem in a poultry farm.

## 1.1 Thesis objectives and contributions

The overall objective of this thesis is supporting data analysts through KDD processes in energy efficiency and thermal comfort problems in tertiary buildings, by exploiting Semantic Technologies. Towards this aim, the following specific actions are considered:

- The development of a core ontology that captures the relevant domain and expert knowledge for the KDD phases, and facilitates its customization for similar use cases in different types of buildings.
- The description of a process for supporting data analysts in different KDD phases.
- The implementation and evaluation of the proposed process in two real-world use cases.

In particular, this thesis makes the following contributions:

- The proposal of a set of ontology patterns to assist data analysts and overcome weaknesses in existing pattern-based ontologies - Section 4.3.

- The proposal of an ontology composed by a set of ontology modules that provides essential concepts and relations to incorporate the relevant domain and expert knowledge in energy efficiency and thermal comfort problems in tertiary buildings - Chapter 4.
- A process based on Semantic Technologies that assists data analysts in different KDD phases towards the development of enhanced predictive models - Chapter 5.

## 1.2 Thesis structure

The outline below specifies the organization of this thesis.

- Chapter 2: Fundamental Technologies. This chapter presents an overview of the basic technologies addressed in this thesis. This chapter is not aimed at providing an exhaustive insight of these technologies, but instead a brief introduction to them.
- Chapter 3: KDD with Semantic Technologies: Related Work. This chapter shows an extended overview of the existing approaches leveraging Semantic Technologies in the different KDD phases.
- Chapter 4: The EEP SA Ontology. This chapter describes the proposed core ontology itself, as well as its design and development process, the different ontology modules and patterns, proposed customization method, documentation and evaluation.
- Chapter 5: The EEP SA. This chapter presents the data analyst assistant based on Semantic Technologies and for each KDD phase, the main contributions are specified.
- Chapter 6: The EEP SA in an Office. This chapter shows the implementation and evaluation of the EEP SA in a real-world office.
- Chapter 7: The EEP SA in a Poultry Farm. This chapter shows the EEP SA's customization, implementation and evaluation for an animal welfare problem in a real-world poultry farm.
- Chapter 8: Conclusions. This chapter summarizes the major contributions of the thesis and the conclusions reached. Furthermore, future directions of research for exploiting Semantic Technologies in KDD processes are discussed.

## 1.3 Published work

Parts of the work presented in this dissertation were published in journals, conferences or in other venues. Next, a complete list of such publications is shown:

- I. Esnaola-Gonzalez, Semantic Web Technologies to Enhance the Knowledge Discovery Process in Predictive Analytics, in: *Doctoral Consortium on Knowledge Discovery, Knowledge Engineering and Knowledge Management (DC3K 2016)*, Porto, Portugal, 2016, pp. 17-23.
- I. Esnaola-Gonzalez, J. Bermúdez, I. Fernandez, S. Fernandez and A. Arnaiz, Towards a Semantic Outlier Detection Framework in Wireless Sensor Networks, in: *Proceedings of the 13th International Conference on Semantic Systems*, Semantics2017, ACM, New York, NY, USA, 2017, pp. 152-159. ISBN 978-1-4503-5296-3. doi:10.1145/3132218.3132226.
- I. Esnaola-Gonzalez, J. Bermúdez, I. Fernandez and A. Arnaiz, Semantic prediction assistant approach applied to energy efficiency in Tertiary buildings, *Semantic Web* **9**(6) (2018), 735-762. doi:10.3233/SW-180296.
- I. Esnaola-Gonzalez, J. Bermúdez, I. Fernandez and A. Arnaiz, Supporting Predictive Models Results Interpretation for Comfortable Workplaces, in: *Proceedings of the ISWC 2018 Posters & Demonstrations, Industry and Blue Sky Ideas Tracks co-located with 17th International Semantic Web Conference (ISWC 2018)*, Vol. 2180, CEUR, 2018.
- 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.
- I. Esnaola-Gonzalez, J. Bermúdez, I. Fernandez and A. Arnaiz, EROSO: Semantic Technologies Towards Thermal Comfort in Workplaces, in: *Proceedings of the 21th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2018)*, C.F. Zucker, C. Ghidini, A. Napoli and Y. Toussaint, eds, Springer International Publishing, 2018, pp. 519-533. doi:10.1007/978-3-030-03667-6\_33.

Furthermore, parts of the work presented in this dissertation also belongs to an article which is under review:

- I. Esnaola-Gonzalez, J. Bermúdez, I. Fernandez and A. Arnaiz, EEPsA as a core ontology for energy efficiency and thermal comfort in buildings, *Semantic Web* (Under review).

## Chapter 2

# Fundamental Technologies

This chapter introduces the basics of the fundamental technologies of this thesis: the KDD process, the Semantic Web and Semantic Technologies.

### 2.1 KDD

The KDD (Knowledge Discovery in Databases) is a process leading to the extraction of useful knowledge from raw data [10]. This process is composed of the following five steps:

- **Data Selection.** It consists in selecting the datasets and the subset of variables or data samples where the knowledge discovery is going to be performed. With the advent of new paradigms such IoT or LD, data analysts may get lost in today's chaotic information universe. As a matter of fact, much of this available data may be redundant and therefore, it hinders the knowledge extraction as well as making it more time and resource-consuming. Therefore, in order to ease the upcoming KDD phases, data analysts need to put their domain knowledge to work to select the sets of data and variables used to do the analysis.
- **Preprocessing.** Different methods are applied to ensure quality of the data and prepare the data for a subsequent analysis. Nowadays, datasets are prone to suffer from noise, outliers, missing values, and inconsistencies due to their typical big size and their probable origin from multiple and heterogeneous sources. Not only do these data quality issues compromise knowledge extraction algorithms' performance, but they also may have a negative impact on decision-making processes.
- **Transformation.** The data is changed into a form which data mining algorithms can work with and improve their performance. This phase comprises different tasks although there are two of them which are particularly relevant: feature generation and feature selection. These two tasks are related,

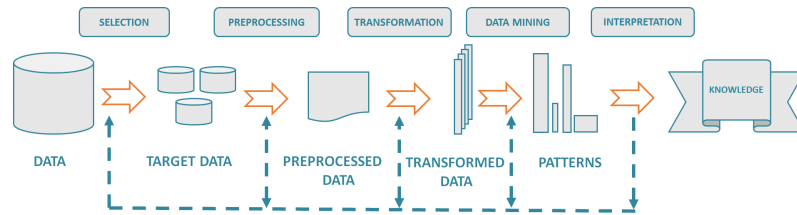


Figure 2.1: An overview of the steps that compose the KDD process proposed by Fayyad et al.

and often applied subsequently, because it is useful to post-process the set of created features and discard features that have little value.

- **Data Mining.** The data analysis or discovery algorithm that best matches the data analyst's goals is applied searching for hidden patterns in the data. Data analyst's role in this phase consists in selecting the suitable algorithm and fine-tuning it with the appropriate parameters. Furthermore, as each algorithm's performance may vary depending on the input data, data analysts expertise and even intuition at times also play a role in this phase.
- **Interpretation.** It is the final phase where the results, patterns and models derived are used to support decision-making processes. This phase also relies on the data analysts knowledge in the domain at hand, and even for a domain-expert, this task may end up being challenging in certain scenarios.

This is an interactive and iterative process rather than a strict workflow. It involves numerous loops and many decisions made between any two of the mentioned steps. The necessity of having such a flexible process arises from the wide range of methods and parameter selections that can be applied in each step. An overview of the flow of KDD process steps is illustrated in Figure 2.1.

## 2.2 The Semantic Web and Semantic Technologies

Nowadays, most Web content is suitable for human consumption, but it is not well supported by machines. This derived in the advent of the Semantic Web, which is not a separate Web but an extension in which information is given well-defined meaning, enabling computers and people to work in cooperation [12]. In fact, the Semantic Web builds upon the principles and technologies of the Web. It reuses the Web's global indexing and naming scheme, and Semantic Web documents can be accessed through standard Web browsers as well as through semantically aware applications [13]. The World Wide Web was derived from a new way of thinking about sharing information. Therefore, it has a set of features that can be summarized as follows [14]:

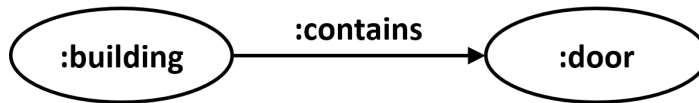


Figure 2.2: An RDF Triple example.

- The AAA (Anyone can say Anything about Any topic) slogan. In a web of documents, this slogan means that anyone can write a page saying whatever they want and publish it. In the case of the Semantic Web, AAA means that any individual can express a piece of data about some entity and this data can be combined with information from other sources.
- The Open World Assumption (OWA). As a consequence of the AAA slogan, there could always be something new. Therefore, statements about knowledge that are not included or inferred from the explicitly recorded data may be considered unknown, rather than wrong or false.
- Non unique naming assumption. This feature is built upon the assumption that not all the contributors to the Web will coordinate with regards to the naming of entities. Therefore, the same entity could be referred to using more than one name.
- The network effect. This is the property thanks to which the value of joining in the Semantic Web increases with the number of people who have already joined, resulting in a spiral of participation.
- The data wilderness. The condition of the data that contains valuable information, but there is no guarantee that it will be readily understandable.

### 2.2.1 The data model: RDF

The Resource Description Framework (RDF) [15] is a W3C (World Wide Web Consortium) recommendation for representing information on the Web. The basic structure are triples, which consist of a subject, a predicate and an object. Figure 2.2 exemplifies the RDF triple structure with the subject (building) on the left and object (door) on the right connected by a predicate (contains). A set of RDF triples constitutes an RDF graph, which can be viewed as node and directed-arc diagrams.

These resources are described using IRIs (Internationalized Resource Identifier). The IRI extends the ASCII characters subset of the URIs (Uniform Resource Identifier). Since a property is also an IRI, it can again be used as a resource interlinked to another resource. Furthermore, in RDF, IRIs can refer to anything. This flexibility makes the data model suitable for the context of an open Web.

It is important to note that RDF is not a data format, but a data model for describing resources as node-and-arc-labelled directed graphs. Therefore, although expressing RDF triples as a graph may be suitable to display data, this

may not be the most compact or human-friendly way to see the relation between entities. These needs derived in different RDF serialization formats. RDF/XML and RDFa are standardized by the W3C, but there are many other more easily understandable non-standard serialization formats such as N-Triples and Turtle.

**RDF/XML.** This syntax is widely used to publish Linked Data on the Web. However, this syntax may be rather difficult for humans to read or write. The following RDF excerpt represents a building with twelve storeys.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:bo="http://example.org/buildingOntology#">
  <rdf:Description rdf:about="http://example.org/myBuilding">
    <bo:numberOfStoreys>12</bo:numberOfStoreys>
  </rdf:Description>
</rdf:RDF>
```

**RDFa.** This serialization format embeds RDF triples in HTML documents. RDF data is not embedded in comments within the HTML document, but instead, it is interwoven within the HTML document. The following excerpt represents a given building constructed by the architect William Graham and inaugurated on 5<sup>th</sup> December 2009.

```
<div vocab="http://example.org/buildingOntology#"
  typeof="Construction">
  <span property="buildingConstructed">Building
  AF-29084</span>
  Constructed by
  <span property="architect">William Graham</span> and
  inaugurated on <span property="constructionDate"
  content="2009-12-05">December 5</span>.
  </span>
</div>
```

**N-Triples.** This serialization form refers to resources using their fully unabbreviated IRIs [16]. The simplest triple statement is a sequence of IRIs separated by a white space and ended with a dot ('.'). The following triple asserts that a given building contains a given room.

```
<http://example.org/myBuilding>
<http://example.org/buildingOntology#containsRoom>
<http://example.org/room03> .
```

**Turtle.** This serialization combines the display of N-Triples with the terseness of QNames [17], which results in a more compact triple representation [18]. The following triple asserts that a given resource is a building.



```
@PREFIX ex:<http://example.org/>
@PREFIX bo:<http://example.org/buildingOntology#>

ex:myBuilding rdf:type bo:Building .
```

### 2.2.2 Linked Data

The term Linked Data (LD) refers to a set of best practices for publishing and interlinking structured data on the Web [19]. These best practices are also known as Linked Data principles<sup>1</sup>, and they can be summarized as follows:

- Use URIs as names for things.
- Use HTTP URIs, so that people can look up those names.
- When someone looks up a URI, provide useful information, using the standards.
- Include links to other URIs so that they can discover more things.

To publish data on the Web, Linked Data uses HTTP URIs to identify the real-world items of a domain of interest. Other URI schemes such as URNs (Uniform Resource Name) and DOIs (Digital Object Identifier) are avoided, as HTTP URIs enable creating globally unique names in a decentralized way, and they serve as a means of accessing information describing the identified entity.

Any HTTP URI should be dereferenceable, which means that HTTP clients should be able to look up the URI and retrieve a description of the resource identified by such a URI. Furthermore, these descriptions should ideally be represented as HTML when they are intended to be read by humans, and as RDF data if intended consumers are machines. This can be achieved with an HTTP mechanism called content negotiation. This mechanism consists in HTTP clients sending HTTP headers with each request indicating which kind of documents they prefer. Afterwards, servers examine these headers and select the appropriate response.

The LOD (Linked Open Data) Cloud<sup>2</sup> presents datasets published in the Linked Data format. As of June 2018, the LOD cloud contained 1,231 datasets with 16,132 links.

### 2.2.3 The Query Language: SPARQL SELECT

SPARQL [20] (SPARQL Protocol and RDF Query Language) is a query language which can be used to express queries across diverse data sources, whether

---

<sup>1</sup><https://www.w3.org/DesignIssues/LinkedData.html>

<sup>2</sup><https://lod-cloud.net/>

the data is stored natively as RDF or viewed as RDF via middleware. It is a W3C recommendation as of 2008 and enables querying information that can be RDF graphs or results sets.

The syntax of a SPARQL query is similar to the SQL query syntax, as both of them use keywords such as SELECT to determine which subset of the selected data is returned, and WHERE to define graph patterns to find.

A SPARQL SELECT query has two parts: a set of question words, and a question pattern. The following SPARQL query retrieves the height of a given wall.

```
@PREFIX bo:<http://example.org/buildingOntology#>

SELECT ?wallHeight
WHERE { :wall105 bo:hasHeight ?wallHeight .}
```

More complex SPARQL queries can also be constructed by left join (OPTIONAL operator), union (UNION operator) and constraints (FILTER operator).

### 2.2.4 The Inferencing: RDFS, OWL and rule languages

RDF Schema (RDFS) [21], officially called “RDF Vocabulary Description Language”, is an extension of the RDF schema to describe vocabularies used in RDF descriptions with more complex semantic constraints. It contains mechanisms for representing groups of related resources and the relationships between them.

RDF and RDFS allow the definition of classes (*rdfs:Class*) and their instantiations (*rdf:type*), properties (*rdf:Property*) and the domain (*rdfs:domain*) and the range (*rdfs:range*) of their related individuals, as well as hierarchical relationships such as subclasses (*rdfs:subClassOf*) and subproperties (*rdfs:subPropertyOf*). In addition, they enable the representation of other annotation properties such as the readable name of a resource (*rdfs:label*) or the relation of a resource to another which explains it (*rdfs:seeAlso*).

The Web Ontology Language (OWL) [22] was designed to address the need to process the content of information rather than just representing it. OWL enables a greater machine interpretability of Web content compared with RDF or RDFS by providing additional vocabulary and formal semantics. For example, OWL provides a set of mechanisms to define inverse, transitive, symmetric or functional properties.

The OWL 2 Web Ontology Language [23], often referred to as OWL 2, is an ontology language for the Semantic Web with formally defined meaning. There are two alternative ways of assigning meaning to ontologies in OWL 2 called the Direct Semantics<sup>3</sup> and the RDF-Based Semantics<sup>4</sup>. The former can be applied to

<sup>3</sup><http://www.w3.org/TR/owl-direct-semantics>

<sup>4</sup><http://www.w3.org/TR/owl-rdf-based-semantics>

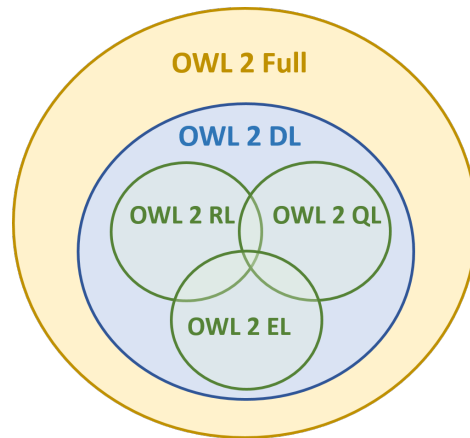


Figure 2.3: Venn diagram showing the relation between OWL 2 profiles.

ontologies that are within the OWL 2 DL subset of OWL 2. Ontologies that are not in OWL 2 DL belong to OWL 2 Full, and can only be interpreted under RDF-Based Semantics. Furthermore, OWL 2 defines three different profiles: OWL 2 EL, OWL 2 QL and OWL 2 RL. Each profile is defined as a subset of the OWL 2 structural elements that can be used in a conforming ontology. That is, in each profile, a number of statements that can be used in OWL2DL is not allowed. Furthermore, each profile trades off different aspects of OWL’s expressive power in return for important advantages in particular application scenarios. Figure 2.3 shows the relation between the OWL 2 profiles.

OWL 2 EL is suitable for applications where ontologies defining very large numbers of classes and/or properties are employed. It captures the expressive power used by many such ontologies, and for which ontology consistency, class expression subsumption, and instance checking can be decided in polynomial time. OWL 2 QL is designed so that complete query answering is in LOGSPACE (more precisely, in  $AC^0$ ) with respect to the size of the data (assertions). It is suitable for applications where relatively lightweight ontologies are used to represent large numbers of individuals and data needs to be accessed directly via relational queries such as SQL. OWL 2 RL is suitable for applications where relatively lightweight ontologies are used to represent large numbers of individuals and it is necessary to operate on data in the form of RDF triples.

Furthermore, OWL 2 adds new functionalities to address some of the issues identified in OWL’s previous version. Some of the new features are syntactic sugar which enable expressing things in an easier way, and others are new expressivities. These features include among others property chains, richer datatypes and data ranges, and enhanced annotation capabilities.

RDF(S) and OWL provide the basis to enable working with inferencing of implicit knowledge from explicitly asserted knowledge.

There may be scenarios where OWL expressivity may not suffice certain desired inferences. In order to fill this gap, rule languages provide useful knowledge

representation formalisms that, in combination with existing data, allow the discovery of new relationships. These new relationships can be explicitly added to the existing data or returned at query time, depending on the needs. There are many language rules and the main ones include SWRL, SPIN and SPARQL CONSTRUCT.

**SWRL.** SWRL<sup>5</sup> (Semantic Web Rule Language) is a combination of OWL DL language with RuleML<sup>6</sup>. SWRL includes a high-level abstract syntax for Horn-like rules in OWL DL and all rules are expressed in terms of classes, properties and individuals. The following SWRL rule asserts that if a building has a room (*hasRoom*) and such a room contains a given device (*roomContainsDevice*), then it is implied that such device is also contained in the building (*buildingContainsDevice*).

```
Room(?r) ^ hasDoor(?r, ?d) ^ OutDoor(?d) ^ isAdjacent(?p, ?r)
  => isForExit(?p, ?r)
```

**SPIN.** SPIN<sup>7</sup> (SPARQL Inference Notation) is a W3C Member Submission rule and constraint language based on SPARQL. It provides reusable query templates and extends SWRL capabilities, as it leverages object-oriented principles and it is more expressive than SWRL thanks to being based on SPARQL syntax. The following SPIN rule defines the individual *:door035* of type *ex:SteelDoor* instead of type *ex:WoodenDoor*.

```
[ a      sp:Modify ;
  sp:graphIRI <urn:building:graph> ;
  sp:deletePattern ([ sp:object ex:WoodenDoor ;
                    sp:predicate rdf:type ;
                    sp:subject :door035
                    ]) ;
  sp:insertPattern ([ sp:object ex:SteelDoor ;
                    sp:predicate rdf:type ;
                    sp:subject :door035
                    ]) ;
  sp:where      ([ sp:object ex:WoodenDoor ;
                  sp:predicate rdf:type ;
                  sp:subject :door035
                  ])
]
```

**SPARQL CONSTRUCT.** Although the previous SPARQL section focused on SPARQL SELECT queries, SPARQL has other three forms: CONSTRUCT, ASK, and DESCRIBE. The CONSTRUCT query form returns a single RDF graph specified by a graph template. Additionally, the SPARQL CONSTRUCT

<sup>5</sup><https://www.w3.org/Submission/SWRL/>

<sup>6</sup>[ruleml.org](http://ruleml.org)

<sup>7</sup><https://www.w3.org/Submission/spin-overview/>

query can be used to define rules. The following SPARQL CONSTRUCT classifies rooms containing windows as individuals of class *bo:RoomWithWindow*.

```
@PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@PREFIX bo:<http://example.org/buildingOntology#>

CONSTRUCT { ?room rdf:type bo:RoomWithWindow }
WHERE      { ?room rdf:type bo:Room;
             bo:hasWindow ?win. }
```

### 2.2.5 Ontologies

Ontologies appear as a way to describe and represent the concepts and relationships of a certain domain. The term ontology was first used in philosophy to define the study of the nature of being, existence, or reality, as well as the basic categories of being and their relations. In computer and information science field an ontology can be defined as “a formal, explicit specification of a shared conceptualization” [24].

An ontology can represent a certain phenomenon, topic, or subject area through the description of classes, properties and instances (also known as individuals).

- Classes are abstract groups, sets, or collections of objects and represent ontology concepts. Furthermore, these classes can have a hierarchical relation and can be arranged in taxonomies of super classes and sub classes.
- Properties represent features or characteristics of individuals as well as the relationship between them.
- Instances represent individuals of the classes described in the ontology.

Ontologies can be constructed based on different ontology languages such as OWL 2 and their profiles. Certainly, an ontology language provides the expressive capability to encode knowledge about specific domain and often include reasoning rules that support the processing of such knowledge.

According to their level of generality, there are different ontology types and categories [25]:

- Top-level ontologies (often referred to as upper ontology or foundation ontology, general, or cross domain ontology) represent very general concepts such as time, space, events which are independent of a specific domain or problem.
- Domain ontologies describe fundamental concepts according to a generic domain and specialize terms introduced in top-level ontology.

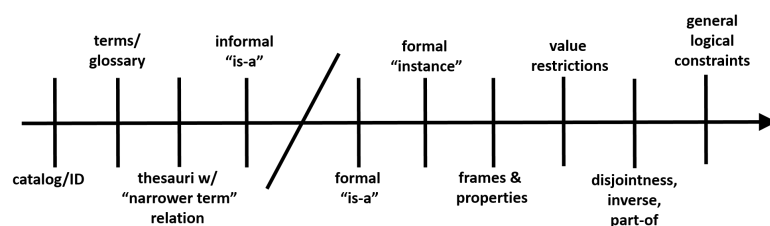


Figure 2.4: The Ontology spectrum as defined by Lassila and McGuinness.

- Task ontologies describe fundamental concepts according to a general activity or task and specialize the terms introduced in top-level ontologies.
- Application ontologies are specialized ontologies focused on a specific task and domain. They are often a specialization of both task and domain ontologies, and they also often specify roles played by domain entities for specific activity.

Ontologies can also be viewed as a spectrum of detail in their specification [26], as shown in Figure 2.4.

At the lower end of spectrum there are catalogues, which consist in a finite list of terms used for expressing knowledge of information. This list of terms may not have descriptions at all, and their meaning can only be estimated because they are chosen from natural language. Likewise, there are no formal relations expressed between these terms, apart from informal relations in natural language. Usually, such specifications are not referred to as ontologies.

When at least one formal relation is defined and used between terms, the concept “ontology” can be used to refer to such a catalogue. From this point onwards, there are languages that provide sets of constructs to describe an ontology, such as frames or simplified logics. As the specificity increases, the precision and the ability to use tools to automatically integrate systems also increases. However, the cost of building and maintaining a metadata registry also increases.

## Chapter 3

# KDD with Semantic Technologies: Related Work

As mentioned in the previous chapter, the KDD process and Semantic Technologies are the two fundamental technologies of this thesis. This chapter makes an extended review of the presence of Semantic Technologies in the different KDD phases. Furthermore, as the incorporation of Semantic Technologies to the KDD process requires from a previous semantic annotation phase, related work in this area is also reviewed.

### 3.1 Semantic Technologies in Annotation

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. 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 [27]. 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 these benefits, annotating data semantically can contribute improving the upcoming KDD phases [28, 29, 30]. In energy efficiency and thermal comfort problems for tertiary buildings, features of interest and their respective qualities, as well as observations and actuations, the sensors and actuators that generate them, and the procedures used are relevant data to be semantically annotated. Furthermore, not only observations' and actuations' values but also their spatial

and temporal context are of utmost importance for the mentioned problems, and therefore, they are also worth being semantically annotated.

Next, the most relevant ontologies covering domains of discourse of this thesis are reviewed. Specifically, building domain ontologies are reviewed in section 3.1.1; ontologies addressing observations, actuations and related concepts in section 3.1.2; ontologies representing the spatio-temporal context and units of measurement of these observations in section 3.1.3; ontologies covering the KDD process in section 3.1.4; and other related domain ontologies in section 3.1.5. Reviewed ontologies are further discussed in section 4.4.

### 3.1.1 Building domain ontologies

BIM (Building Information Model) is a process used by different stakeholders involved in the construction process of a building, and deals with the digital representation of functional and physical characteristics of a building [31]. 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.

A BIM model may contain static information of a building element. For example, 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 can be queried. Nevertheless, it is not possible to know whether the window is opened or closed in a given moment. As a matter of fact, the integration of static building information and sensing data becomes a prime challenge [32]. Furthermore, the use of IFC<sup>1</sup> (Industry Foundation Classes) files for exchanging BIM data has arisen several issues due to its complexity and time-consumption [33]. Therefore, it can be stated that more often than not easy and intuitive ways to rapidly browse, query and use BIM information are not available [34].

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 [34]. 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 are 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 [35].

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 and limits the scalability of applications. In this section, the most

---

<sup>1</sup>IFC is the open standard developed by buildingSMART (<https://www.buildingsmart.org/>).



relevant ontologies for modelling buildings are reviewed.

### 3.1.1.1 ifcOWL Ontology

The ifcOWL ontology<sup>2</sup> [36] provides an OWL representation of the EXPRESS schemas of IFC (ISO 16739:2013<sup>3</sup>) 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. 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 [34]. For example, the ifcOWL conceptualization of some relationships and properties as instances of classes (e.g. *ifc:IfcRelationship* and *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<sup>4</sup> [37], 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 [38] or making the ontology more flexible in terms its capability to deal with the real-world scenarios [39].

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.

### 3.1.1.2 SAREF4BLDG

SAREF4BLDG<sup>5</sup> [40] is an extension of the SAREF ontology (explained in Section 3.1.2.3) 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, SAREF4BLDG 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

---

<sup>2</sup>[http://ifcowl.openbimstandards.org/IFC4\\_ADD2.owl](http://ifcowl.openbimstandards.org/IFC4_ADD2.owl)

<sup>3</sup><https://www.iso.org/standard/51622.html>

<sup>4</sup>At the moment of writing this dissertation, the ontology is not publicly available.

<sup>5</sup><https://w3id.org/def/saref4bldg>

the class representing physical objects, are declared as subclasses of *geo:Spatial-Thing* in order to reuse the conceptualization for locations already proposed by the Basic Geo vocabulary (also known as WGS84 Geo Position vocabulary).

### 3.1.1.3 DogOnt

The DogOnt ontology<sup>6</sup> [41] formalizes IDE (Intelligent Domotic Environment) aspects and it is designed with a particular focus on interoperation between domotic systems. Although it primarily models devices, states and functionalities, DogOnt 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) [42].

DogOnt authors claim that it 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 dissertation (version 4.0.1), counts with over 1,000 classes and over 70 properties, which may be rather large in some cases.

### 3.1.1.4 EEOnt

The Energy Efficiency Ontology [43] (EEOnt) is an ontology that defines the general structure of a building, the distribution and the connectivity of its systems, objects, and spaces. Furthermore, the functionality and characteristics of the energy consuming devices and systems are also represented. EEOnt describes  $EEL_B$  (Energy Efficiency Index for Buildings) and  $EEL_L$  (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<sup>7</sup> [44]). Therefore, the modelling of the building environment, space and object topology is very similar to DogOnt, 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

<sup>6</sup><http://elite.polito.it/ontologies/dogont.owl>

<sup>7</sup><http://elite.polito.it/ontologies/poweront.owl>

properties of those materials are specified following the IFC model.

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

### 3.1.1.5 ThinkHome Ontology

The ThinkHome ontology<sup>8</sup> [45] formalizes all the 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<sup>9</sup> and EnergyResourceOntology<sup>10</sup>.

The building information module (BuildingOntology<sup>11</sup>) 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.

The ThinkHome BuildingOntology comprises all the necessary 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. However, being such an extensive ontology (with more than 250 classes and 400 properties), its scarce documentation hinders its understanding.

### 3.1.1.6 BOT

The Building Topology Ontology<sup>12</sup> [46] (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 [47]. 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.

---

<sup>8</sup><https://www.auto.tuwien.ac.at/downloads/thinkhome/ontology/>

<sup>9</sup><https://www.auto.tuwien.ac.at/downloads/thinkhome/ontology/WeatherOntology.owl>

<sup>10</sup><https://www.auto.tuwien.ac.at/downloads/thinkhome/ontology/EnergyResourceOntology.owl>

<sup>11</sup><https://www.auto.tuwien.ac.at/downloads/thinkhome/ontology/BuildingOntology.owl>

<sup>12</sup><https://w3id.org/bot>

Moreover, the W3C LBD (Linked Building Data) Community Group<sup>13</sup> 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<sup>14</sup>) 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<sup>15</sup> [48] 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<sup>16</sup>.

It is worth mentioning that, BOT is aligned with related domain ontologies such as ifcOWL, DogOnt and Brick [49].

### 3.1.1.7 FIEMSER Ontology

The FIEMSER ontology<sup>17</sup> 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 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 file 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.

### 3.1.1.8 Brick Ontology

Brick<sup>18</sup> [50] 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

<sup>13</sup><https://www.w3.org/community/lbd/>

<sup>14</sup><https://github.com/w3c-lbd-cg/product>

<sup>15</sup><https://github.com/w3c-lbd-cg/opm/blob/master/opm.ttl>

<sup>16</sup><https://github.com/w3c-lbd-cg/props>

<sup>17</sup><https://sites.google.com/site/smartappliancesproject/ontologies/fiemser-ontology>

<sup>18</sup><https://brickschema.org/>

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<sup>19</sup>) 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.

### 3.1.2 Observations, actuations and other related domain ontologies

The rapid adoption of the IoT leads to an exponential growth of the number of existing devices worldwide. 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 [51].

It has been proved that ontology-based approaches could contribute in achieving semantic interoperability [52], for example by linking each data element to ontology terms thus providing them with semantics [53, 54, 55]. Furthermore, these approaches enable the discovery of IoT services, data and resources [56]. However, defining a comprehensive unified ontology for the domain of IoT may be challenging as there are more than 200 ontologies available [57].

There are some concepts that are common to the majority of IoT platforms [58], such as sensors, actuators and their corresponding observations<sup>20</sup> and actuations. In fact, these concepts comprise an important area of discourse of the problem tackled in this thesis. Next, a set of relevant ontologies covering these concepts are reviewed.

#### 3.1.2.1 SSN Ontology

The initial Semantic Sensor Network (SSN) ontology<sup>21</sup> [59] was developed by the W3C Semantic Sensor Networks Incubator Group (SSN-XG) and it proposed

---

<sup>19</sup><http://project-haystack.org/>

<sup>20</sup>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.

<sup>21</sup><http://purl.oclc.org/NET/ssnx/ssn>

a conceptual schema for describing sensors, accuracy and capabilities of such sensors, their observations and methods used for sensing them. 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 ultra-lite (DUL) ontology and built on top of the Stimulus-Sensor-Observation (SSO) [60] Ontology Design Pattern (ODP) describing the relationships between sensors, stimulus, and observations.

The W3C Spatial Data on the Web Working Group (SDWWG<sup>22</sup>) proposed an update of the SSN ontology<sup>23</sup> [61] (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<sup>24</sup> (Sensor, Observation, 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<sup>25</sup>. 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.

### 3.1.2.2 om-lite Ontology

The om-lite ontology<sup>26</sup> [62] 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)<sup>27</sup>. 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

<sup>22</sup><https://www.w3.org/2015/spatial>

<sup>23</sup><http://www.w3.org/ns/ssn/>

<sup>24</sup><http://www.w3.org/ns/sosa/>

<sup>25</sup><https://www.w3.org/ns/ssn/dul>

<sup>26</sup><http://def.seegrid.csiro.au/ontology/om/om-lite>

<sup>27</sup><https://www.iso.org/standard/32574.html>

classified by the result type. This way, the class *oml:Observation* has subclasses such as *oml:CountObservation* for observations whose results are integer values, *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 pre-existing 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 run-time by a suitable concrete representation of the concept. Finally, it is aligned with PROV-O (explained in section 3.1.2.9), as well as some other domain ontologies (e.g. the previous version of the SSN ontology).

### 3.1.2.3 SAREF Ontology

The Smart Appliances REference (SAREF) ontology<sup>28</sup> [63] 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<sup>29</sup> for the environment domain, and SAREF4ENER<sup>30</sup> for the energy domain. Furthermore, at the moment of writing this dissertation there are three new planned extensions: SAREF4CITY for smart cities, SAREF4INMA for industry and manufacturing, and SAREF4AGRI for the agricultural domain.

---

<sup>28</sup><http://ontology.tno.nl/saref>

<sup>29</sup><https://w3id.org/def/saref4envi>

<sup>30</sup><https://w3id.org/saref4ener>

### 3.1.2.4 SEAS Ontology

The SEAS Ontology<sup>31</sup> [64] 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<sup>32</sup>. The SEAS ontology modules are developed based on the following three core modules: the SEAS Feature of Interest ontology<sup>33</sup> which defines features of interest (*seas:FeatureOfInterest*) and their qualities (*seas:Property*), the SEAS Evaluation ontology<sup>34</sup> describing evaluation of these qualities, and the SEAS System ontology<sup>35</sup> representing virtually isolated systems connected with other systems. The Procedure Execution (PEP) ontology<sup>36</sup>, 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 example, the SEAS Electric Power System ontology<sup>37</sup> 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.

### 3.1.2.5 IoT-O Ontology

The IoT-O ontology<sup>38</sup> [65] is an 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<sup>39</sup>.
3. A service module, based on MSM<sup>40</sup> (Minimal Service Model) and hRESTS ontology<sup>41</sup>.

---

<sup>31</sup><https://w3id.org/seas/>

<sup>32</sup><https://portal.etsi.org/STF/STFs/STFHomePages/STF556>

<sup>33</sup><https://w3id.org/seas/FeatureOfInterestOntology>

<sup>34</sup><https://w3id.org/seas/EvaluationOntology>

<sup>35</sup><https://w3id.org/seas/SystemOntology>

<sup>36</sup><https://w3id.org/pep/>

<sup>37</sup><https://w3id.org/seas/ElectricPowerSystemOntology>

<sup>38</sup><https://www.irit.fr/recherches/MELODI/ontologies/IoT-0>

<sup>39</sup><https://www.irit.fr/recherches/MELODI/ontologies/SAN>

<sup>40</sup><http://iserve.kmi.open.ac.uk/ns/msm>

<sup>41</sup><http://www.wsmo.org/ns/hrests/>



4. A lifecycle module<sup>42</sup>, based on a lifecycle vocabulary (a lightweight vocabulary defining state machines) and an IoT-specific extension.
5. An energy module, based on PowerOnt.

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 (Actuation-Actuator-Effect) pattern, which intends to model the relationship between an actuator and the effect it has on its environment through actuations.

### 3.1.2.6 FIESTA-IoT Ontology

The FIESTA-IoT Ontology<sup>43</sup> [66] aims at creating a lightweight ontology that achieves semantic interoperability among heterogeneous testbeds. The ontology is focused on the description of the underlying testbeds' resources 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 vocabulary, IoT-Lite ontology, OWL-Time ontology, and DUL ontology<sup>44</sup>.

The previous version of the SSN ontology has a strong influence in FIESTA-IoT when describing sensors and observations. The central class is *ssnx:Observation*, which is related with *ssnx:Sensor* who made it, the property it observes (*qu:QuantityKind*) 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<sup>45</sup> [67] 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 sensor 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<sup>46</sup> (Semantic Web for Earth and Environmental Terminology) and the previous version of the SSN.

<sup>42</sup><https://www.irit.fr/recherches/MELODI/ontologies/IoT-Lifecycle>

<sup>43</sup><http://ontology.fiesta-iot.eu/ontologyDocs/fiesta-iot/doc>

<sup>44</sup><http://www.ontologydesignpatterns.org/ont/dul/DUL.owl>

<sup>45</sup><http://www.w3.org/Submission/iot-lite/>

<sup>46</sup><https://sweet.jpl.nasa.gov/>

IoT-Lite is built around the main three concepts which according to 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<sup>47</sup> is a light version of the M3 ontology [68], 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:QuantityKind*) 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.

### 3.1.2.7 SmartEnv Ontology

The SmartEnv ontology<sup>48</sup> [69] 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.

---

<sup>47</sup><http://ontology.fiesta-iot.eu/ontologyDocs/fiesta-iot/doc>

<sup>48</sup><https://w3id.org/smartenvironment/smartenv.owl>

### 3.1.2.8 The S3N Ontology

The Semantic Smart Sensor Network (S3N) ontology<sup>49</sup> [70] 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 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.

### 3.1.2.9 PROV-O

PROV-O<sup>50</sup> [71] (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.

<sup>49</sup><https://github.com/s3n-ontology/s3n/blob/master/s3n.ttl>

<sup>50</sup><https://www.w3.org/TR/prov-o/>

### 3.1.3 Spatio-temporal and unit context ontologies

Observations and actuations are the central elements of the problem tackled in this thesis, 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.

#### 3.1.3.1 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 [72, 73, 74, 75, 76], even though the most commonly used ontology is the Time Ontology in OWL<sup>51</sup> [77] (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.

#### 3.1.3.2 Location

Together with time, spatial location is the other primary aspect that may help specifying a context. The Basic Geo vocabulary<sup>52</sup> 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 [78]. GeoSPARQL [79] 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.

#### 3.1.3.3 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. [80] 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 example, the concept “kind of quantity”, is denoted as “physical quality” by

---

<sup>51</sup><https://www.w3.org/TR/owl-time/>

<sup>52</sup><https://www.w3.org/2003/01/geo/>

MUO<sup>53</sup> (Measurement Units Ontology), and as “quantity kind” by QU<sup>54</sup> (Ontology for Quantity Kinds and Units) and QUDT<sup>55</sup> (Quantities, Units, Dimensions and Data Types Ontologies). OBOE<sup>56</sup> (Extensible Observation Ontology), OM<sup>57</sup> (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<sup>58</sup> 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 quality 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 dissertation 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.
```

**UCUM Datatypes.** The work presented by Lefrançois et al. [81] 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.

<sup>53</sup><http://idi.fundacionctic.org/muo/>

<sup>54</sup><https://www.w3.org/2005/Incubator/ssn/ssnx/qu/qu.owl>

<sup>55</sup><http://www.qudt.org/>

<sup>56</sup><https://code.ecoinformatics.org/code/semtools/trunk/dev/oboe/>

<sup>57</sup><http://www.ontology-of-units-of-measure.org/page/om-2>

<sup>58</sup><http://www.qudt.org/>

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 dissertation, this work has not yet been implemented in the main RDF stores.

### 3.1.4 KDD ontologies

This thesis is aimed at assisting data analysts through the different phases of a KDD process. Therefore, it is of interest to review existing ontologies representing the KDD process as a whole, KDD phases, or similar processes.

#### 3.1.4.1 OntoDM

The Ontology of Data Mining (OntoDM<sup>59</sup>) [82] aims to provide a structured vocabulary of entities to describe the data mining domain. It focuses on the definition of the following set of entities: datasets, datatypes, data mining tasks, generalizations, data mining algorithms, algorithms' components, and constraints. These basic entities are the resources to define more complex entities that may appear in data mining applications. For example, the proposed entities could be used for the formalization and description of KDD scenarios.

OntoDM is not designed to support a specific data mining use case. Instead, it is designed as a general-purpose ontology and it can be used to support different data mining activities that range from services to workflows. Although being a general-purpose ontology, OntoDM is a rather heavyweight ontology, representing over 800 classes.

---

<sup>59</sup><http://kt.ijs.si/panovp/OntoDM/OntoDM.owl>

#### 3.1.4.2 DM<sup>3</sup> ontology

The DM<sup>3</sup> ontology<sup>60</sup> [83] is an ontology that serves as a user-centric semantic model for DM model selection and reuse. The ontology is based on the CRISP-DM model, which is an alternative to the KDD process, and DM<sup>3</sup>'s core concepts and relations emphasize on data mining model management capabilities.

This ontology's two main classes are guided by these two concepts: data mining goals and data mining models. The former concept is captured representing the purpose (*DMPurpose*), object (*DMObject*) and focus (*ModelSelectionCriteria*). As for the data mining models concept, the *DMMModel* class is modelled based on the existing DM techniques.

#### 3.1.4.3 DQM Ontology

The DQM (Data Quality Management) ontology<sup>61</sup> [84] provides a structured representation of data quality management aspects in Semantic Web architectures. This ontology enables the suggestion of corrective actions for invalid data (via the class *dqm:DataCleansingRule* and its subclasses), the assesment of data quality (with the *dqm:DataQualityScore* and its subclasses) or the identification of data requirement violations (with the *dqm:DataRequirementViolation* class and its subclasses) among others.

Although not being an ontology focused on data mining like the previous two, it covers the Data Preprocessing phase of the KDD process, so that it is worth being mentioned in this section.

### 3.1.5 Other related domain ontologies

Ontologies covering main areas of discourse of this thesis were already reviewed. Although they do not exactly span the main areas of discourse, there are also other related domains which are worth being considered. In this section, some ontologies related to human comfort and anomaly detection are reviewed.

#### 3.1.5.1 HBC Ontology

The HBC (Human Comfort in Building) ontology<sup>62</sup> [85] formally describes human experiences of the IEQ (Indoor Environmental Quality) dimensions in building spaces. These experiences contained in the Hex ontology module<sup>63</sup> are defined as good, neutral or bad, depending on the user perception of thermal comfort (cold or warm), visual comfort (bright or dim), acoustic comfort

---

<sup>60</sup><http://128.172.188.35:8080/webprotege>, not available at the moment of writing this dissertation.

<sup>61</sup><http://purl.org/dqm-vocabulary/v1/dqm>

<sup>62</sup><https://w3id.org/ibp/hbc>

<sup>63</sup><https://w3id.org/ibp/hbc/hex>

(loud or quiet) and indoor air quality (positive, neutral or negative). Furthermore, there is a categorisation of building space types in the bim4Hex ontology module<sup>64</sup>, representing spaces according to the building objects they have, such as *bim4hex:SpaceWithHeater* or *bim4hex:SpaceWithoutWindow*. Finally, this ontology-based approach representation can be used for inferring and retrieving rooms with a certain quality (e.g. rooms with a low level of noise) or even for suggesting actions to reach a certain level of comfort in a given room.

### 3.1.5.2 ThinkHome Actor Ontology

The Actor ontology<sup>65</sup> is a module of the ThinkHome ontology (reviewed in section 3.1.1.5) and describes user information and preferences for a Smart Home System. It describes humans that interact with smart home systems in terms of age, gender and their level of satisfaction with the performance of the system. Other comfort parameters which describe the user preference are also represented in the ontology, such as the preference schedule and values related to air flow velocity, temperature or relative humidity. This knowledge representation allows the ThinkHome system to infer implicit knowledge from the description of a newly integrated element. For example, after adding a new human actor to the system, the reasoning mechanism can deduce appropriate default comfort parameters according to his/her age and gender. This ensures an adequate system behaviour from the start, even if new or unknown components are introduced.

### 3.1.5.3 FMECA ontology

The FMECA (Failure Mode, Effects and Criticality Analysis) ontology<sup>66</sup> [86] captures knowledge related to anomalies and faults that can happen in wind turbines. The ontology has two main classes: failure modes and equipment components. Failure causes and effects are defined as subclasses of the former concept. With regards to the latter, it defines subclasses such as devices, systems, sub-assemblies and parts. Instances of all these classes use serial numbers to distinguish from one another.

### 3.1.5.4 Folio Ontology

The Folio ontology<sup>67</sup> [87] captures concepts that occur within FMEA (Failure Mode and Effect Analysis), FTA (Fault Tree Analysis) documents and anomaly detection methods. The central class of the ontology is *AnomalyKnowledge*, which describes an anomaly. The causes and effects of a given anomaly can be related to the failure causes and effects, and all of them are related to anomaly detection methods and a degree of severity.

<sup>64</sup><https://w3id.org/ibp/hbc/bim4hex>

<sup>65</sup><https://www.auto.tuwien.ac.at/downloads/thinkhome/ontology/ActorOntology.owl>

<sup>66</sup>At the moment of writing this dissertation, the ontology is not publicly available.

<sup>67</sup><https://github.com/IBCNservices/Folio-Ontology/blob/master/Folio.owl>



In this thesis, the EEP SA ontology<sup>68</sup> is proposed in Chapter 4 for the semantic annotation of the addressed problem’s relevant data. This ontology, which is developed considering reviewed ontologies, is the cornerstone of the EEP SA data analyst assistant and captures the necessary domain and expert knowledge for the different KDD phases.

## 3.2 Semantic Technologies in Data Selection

This is the first phase of a typical KDD process where a dataset, a subset of variables or data samples on which discovery will be performed are selected. This data selection task is important, as data fragments containing relevant hidden knowledge may end up being excluded from the KDD process. To avoid this, the data analyst has to understand the data itself: which is the knowledge that represents and which is the additional knowledge that can be extracted from it. However, this is often not trivial and in most cases, a domain-specific knowledge is needed.

Data visualization and exploration methods may facilitate the understanding of data. In this regard, Dadzie and Rowe [88] provide an extensive survey of current efforts in the Semantic Web community to visualise LD in a coherent and legible manner, allowing non-domain and non-technical audiences to obtain a good understanding of its structure, and therefore implicitly compose queries, identify links between resources and intuitively discover new pieces of information. However, there is a lack of support of 3D data (which is fundamental in many scientific fields) by the analysed data visualization tools.

More classical approaches such as the attribute relevance analysis may also be useful for the Data Selection phase. Such approaches attempt to identify the highly relevant attributes and remove the irrelevant ones from a given dataset, for further analysis [89]. The relevance or significance of each attribute is evaluated according to the target variable (i.e. the variable to be predicted). However, the performance of these methods may be affected by the vast amount and heterogeneity of data that data analysts may face nowadays.

In the Data Selection phase, this thesis leverages Semantic Technologies to assist data analysts discovering which are the variables that should be taken into account in order to make accurate predictive models (section 5.2). This is a different approach compared to existing work in this KDD phase, which focus on visualizing data (e.g. data visualization tools) and cannot suggest new relevant variables that are not present in current datasets (e.g. relevance analysis).

---

<sup>68</sup><https://w3id.org/eeepsa>

### 3.3 Semantic Technologies in Preprocessing

Poor data quality has far-reaching effects and consequences. It has a direct impact on organizational success as it is the primary reason for 40% of all initiatives failing to achieve their targeted benefits, and it affects overall labour productivity as much as 20% [90]. Furthermore, as more business processes become automated, data quality is turning into the rate limiting factor for overall process quality.

Owing to the economic challenges and budget limitations that most organizations face, there is a dire need to eradicate quality issues in data as a way to minimize costs and increase efficiency. Data quality is a multidimensional concept, so that its definition may vary depending on the viewpoint. However, most definitions share the assumption that data quality is relative to formally or informally defined quality expectations such as consumer expectations and intentions, specifications, or requirements imposed by the usage of data [91]. In the context of this thesis, data quality is understood as the degree to which data fulfils requirements, while data quality issues are the direct effect of the violation of these requirements.

Today's real-world datasets are highly susceptible to noisy, missing, and inconsistent data due to their typically big size and their likely origin from multiple, heterogeneous sources [89]. These factors influence directly in the data quality and low quality data will lead to low quality mining results. There are many existing data preprocessing techniques including the ones that detect outliers or handle missing data fields. These techniques are not mutually exclusive, on the contrary, many different techniques may be applied together to improve the quality of the data.

#### 3.3.1 Outlier detection

Outliers are data objects that stand out amongst other data objects and do not conform to the expected behaviour in a dataset [92]. Furthermore, outliers can affect data quality, hindering the knowledge extraction process and leading to inaccurate or even wrong conclusions. Therefore, the process of finding these data objects, which is known as outlier detection, is an essential task for a wide range of domains including intrusion detection for cyber-security, fault detection in safety critical systems, fraud detection for credit cards and data monitoring in WSNs. Depending on the application's goal, there may be different ways of handling these outliers. For example, in a data analysis application, outliers may be filtered out to avoid unnecessary noise, while in fraud detection applications, detected outliers may be isolated and analysed, as they may represent potential frauds.

The outlier detection process has been widely researched for many years from statistics, geometry or machine learning communities. As a consequence, there are many different outlier detection methods. Further information regarding these methods can be found in different surveys [93, 94].

Outliers can occur for various reasons and understanding their provenance helps to determine how to handle them. However, identifying the potential cause of outliers still remains an unsolved challenge in most cases and discovering this cause may become an arduous process. Moreover, there are also challenging domains like the WSNs, where there are several factors like resource constraints (e.g. limited battery power or computational capacity), effects of harsh and unattended environments, or even malicious attacks, making the data generated by sensor nodes prone to outliers [95]. Even more, there are also scenarios where a data object may be considered an outlier in a given context (e.g. an observation of 40°C made by a temperature sensor located in the north of Spain during a winter day), but a usual data object in another different context (e.g. an observation of 40°C made by a temperature sensor located in the south of Spain during a summer day). In these cases, the application of conventional outlier detection techniques may produce skewed results. Despite the vast amount of existing data preprocessing methods, the data preprocessing remains an active area of research on account of the low quality of the existing data.

There is several work where outlier detection methods are applied to LD. Wienand and Paulheim [96] apply unsupervised numerical outlier detection methods to identify wrong statements (namely numerical outliers) in DBpedia. Moreover, Paulheim [97] focuses on the detection of wrong links between LD by means of different multidimensional outlier detection methods. As for Kontokostas et al. [98], data quality problems are formalized in the form of SPARQL query templates, and a pattern-based approach for data quality testing of RDF knowledge bases is proposed. The tool TripleCheckMate [99] is a tool for crowdsourcing the quality assessment of LD. The user selects a set of resources, and then evaluates the triples related to those resources. For each triple, the user determines if it has a data quality problem or not. In the case it has a problem, the user can also define that problem from a data quality problem taxonomy. There is also a survey where existing approaches for measuring the quality of LD are reviewed [100]. Furthermore, the common terminology used across the reviewed papers are formalized, and a list of 18 LD quality dimensions and 69 metrics is provided. Fürber et al. present a set of work [91, 101, 102] where data quality problems in Semantic Web data (e.g. missing and illegal values or functional dependency violations) are identified by means of data validation and SPIN rules.

In addition, the Preprocessing phase can also benefit from constraints represented in ontologies to perform data validity checks as well as to guiding users through data cleaning tasks. Khasawneh and Chan [103] take leverage of a domain ontology for mapping a user browsing sequence into sessions, allowing the identification of tasks or activities associated with different sessions of the same user. These mappings are relevant later on in a data cleaning process to remove data that is irrelevant for the user identification process. Another proposal in this area is OntoClean [104], an ontology-based approach to data cleaning. This approach takes leverage of an ontology, first of all to identify both the cleaning problem and the relevant data. Then, the user goals are translated into queries, and after a reasoning process, the potential suitable methods for meeting these goals are specified. Afterwards, the selected data cleaning algorithm is applied to the selected dataset, and the results of the cleaning process and a corresponding explanation are shown. Authors of this approach state that incorporating

domain ontologies and task ontologies in data cleaning algorithms can enhance the quality of the cleaning. The *OntoDataClean* [105] system on the contrary, integrates data and guides the data cleaning process in distributed environments. The system leverages an ontology that captures information about the sources to be preprocessed and the transformation tasks. Wang and Yang [106] present a domain ontology which supports the outlier detection in short documents, based on a density-based outlier detection method. As for the *DQM-ontology* [107] which captures data quality management knowledge, it is used for data structuring and to provide correction suggestions for invalid data, identify duplicates, and to store data quality annotations at schema and instance level. Preece et al. [108] describe a framework for managing information quality in an e-science context, where users state their quality requirements making use of a domain ontology's concepts.

In the domain of WSNs, Gao et al. [109] detect segment outliers and unusual events by combining statistical analysis and domain expert knowledge captured via an SSN-based ontology and semantic inference rules. This approach determines whether the sensor collects suspicious data or not by calculating its similarity with neighbours. However, it may not be applicable to isolated nodes where there are no nearby sensors to compare its similarity. Moreover, the system presented by Steenwinckel et al. [87], semi-automatically generates ontologies and SWRL rules based on the information collected in the FMEA and FTA documents. Afterwards, this knowledge is used both to annotate and reasoning over the observations. To the extent of our knowledge, these proposals are the only works where Semantic Technologies have a direct role in outlier detection methods.

In the outlier detection task, this thesis focuses on assisting data analysts towards the detection of outliers. Although Gao et al. [109] proposed a combination of statistical analysis and ontology usage to detect outliers in WSNs, it was found that its dependency with nearby sensors may hinder the usage of their method in isolated nodes. Therefore, in the *SemOD* framework proposed in this section (section 5.3.1) Semantic Technologies are exploited to annotate the context of sensors and observations and determine the existence of outliers. Furthermore, unlike in existing work, analysts are also guided to detect the cause of those outliers, which may be helpful to make decisions beyond predictive modelling (e.g. to decide the relocation of an existing sensor registering those outliers). All that, with a set of resources that abstract the data analyst from the underlying semantic technologies, so that neither a domain knowledge nor semantic technologies expertise is required.

### 3.3.2 Missing values

Missing Values are data quality problems that occur when values are empty or null in attributes where a value should have been recorded [110]. They are an issue affecting almost every type of real world datasets, and they are specially recurrent in datasets derived from WSNs due to their proneness to generate inconsistent and unreliable data [95, 111, 112].

With regard to analyzing missing values, different categories can be identified, usually differentiated by the reason that caused them. Each of these originators can produce different patterns on the data that goes lost. The most common three patterns are [110]:

- Missing Completely At Random (MCAR): When there is no identifiable pattern to describe the missing values.
- Missing At Random (MAR): When there is a pattern that relates an observed variable and the missing values.
- Missing Not At Random (MNAR): When a pattern exists, but it cannot be associated with any observed values.

In order to illustrate these missing values patterns, let us consider the following scenario where the mean income of a certain population is estimated via questionnaires. For different reasons, some income measures are missing. When some questionnaires are lost by chance, missingness is completely random, and it would be categorized as MCAR. When missingness is random within subgroups of other observed variables, missingness is MAR. For example, supposing that data on the profession of the subject is also collected and that managers are more likely not to share their income, then, within subgroups of profession, missingness is random. When the reason for missingness depends on missing values themselves, missingness is MNAR. For example, this happens when people don't want to share their income when it is below a certain amount because they don't feel comfortable with it.

When data analysis tasks are applied upon these datasets with missing values, obtained results are not as accurate as they could be, and they can even lead to wrong conclusions. Furthermore, several algorithms that try to extract patterns from data cannot process datasets with missing values. This fact creates a strong necessity of methods that can restore the incomplete pieces of data properly so that they are valid inputs to mentioned algorithms.

Different methods for handling missing values can be found in the literature [110, 113, 114, 115, 116]. A straightforward way of dealing with missing data is the deletion of incomplete observations or variables. This method is effective in some cases, such as when the quantity of incomplete observations or variables is low with respect to the dimension of the data, and when independence of the observations can be assumed. When this is not the case, however, the deletion strategy becomes a bad choice. An example of data where observations are not independent of each other are time series, which require a special treatment when they are incomplete. The approach commonly followed in this case is an imputation method, which consists in replacing missing data with substituted values.

Even though it has been proved that data quality is a relevant aspect for process quality and organizational success, it has not received sufficient attention from the Semantic Web Community. Egami et al. [117] estimate the temporal missing data from the LOD source containing the distribution of illegally parked

bicycles in Tokyo, taking leverage of a hybrid method using computational fluid dynamics and data coming from DBpedia Japanese<sup>69</sup>. As for the Mannheim Search Join (MSJ) Engine [118], it retrieves data from multiple sources to extend a local table with additional attributes. Although not being the goal of the approach, the discovered data can be used to fill the missing values in the table.

This thesis tries to raise awareness of the potential of Semantic Technologies in the handling of missing values. To do so, a set of some experiments is performed in section 6.1.3, which leads to future research lines in this regard.

### 3.4 Semantic Technologies in Transformation

This KDD phase spans different methods and tasks to project the data into a form in which data mining algorithms can work to extract the hidden knowledge. These set of methods and tasks can alter the data space dimensionality by enlarging it (e.g. with feature generation tasks), reducing it (e.g. with space embedding methods) or even acting in either direction (e.g. with the extraction of local features) [119].

Nowadays with the advent of LOD, third-party repositories are a valuable source of knowledge that can be incorporated to the set of data available, by creating additional features. Augmenting a dataset with features taken from LOD can contribute to the improvement of the results obtained in a KDD process. The LIDDM (Linked Data Data Miner) system [120] allows retrieving LD data from multiple SPARQL endpoints by writing SPARQL queries, and integrating this data after applying some filtering and segmentation tasks. Afterwards, the system enables the application of classification, clustering or association rules as part of a data mining process. Collecting and integrating large amounts of background knowledge can become an arduous and time-consuming task. This is why, unsupervised or (semi)automatic feature generation tasks have been proposed. FeGeLOD (FEature GEneration from Linked Open Data) [121] is an open source toolkit, which automatically creates data mining features from LOD. It consists of three phases: the entity recognition where raw data is mapped to the corresponding DBpedia URIs, the feature generation where properties and values related to those URIs are extracted, and the feature selection which discard the less relevant features. The RapidMiner Linked Open Data extension [122] is the descendent of FeGeLOD. This extension of the Rapidminer data analysis platform, offers operators for accessing LOD and gathering additional background knowledge from DBpedia such as direct types and categories. The framework presented by Cheng et al. [123] enables the construction of semantic features from a given knowledge base organized as a triple store. The framework leverages YAGO as the knowledge base from which features are retrieved.

In the Transformation phase, this thesis leverages domain-specific (Linked) Open Data repositories to generate new features that may contribute to develop more accurate predictive models (section 5.4). More specifically, it proposes

---

<sup>69</sup><http://ja.dbpedia.org>

making use of meteorological Open Data repositories. To do so, it proposes a process that extracts, semantically annotates and stores weather stations' data, as well as two parameterizable SPARQL queries to access this information.

## 3.5 Semantic Technologies in Data Mining

This phase is where artificial intelligence methods such as machine learning algorithms are applied to extract insight and knowledge from data. Depending on the final goal of the analysis, different data mining techniques may be necessary. Some of these techniques are:

- Classification: predicts the label or class for a given unlabelled point.
- Regression: predicts the numeric value of a given point.
- Clustering: partitions the points into natural groups called clusters, such that points within a group are very similar, whereas points across clusters are as dissimilar as possible.
- Frequent pattern mining: extracts informative patterns from datasets. Patterns comprise sets of co-occurring attribute values, called itemsets, or more complex patterns, such as sequences, which consider explicit precedence relationships (either positional or temporal), and graphs, which consider arbitrary relationships between points.

Once the data mining technique that best fits with the final goal is chosen, the suitable data mining algorithm has to be selected. Furthermore, the parameters of the algorithm need to be adequately adjusted to ensure its good performance and enabling the generation of accurate predictions.

To the extent of our knowledge, there are no approaches that incorporate semantics into data mining algorithms to directly influence their results. This could be caused because performance of algorithms is more dependent on data, so that semantics have little room for improving them. There are other statistical approaches such as Intelligent Discovery Assistants (IDA) which assist users select and parametrize algorithms based on available data.

In this thesis, this is the KDD phase where assistance through semantics has been left as future work. Therefore, this Data Mining phase improvement relies on the data enriched in previous phases.

## 3.6 Semantic Technologies in Interpretation

The interpretation is the final phase of a KDD process where the knowledge is extracted from data, by discovering hidden patterns from the results obtained in

the KDD Data Mining phase. Once the knowledge is at hand, it can be employed in a decision-making process. However, if this knowledge is not significant or reasonable enough, it can involve returning to any of the previous KDD steps for further iteration.

Usually, these results are interpreted by humans who use their expertise in possibly different domains. However, nowadays there is a shortage of people with analytical skills to interpret data [124] and even for expert data analysts without such domain knowledge it may not be easy to adequately understand and interpret those results [125]. Furthermore, even for a domain expert, obtaining a complete and satisfactory explanation may become a tedious and time-consuming process, and part of the knowledge can still remain unrevealed or unexplained. This could result in making decisions far from optimal, with all the associated risks this entails.

This has motivated the dedication of some research efforts to bridge the semantic gap between users and the results obtained after applying different data mining techniques. The Explain-a-LOD toolkit [126] makes use of LOD (e.g. DBpedia and Eurostat) as background knowledge to generate hypothesis for interpreting statistics. Furthermore, this background knowledge is also exploited for generating visualizations that may also contribute to the interpretation of these statistics. Tiddi [127] aims at using background knowledge found in the LD to explain patterns and regularities in data. To do so, additional information is extracted from LD, generating hypotheses, and evaluating them according to different ranking strategies. With regards to subgroup discovery methods<sup>70</sup>, Vavpetič et al. [128] propose a methodology for explaining subgroups or sets of instances, using higher-level ontological concepts. Once the subgroups of instances are identified, they are characterized using an ontological concept, giving insight into the differences between a given subgroup and the remaining data. Clustering data mining methods, which have similarities with subgroup discovery methods, also received attention from the semantic web community. Dedalo [129] is a framework which enables the exploitation of external data to generate explanations of results of clustering techniques. The framework traverses LD with different strategies such as heuristic scoring measures of the properties to inspect, in order to find the best explanation items of a cluster. Another data mining technique's result interpretation is tackled by d'Aquin and Jay [130]. Specifically, this data mining techniques is the sequential pattern extraction. Towards this goal, authors present a method that exploits available LD through the automatic building of a navigation exploration structure of results, based on data dimensions chosen by the data analyst. Svátek et al. [131] propose that given some previously created mappings between data and ontologies, some discovered associations can be matched with semantic relations or their more complex chains from the ontology. This semantic relation represents a potential explanation for the discovered association. According to Dou et al. [132], data mining results and discovered patterns should be presented in a formal and structured format, so that they are capable to be interpreted as domain knowledge. Encoding these results in the formal structure of resources like ontologies could in turn enable other processes (e.g. decision-making) to take leverage of current results.

---

<sup>70</sup>It can be defined as the extraction of interesting subgroups for a target value.



In the Interpretation phase, this thesis proposes EROSO in section 5.6. Unlike most of reviewed work, EROSO does not leverage LD to interpret results, but instead, it exploits expert knowledge captured in the form of ontology-driven rules and queries. Furthermore, it focuses on the interpretation of predictive models' results which, to the extent of our knowledge, at the moment of writing this dissertation still remains untackled.



## Chapter 4

# The EEP SA Ontology

Towards the incorporation of the Semantic Technologies in the EEP SA (Energy Efficiency Prediction Semantic Assistant) data analyst assistant, it is of utmost importance to rely on proper ontologies and vocabularies that codify the required knowledge and enables an adequate annotation of the data. Previous chapters 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 chapter describes the EEP SA ontology which is focused on energy efficiency and thermal comfort in tertiary buildings but it is aimed at being reusable and easily customizable for similar problems in different types of buildings. The latest version of the EEP SA ontology is available online in <https://w3id.org/eepsa>.

### 4.1 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 such as On-To-Knowledge [133], DILIGENT [134] or the NeOn Methodology [135] is advised. For the development of the EEP SA ontology, the NeOn Methodology was followed mainly because unlike other methodologies it does not prescribe a rigid workflow, but instead it suggests a variety of paths. The NeOn Methodology is a scenario-based methodology supporting different aspects of the ontology development process: from the reuse of existing resources, to the dynamic evolution of ontologies in distributed environments where knowledge is introduced by different people at different stages. Furthermore, the proposed scenarios are decomposed into different activities which can be combined in a flexible manner towards the achievement of the expected goal. Specifically, these are the nine scenarios defined in the NeOn Methodology:

- Scenario 1: From specification to implementation, where the requirements

the ontology should fulfil are specified.

- Scenario 2: Reusing and re-engineering non-ontological resources, where existing non-ontological resources are searched, re-engineered and reused.
- Scenario 3: Reusing ontological resources, where existing ontological resources are searched and reused.
- Scenario 4: Reusing and re-engineering ontological resources, where existing ontological resources are searched, re-engineered and reused.
- Scenario 5: Reusing and merging ontological resources, where a new ontological resource is created from two or more existing ontological resources.
- Scenario 6: Reusing, merging and re-engineering ontological resources, where a new ontological resource is created from two or more existing re-engineered ontological resources.
- Scenario 7: Reusing ontology design patterns, where ontology design patterns are reused.
- Scenario 8: Restructuring ontological resources, where ontological resources are restructured (e.g. modularized or extended) and integrated in the ontology.
- Scenario 9: Localizing ontological resources, where ontologies are adapted to other languages and culture communities.

In the EEPsA ontology's development the following set of scenarios defined by the NeOn Methodology were applied. First of all, the scenario 1 was applied to collect the ontology requirements and moreover, it served as a main workflow where the results of other scenarios were integrated. Then, scenario 7 was applied to define the basic building blocks in the form of ODPs on top of which the ontology was going to be implemented. Finally, scenarios 3 and 4 were applied to decide the ontologies to be reused and re-engineered prior to their reuse. The application of the other scenarios was not considered necessary. An overview of these scenarios is presented next.

#### 4.1.1 Scenario 1

This scenario comprises core activities that need to be performed in any ontology development. First of all, the ontology requirements specification activity is performed to create the Ontology Requirements Specification Document (ORSD) [136]. This document includes among others, the ontology purpose, its intended uses, and the set of ontology requirements mainly in the form of Competency Questions (CQs).

Furthermore, this scenario 1 may also involve the selection of tools used to develop the final ontology, as well as the selection of tools and technologies to manage the different versions of the ontology.

### 4.1.2 Scenario 7

In this scenario, ODP repositories (e.g. [OntologyDesignPatterns.org](http://ontologydesignpatterns.org)<sup>1</sup>) are accessed to find patterns to be reused in the ontology being developed. The application of this scenario for the EEP SA ontology is discussed in section 4.3.

### 4.1.3 Scenarios 3 and 4

The reuse of ontological resources built by others that have already reached some degree of consensus is good practice in ontology development processes [140]. According to W3C's Data on the Web Best practices [141], the reuse of an existing vocabulary not only captures and facilitates consensus in communities, but also increases interoperability and reduces redundancies. Furthermore, this practice brings other important benefits:

- It increases the quality of the applications reusing ontologies, as these applications become interoperable and they are provided with a deeper, machine-processable and commonly agreed-upon understanding of the underlying domain of interest.
- It reduces the costs related to ontology development because it avoids the reimplementing of ontological components, which are already available on the Web and can be directly (or after some additional customization tasks) integrated into a target ontology.
- It potentially improves the quality of the reused ontologies, as these are continuously revised and evaluated by various parties through reuse.

In this scenario, the Ontological Resource Reuse Process [142] is proposed as an activity to perform the reuse of existing ontological resources. This process is a necessary first step for scenarios 3, 4 and 5 of the NeOn Methodology, and it comprises the following activities:

1. **Ontology Search.** This activity consists in finding appropriate ontological resources that meet the requirements described in the ORSD. The existing ontology catalogues such as LOV [143] or LOV4IoT [57] (specialized in ontologies related to IoT) can ease this task [144].
2. **Ontology Assessment.** This activity deals with assessing the usability of an ontology with respect to the requirements previously defined in the ORSD. This may end up being a laborious task due to the different criteria that may make ontologies suitable for a certain use case. Furthermore, the frequent scarce documentation of ontologies may hinder this activity.
3. **Ontology Comparison.** In this activity, assessed ontologies should be compared according to criteria that encompass the content of the ontology, the

---

<sup>1</sup><http://ontologydesignpatterns.org>

organization of these contents, the language in which it is implemented, the methodology that has been followed to develop it, the software tools used to build and edit the ontology, and the costs of the ontology [145].

4. Ontology Selection. After assessing and comparing ontologies, the most appropriate one or ones (preferably standardized ones) have to be selected in order to reuse them by integrating them in the new ontology being developed.

In the case of scenario 4, ontological resources to be reused need to be previously re-engineered to serve to the intended purpose or problem.

## 4.2 The EEPISA Ontology Scope

The EEPISA ontology's ORSD resulting from applying NeOn Methodology's scenario 1, defines 67 CQs, represents the most frequent terminology in the problem at hand (e.g. actuator or feature of interest) and a CamelCase naming convention is advised. A more detailed description of the EEPISA ontology's ORSD is shown in Appendix B.

Among the available software for building and maintaining ontologies (e.g. PoolParty<sup>2</sup> [137] or TopBraid Composer<sup>3</sup>), Protégé<sup>4</sup> [138] was chosen. Protégé exists in a variety of frameworks (e.g. desktop system or web-based), and in this thesis the Protégé desktop version 5.1.0<sup>5</sup> was used. As for managing the different versions of the ontology, a version control system was necessary. A version control system records changes to a file or set of files over time so that specific versions can be retrieved later on [139]. The development of the EEPISA ontology was managed with a Git repository.

## 4.3 Developing the EEPISA 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 [146]. 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,

<sup>2</sup><https://www.poolparty.biz/>

<sup>3</sup><https://www.topquadrant.com/tools/modeling-topbraid-composer-standard-edition/>

<sup>4</sup><https://protege.stanford.edu/>

<sup>5</sup><https://github.com/protegeproject/protege-distribution/releases/tag/v5.1.0>

span more than one application area or domain, address a single invariant instead of targeting multiple recurring issues at the same time, follow established modelling best practices, and so forth [147].

Developing the EEPISA ontology on top of ODPs was found a suitable option due to the great flexibility provided by this modelling solution, which allows a proper segmentation of the intended conceptualization. As a matter of fact, the NeOn Methodology’s scenario 7 was applied for this purpose.

Taking into consideration the 67 CQs identified in the OSRD shown in Appendix B, a list of 14 CQs that summarize the basic requirements for assisting data analysts in certain recurrent IoT-related problems was created. More specifically, the following CQ list addresses problems related with 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 development of a set of core ODPs that satisfies the following CQ list 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/actuation?
- CQ12: When was an observation/actuation generated?
- CQ13: For what time interval or instant is valid an actuation/observation?
- CQ14: For what spatial location is valid an observation/actuation?

For each competency question  $CQ_n$ , a twin competency question  $CQ_n^i$  can be considered, which consists in rephrasing the question in the opposite direction. For example,  $CQ01^i$  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. These twin competency questions are present in this section in the examples provided for every ODP.

In this case, the considered CQs were divided in three subsets according to their domain coverage: {CQ01, CQ02, CQ03}, {CQ04, CQ05, CQ06, CQ07, CQ08, CQ09, CQ10} and {CQ11, CQ12, CQ13, CQ14}. In order to solve those subsets, an ODP was defined for each of them. The proposed ODPs are inspired by existing ontologies and ODPs which address the mentioned CQs in an inadequate manner.

Even though these ODPs are motivated by energy efficiency and thermal comfort problems in tertiary buildings, they are designed to be applicable to similar problems in other types of buildings. 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 [148], helps to ensure clarity in modelling and avoids errors that have unintended reasoning implications [62]. These alignments are kept in separate files and are available online in each ODP's documentation page.

Next, a brief review of related ODPs is presented, followed by the three proposed ODPs: the AffectedBy<sup>6</sup>, the EEP<sup>7</sup> (Execution-Executor-Procedure) [149] and the RC<sup>8</sup> (Result-Context) ODPs.

### 4.3.1 Related ODPs

The initial version of the SSN ontology<sup>9</sup> [59] was built around a central ODP called Stimulus-Sensor-Observation [60] (SSO) describing the relationship between sensors, stimulus and observations. The new version of the SSN ontology<sup>10</sup> follows a horizontal and vertical modularization architecture by including a lightweight but self-contained core ontology called SOSA<sup>11</sup> (Sensor, Observation, Sample, and Actuator) for its elementary classes and properties. Furthermore, similar to the original SSO patterns, SOSA acts as a central building block for the new SOSA/SSN ontology.

The Actuation-Actuator-Effect (AAE) ODP<sup>12</sup> intends to model the relationship between an actuator and the effect it has on its environment through actuations. This pattern adapts the SSN ontology's SSO ODP for actuators. The SOSA/SSN ontology covers the function of the AAE ODP for actuators by expanding the SSO pattern in the SOSA ontology.

The SOSA/SSN ontology does not provide enough constraints to the defini-

---

<sup>6</sup><https://w3id.org/affectedBy>

<sup>7</sup><https://w3id.org/eep>

<sup>8</sup><https://w3id.org/rc>

<sup>9</sup><http://purl.oclc.org/NET/ssnx/ssn>

<sup>10</sup><http://www.w3.org/ns/ssn/>

<sup>11</sup><http://www.w3.org/ns/sosa/>

<sup>12</sup><http://ontologydesignpatterns.org/wiki/Submissions:Actuation-Actuator-Effect>



tions of classes and properties to guarantee a proper answer to a question like: what is the feature of interest corresponding to a given property that has been observed by a sensor? And neither to this other question: which sensors observe a given property of a feature of interest?

The SmartEnv ontology<sup>13</sup>, proposed as a representational model to assist the development process of smart environments, is a network of 8 different ODPs [69]. These ODPs are used to modularize the proposed solution, while at the same time avoiding strong dependencies between the modules to manage the representational complexity of the ontology. The SmartEnv relies on the SOSA/SSN ontology without introducing enough constraints to solve the aforementioned weaknesses.

The SEAS Ontology<sup>14</sup> [64] is an ontology designed as a set of simple core ODPs that can be instantiated for multiple engineering related verticals. The SEAS Feature of Interest ontology, is one of the core modules that forms the SEAS ontology, and defines features of interest (*seas:FeatureOfInterest*) and properties (*seas:Property*). The Procedure Execution (PEP) ontology<sup>15</sup> 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.

The Observation ODP<sup>16</sup> aims at representing observations of things, under a set of parameters. This set of parameters may include the place where the observation was made, the time when it was made, and any other feature concerning the specific thing being observed.

The IoT Application Profile (IoT-AP) ontology<sup>17</sup> [150], is an ontology for representing and modelling the knowledge within the domain of the IoT. The ontology is designed re-using ODPs such as the aforementioned Observation and the time indexed situation<sup>18</sup>. It focuses on observations, but it also covers sensors that generate those observations, their values and observation collections. However, this ontology suffers from similar weaknesses to those previously commented about the SSN ontology. This is basically due to the lack of proper constraints on property definitions.

### 4.3.2 The AffectedBy ODP

Data analysts dealing with energy efficiency and thermal comfort problems in tertiary 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

<sup>13</sup><https://w3id.org/smartenvironment/smartenv.owl>

<sup>14</sup><https://w3id.org/seas/>

<sup>15</sup><https://w3id.org/pep/>

<sup>16</sup><http://ontologydesignpatterns.org/wiki/Submissions:Observation>

<sup>17</sup><http://stlab.istc.cnr.it/IoT-AP/IoT-AP.rdf>, not available at the moment of writing this dissertation.

<sup>18</sup><http://ontologydesignpatterns.org/wiki/Submissions:TimeIndexedSituation>

CQ03 (described in the CQ list presented in this section) 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 [61] 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<sup>19</sup>, 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<sup>20</sup> 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

<sup>19</sup><https://w3id.org/seas/FeatureOfInterestOntology>

<sup>20</sup><https://w3id.org/affectedBy>

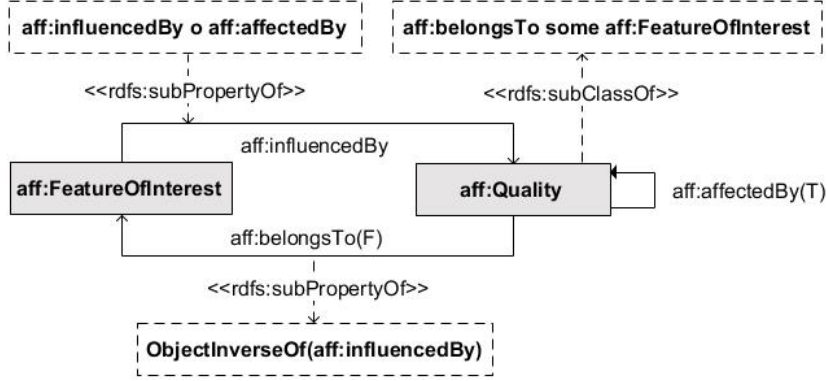


Figure 4.1: The AffectedBy ODP.

range, and it solves CQ03. Furthermore, the following axiom formalizes that every quality belongs to a feature of interest:

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

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 thesis and sometimes even inappropriate. For example, the temperature of individual *:room03* may derive from the occupancy of the room, but the room’s 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:

$$\text{seas:hasProperty} \circ \text{seas:derivesFrom} \sqsubseteq \text{seas:hasProperty} .$$

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.

For the purpose of solving CQ01, the object property *aff:influencedBy*<sup>21</sup> with *aff:FeatureOfInterest* as its domain and *aff:Quality* as its range is introduced,

<sup>21</sup>In the previous version of the AffectedBy ODP [149] this object property was named *aff:hasQuality*. However, it was renamed after *aff:influencedBy* to avoid misleading interpretations.

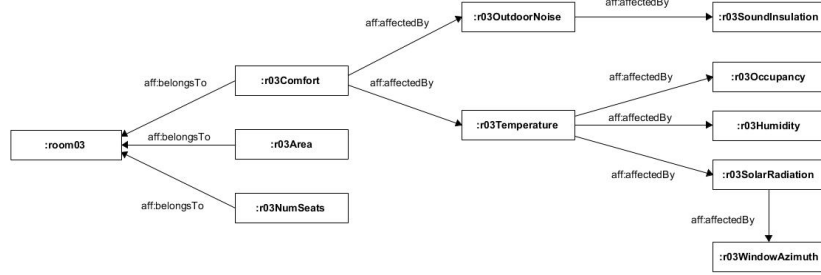


Figure 4.2: Triples using the AffectedBy ODP vocabulary.

alongside with the next property chain axiom:

$$aff:influencedBy \circ aff:affectedBy \sqsubseteq aff:influencedBy .$$

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 4.1. (F) represents a functional object property and (T) a transitive object property.

**AffectedBy ODP Example.** Figure 4.2 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*?  
 SELECT ?x  
 WHERE { :room03 aff:influencedBy ?x. }  
 Answer: *:r03Area, :r03NumSeats :r03Comfort, :r03Temperature, :r03OutdoorNoise, :r03Occupancy, :r03Humidity, :r03SolarRadiation, :r03SoundInsulation, :r03WindowAzimuth.*  
 (After inferences provided by axioms  $aff:influencedBy \circ aff:affectedBy \sqsubseteq aff:influencedBy$  and  $aff:belongsTo \sqsubseteq aff:influencedBy^{-1}$ ).
- (CQ01<sup>i</sup>): 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*  $\circ$  *aff:affectedBy*  $\sqsubseteq$  *aff:influencedBy* and *aff:belongsTo*  $\sqsubseteq$  *aff:influencedBy*<sup>-1</sup>).

- (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*.

**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<sup>22</sup>. These alignments are kept in separate files and are available online in the AffectedBy ODP's documentation page <https://w3id.org/affectedBy>.

### 4.3.3 The EEP ODP

Another interesting information for data analysts working on energy efficiency and thermal comfort problems in tertiary buildings could be addressed by competency questions CQ04, CQ05, CQ06, CQ07, CQ08, CQ09 and CQ10 (described in the CQ list presented in this section). These CQs are the requirements considered for the EEP (Execution-Executor-Procedure) ODP<sup>23</sup>.

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<sup>24</sup>, 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 (i.e. the RC ODP presented in section 4.3.4) 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.

<sup>22</sup><http://www.ontologydesignpatterns.org/ont/dul/DUL.owl>

<sup>23</sup><https://w3id.org/eep>

<sup>24</sup><https://w3id.org/seas/EvaluationOntology>

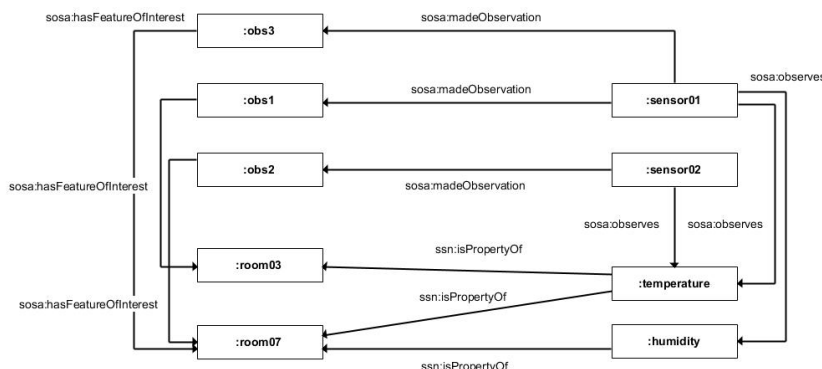


Figure 4.3: A SOSA/SSN annotated set of triples.

The aforementioned subset of CQs (CQ04 to CQ10) have been tackled by the SOSA/SSN ontology. However, a set of triples annotated with SOSA/SSN (for example the set shown in Figure 4.3) cannot properly solve a question like CQ10<sup>1</sup>: 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.
: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.

PEP ontology generalizes the core concepts of SOSA/SSN (i.e. Observation, Actuation, Sensor, Actuator, and Procedure). The proposed 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 4.4, where (F) represents a functional object property and (T) a transitive object property). An individual of *eep:Execution* is an event upon a quality of a feature of interest, produced by an agent by

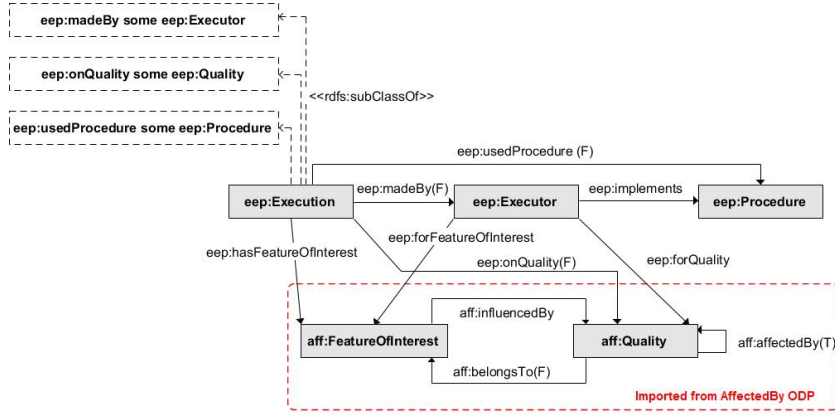


Figure 4.4: The Execution-Executor-Procedure (EEP) ODP.

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:

$$\begin{aligned}
 eep:Execution &\sqsubseteq \exists eep:madeBy.eep:Executor, \\
 eep:Execution &\sqsubseteq \exists eep:onQuality.eep:Quality, \text{ and} \\
 eep:Execution &\sqsubseteq \exists eep:usedProcedure.eep:Procedure
 \end{aligned}$$

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:

$$\begin{aligned}
 eep:madeBy^{-1} \circ eep:usedProcedure &\sqsubseteq eep:implements, \\
 eep:onQuality \circ eep:belongsTo &\sqsubseteq eep:hasFeatureOfInterest, \\
 eep:madeBy^{-1} \circ eep:onQuality &\sqsubseteq eep:forQuality, \text{ and}
 \end{aligned}$$

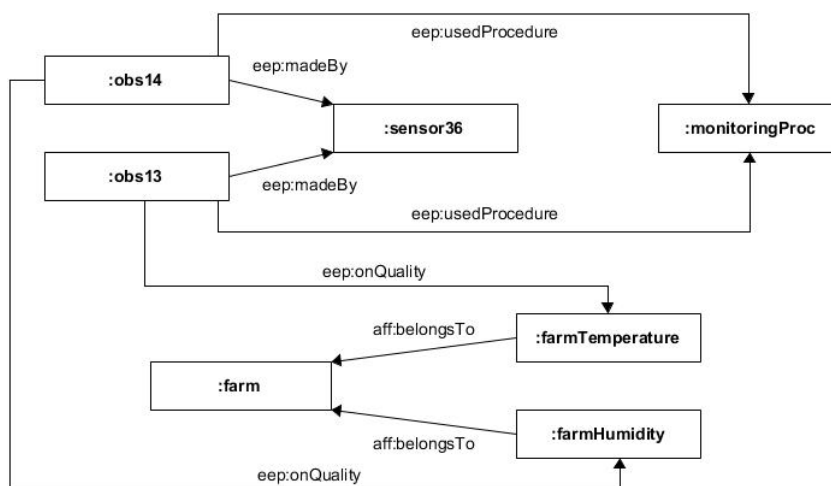


Figure 4.5: Triples using the EEP ODP vocabulary.

$eep:forQuality \circ eep:belongsTo \sqsubseteq eep:forFeatureOfInterest$  .

**EEP ODP Example.** Figure 4.5 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 *:farm* is in charge of measuring both farm’s temperature and humidity (i.e. *:farmTemperature* and *:farmHumidity*). Furthermore, this sensor implements a 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  
 $eep:madeBy^{-1} \circ eep:usedProcedure \sqsubseteq eep:implements$ ).

- (CQ07<sup>i</sup>): 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  
 $eep:onQuality \circ eep:belongsTo \sqsubseteq eep:hasFeatureOfInterest$ ).
- (CQ08): What are the qualities observed by the observation *:obs13?*  
 SELECT *?x*  
 WHERE {*:obs13 eep:onQuality ?x.*}  
 Answer: *:farmTemperature.*
- (CQ09<sup>i</sup>): 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  
 $eep:forQuality \circ eep:belongsTo \sqsubseteq eep:forFeatureOfInterest$  and  $eep:madeBy^{-1} \circ eep:onQuality \sqsubseteq eep:forQuality$ ).
- (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  $eep:madeBy^{-1} \circ eep:onQuality \sqsubseteq eep:forQuality$ ).

**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>.

#### 4.3.4 The RC ODP

Although the AffectedBy and EEP ODPs alleviate much of the data analysts' information needs, these data analysts 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 (described in the CQ list presented in this section).

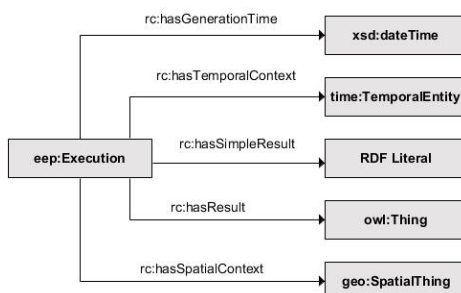


Figure 4.6: The Result-Context (RC) ODP.

Every ontology or ontology network covering observations or actuations need to take into account the representation of these actions' results. For example, the SOSA/SSN ontology uses the *sosa:hasResult* object property, the IoT Application Profile (IoT-AP) ontology [150] uses the *iotap:hasObservationValue* object property and om-lite uses the *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 example, 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<sup>25</sup> (shown in Figure 4.6), 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. 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.

<sup>25</sup><https://w3id.org/rc>

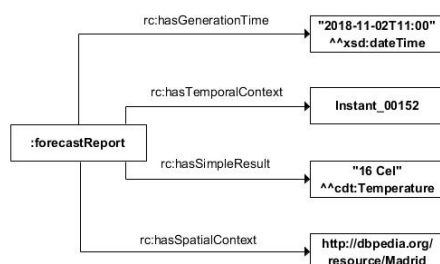


Figure 4.7: Triples using the RC ODP vocabulary.

With respect to the RC ODP, it defines two properties: on the one hand, *rc:hasGenerationTime* which is equivalent to *sosa:resultTime*, and on the other, *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* object property 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 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 datatype property *rc:hasGenerationTime*) and forecasts that there will be a temperature of 16°C (with the datatype property *rc:hasSimpleResult*) in Madrid (with the object property *rc:hasSpatialContext*) on 2018-11-03 at 16:00 (with the datatype property *rc:hasTemporalContext*). Figure 4.7 shows this instantiation example.

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.
```

**RC ODP Alignments.** The RC ODP is aligned with the SOSA/SSN and PROV-O<sup>26</sup> 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 EEP SA 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 EEP SA 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 (FoI4EEP SA, Q4EEP SA, P4EEP SA, EXR4EEP SA and EXN4EEP SA), and an ontology module containing expert knowledge (EK4-EEP SA).

## 4.4 Ontology Reuse Discussion

Following the W3C’s Data on the Web Best practices [141] which say that the reuse of existing ontological resources is good practice, the EEP SA ontology applied NeOn Methodology’s scenarios 3 and 4 to reuse existing vocabularies. Ontologies reviewed in section 3.1 were assessed with the requirements specified in the ORSD (shown in Appendix B) and compared with each other to select the ones to be reused (and previously re-engineered if needed). Three main areas

<sup>26</sup><https://www.w3.org/TR/prov-o/>

of discourse were considered to the application of scenarios 3 and 4: buildings and spaces (under the *eep:FeatureOfInterest* stub class), qualities or properties of features of interest (under the *eep:Quality* stub class), and sensors and actuators (under the *eep:Executor* stub class).

Ontologies like ifcOWL<sup>27</sup> 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 or 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<sup>28</sup>, ThinkHome<sup>29</sup>, ifcOWL and Brick<sup>30</sup>. Worse still, the lack of public access to ontologies, as it happens with EEOnt, makes them impossible to analyze or reuse.

A trend towards a pattern-based design tends to produce modular ontologies that are more understandable and more easily extended or re-engineered when necessary. The initial SSN ontology may be an example of this pattern-based design, and IoT-O ontology<sup>31</sup> and FIESTA-IoT ontology<sup>32</sup> may be considered extensions of such initial SSN. Moreover, when some undesirable design decisions on the original SSN were spotted, its re-engineering 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 example, SmartEnv and S3N<sup>33</sup> were developed as SOSA/SSN extensions. SEAS and BOT are other representative ontologies of this pattern-based design. Furthermore, SOSA/SSN, SEAS, and BOT<sup>34</sup> 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<sup>35</sup> can be considered an example of these

---

<sup>27</sup>[http://ifcowl.openbimstandards.org/IFC4\\_ADD2.owl](http://ifcowl.openbimstandards.org/IFC4_ADD2.owl)

<sup>28</sup><http://elite.polito.it/ontologies/dogont.owl>

<sup>29</sup><https://www.auto.tuwien.ac.at/downloads/thinkhome/ontology/>

<sup>30</sup><https://brickschema.org/>

<sup>31</sup><https://www.irit.fr/recherches/MELODI/ontologies/IoT-0>

<sup>32</sup><http://ontology.fiesta-iot.eu/ontologyDocs/fiesta-iot/doc>

<sup>33</sup><https://github.com/s3n-ontology/s3n/blob/master/s3n.ttl>

<sup>34</sup><https://w3id.org/bot>

<sup>35</sup><http://ontology.fiesta-iot.eu/ontologyDocs/fiesta-iot/doc>

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<sup>36</sup>, 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. These criteria have been taken into consideration when deciding which ontology to reuse in the EEP SA ontology.

## 4.5 The EEP SA Ontology Modules

The modularization of ontologies consists in partitioning them into independent self-contained knowledge components known as modules. A modular approach brings benefits such as flexibility for component reuse [151], support for more efficient query answering [152], and enhancement of components change and evolution [153].

When an existing ontology is large and monolithic, it needs to be splitted up in order to ease its maintenance and use. 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 [154].

In order to avoid performing ontology modularization techniques in the future, modularization is advised to be implemented from an early ontology development stage. This is why the EEP SA ontology is modularized by design. Each ontology module has few dependencies with others (as it is demonstrated in section 4.8) and following best practices, NeOn Methodology's scenarios 3 and 4 are applied in order to reuse existing resources as much as possible. Next, an overview of the EEP SA ontology modules is presented.

---

<sup>36</sup><http://ontology.tno.nl/saref>

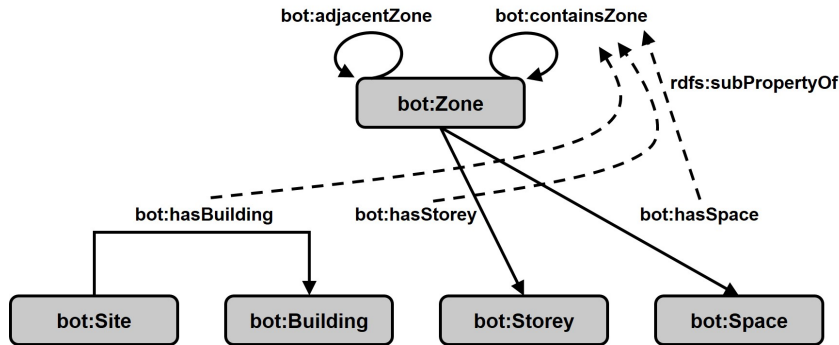


Figure 4.8: Overview of the main classes and properties defined in BOT.

#### 4.5.1 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 thesis, a feature of interest is understood as an abstraction of a real world phenomena (e.g. object and event). 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<sup>37</sup> tries to tackle CQs such as the following (the rest of the CQs tackled are available in Appendix B):

- 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 analysed in section 3.1.1, and finally BOT (shown in Figure 4.8) 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:

- CQ20: Which space does a given door belong to?
- CQ22: How many windows does a given space have?
- CQ23: Is a given window adjacent to outdoors?

<sup>37</sup><https://w3id.org/eeepsa/foi4eeepsa>

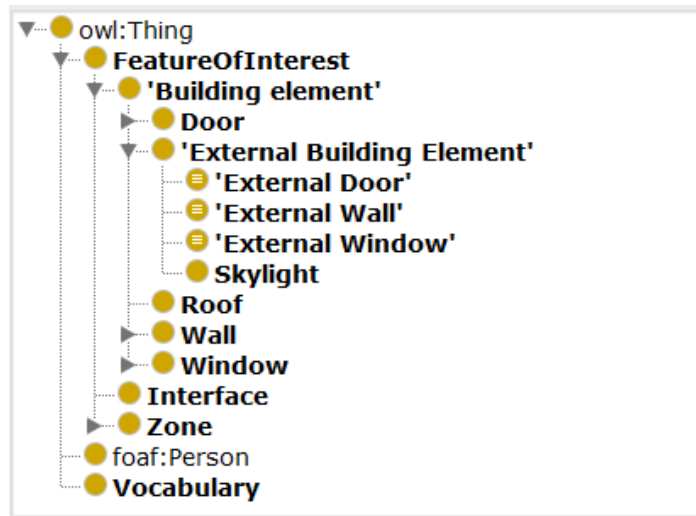


Figure 4.9: Overview of the classes defined in FoI4EPPSA.

To this end, the PRODUCT ontology<sup>38</sup> was considered. PRODUCT (which at the moment of writing this dissertation is still under development) has a much wider coverage scope than the needed, so its importation would result in increasing the 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 EEPsA ORSD such as doors (*foi4eepsa:Door*) and windows (*foi4eepsa:Window*) are defined. Furthermore, a class *foi4eepsa:ExternalBuildingElement* is 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 FoI4EPPSA are mapped to the related PRODUCT ontology terms. PRODUCT is in turn aligned with the IFC4 Addendum 2 standard, making the FoI4EPPSA ontology module interoperable.

Last but not least, information related to the building context is also an important aspect in the matter at hand. Namely, FoI4EPPSA has to solve the following CQs:

- CQ27: Which is the intended use of the building?
- CQ29: When was the building built?
- CQ30: 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 buildings and building-related contexts. However, the conceptualization of these properties in ifcOWL as instances of classes

<sup>38</sup><https://github.com/w3c-lbd-cg/product>



(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. [37], FoI4EEPSA defines a re-engineering of the relevant properties contained in IFC PSET Building Common and IFC PSET Building collections. For example, datatype property *foi4eepsa:hasYearOfConstruction* is used to represent the construction year of a building, and datatype property *foi4eepsa:hasMarketCategory* to define a building’s usage type (e.g. residential or commercial).

Figure 4.9 shows an overview of the main FoI4EEPSA classes.

#### 4.5.2 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<sup>39</sup> tries to tackle CQs such as the following (the rest of the CQs tackled are available in Appendix B):

- CQ31: Which are the actuatable qualities?
- CQ32: Which are the observable qualities?
- CQ33: 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 (which is followed by the EEPsA ontology), this class comprises qualities that can be observed, estimated and even forecasted. On the other hand, the qualities of a feature of interest that can be acted on, are 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 two classes. Likewise, 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 the *q4eepsa:MeteorologicalQuality* class, which are observable but not actuatable as defined with the following axiom:

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


---

<sup>39</sup><https://w3id.org/eepsa/q4eepsa>

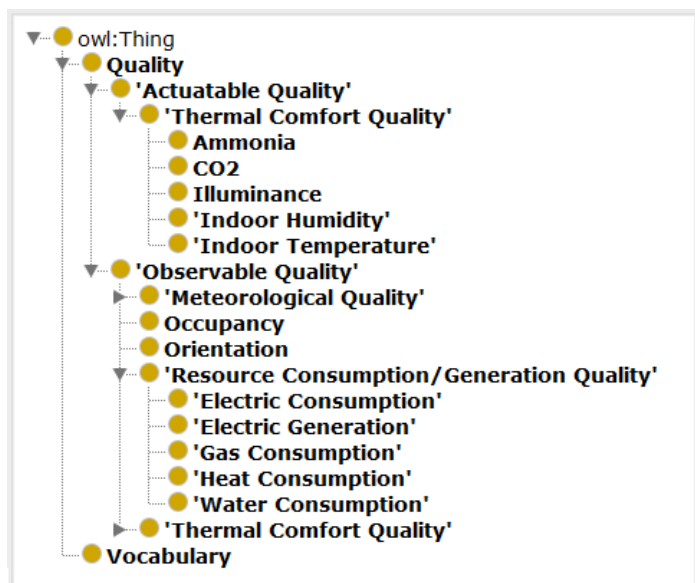


Figure 4.10: Overview of the classes defined in Q4EEPSA.

Qualities related with the thermal comfort within a space such as indoor temperature (*q4eeepsa:IndoorTemperature*) and indoor humidity (*q4eeepsa:IndoorHumidity*) are represented as subclasses of the *q4eeepsa:ThermalComfortQuality* class. These qualities can be observed and acted on. Furthermore, qualities related to the resource consumption/generation such as water consumption (*q4eeepsa:WaterConsumption*) or electric generation (*q4eeepsa:ElectricGeneration*), are also defined. These concepts are described as subclasses of the *q4eeepsa:ResourceConsumptionGenerationQuality* class, 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 subclass of the *q4eeepsa:ActuatableQuality* class. Some of the mentioned classes are re-engineered 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<sup>40</sup>.

Figure 4.10 shows an overview of the main Q4EEPSA classes.

### 4.5.3 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.

<sup>40</sup><https://w3id.org/seas/GenericPropertyOntology>

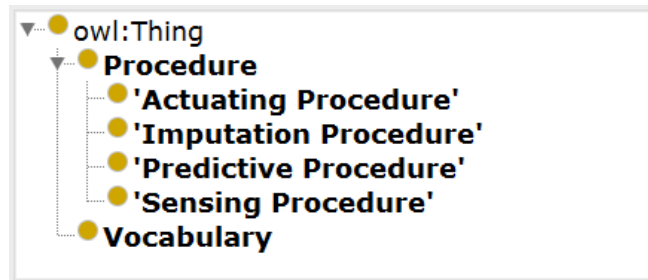


Figure 4.11: Overview of the classes defined in P4EEPSA.

In particular, the P4EEPSA ontology module<sup>41</sup> tries to tackle CQs such as the following (the rest of the CQs tackled are available in Appendix B):

- CQ36: What are the actuating procedures?
- CQ37: What are the predictive procedures?
- CQ38: What are the imputation procedures?

P4EEPSA represents four different types of procedures: actuating procedures (*p4eepsa:ActuatingProcedure*) 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.

An overview of the main classes defined in P4EEPSA are shown in Figure 4.11.

#### 4.5.4 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<sup>42</sup> tries to tackle CQs such as the following (the rest of the CQs tackled are available in Appendix B):

- CQ40: Which type of sensor is a given sensor?
- CQ46: Is a given executor a window actuator?
- CQ48: Is a given executor a predictive model?

<sup>41</sup><https://w3id.org/eepsa/p4eepsa>

<sup>42</sup><https://w3id.org/eepsa/exr4eepsa>

EXR4EEPSA concepts are categorised in four different classes: sensors, actuators, predictive models and imputation methods. The *exr4eeepsa:Sensor* class 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 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 *exr4eeepsa: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 anemometers (*exr4eeepsa:Anemometer*) for sensing wind speed and humidity sensors (*exr4eeepsa:HumiditySensor*). Furthermore, these environment sensors include the *exr4eeepsa:AirQualitySensor* subclass comprising agents sensing air pollution and gases in the surrounding area (e.g. *exr4eeepsa:CO2Sensor*).

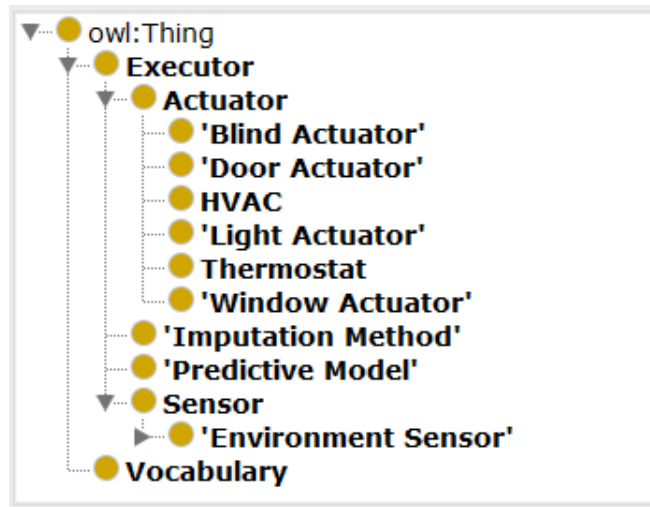


Figure 4.12: Overview of the classes defined in EXR4EEPSA.

The *exr4eeepsa:Actuator* class 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 in tertiary buildings, such as door actuators (*exr4eeepsa:DoorActuator*) and window actuators (*exr4eeepsa: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 tertiary buildings. Furthermore, two additional high-level class of executors are defined in EXR4EEPSA. The first one is the *exr:PredictiveModel* class, 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 re-engineered and reused from the SEAS Smart Meter ontology<sup>43</sup>. Furthermore, the EXR4EEPSA ontology module is aligned with these two related ontologies.

An overview of the main classes defined in EXR4EEPSA is shown in Figure 4.12.

<sup>43</sup><https://w3id.org/seas/SmartMeterOntology>

### 4.5.5 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<sup>44</sup> tries to tackle CQs such as the following (the rest of the CQs tackled are available in Appendix B):

- CQ50: Which executions are actuations?
- CQ51: Which executions are observations?
- CQ52: Which observations are forecasted?

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:Missing Value*), 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. Regarding the outliers, a set of classes represent outliers originated from different causes, such as a poor sensor location (*exn4eepsa:OutlierBySensorLocation*) or an error on a device (*exn4eepsa:OutlierByDeviceError*). Furthermore, EXN4EEPSA 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<sup>45</sup>, SAREF and om-lite ontology<sup>46</sup>. 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.

An overview of the main classes defined in EXR4EEPSA is shown in Figure 4.13.

<sup>44</sup><https://w3id.org/eepsa/exn4eepsa>

<sup>45</sup><https://w3id.org/seas/DeviceOntology>

<sup>46</sup><http://def.seegrid.csiro.au/ontology/om/om-lite>

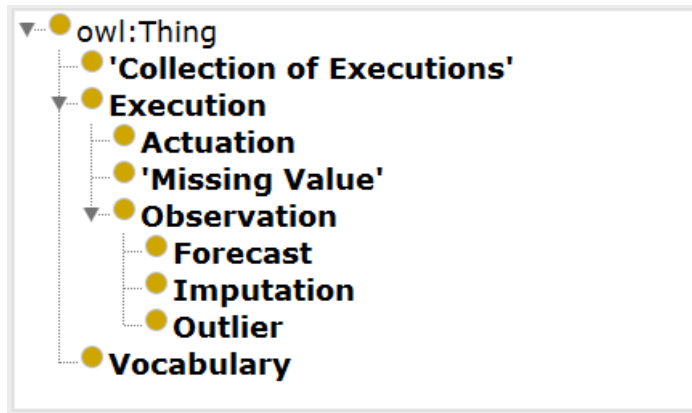


Figure 4.13: Overview of the classes defined in EXN4EEPSA.

#### 4.5.6 EK4EEPSA (Expert Knowledge for EEPsA) ontology module

This ontology module covers the necessary expert knowledge to provide inferring 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, the EK4EEPSA ontology module<sup>47</sup> tries to tackle CQs such as the following (the rest of the CQs tackled are available in Appendix B):

- CQ62: What is a naturally enlightened space?
- CQ63: Which types of spaces are in a building?
- CQ65: 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 the outdoor (*ek4eeppsa:AdjacentToOutdoorSpace*) or spaces located below the ground floor (*ek4eeppsa:BelowGroundLevelSpace*). However, other space definitions such as the proposed by the HBC ontology<sup>48</sup> may also be incorporated, where spaces are mainly characterized by the equipment contained they contain (or not) (e.g. *hbc:SpaceWithHeater* or *hbc:SpaceWithoutHeater*). Note that in the scenario tackled in this thesis, 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

<sup>47</sup><https://w3id.org/eeppsa/ek4eeppsa>

<sup>48</sup><https://w3id.org/ibp/hbc>

dependent on explicit assertions only. For example, the *ek4eepsa:AdjacentToOutdoorSpace* is defined as follows:

$$\begin{aligned} &ek4eepsa:AdjacentToOutdoorSpace \equiv \\ &bot:Space \sqcap \exists bot:hasElement.foi4eepsa:ExternalBuildingElement \end{aligned}$$

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 requires from the ability to ask for interrelationships of entities coming from any other modules. For example, 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:

$$\begin{aligned} &ek4eepsa:AdjacentToOutdoorSpaceIndoorTemperature \sqsubseteq \\ &\exists aff:affectedBy.q4eepsa:IndoorHumidity \\ &\sqcap \exists aff:affectedBy.q4eepsa:Occupancy \\ &\sqcap \exists aff:affectedBy.q4eepsa:SolarRadiation \\ &\sqcap \exists aff:affectedBy.q4eepsa:WindSpeed . \end{aligned}$$

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 dissertation, the EK4EEPSA ontology module solves the presented CQs. However, being an ontology module containing expert knowledge, it is extendible as more requisites are demanded.

## 4.6 The EEPSA Ontology Customization

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 problems in different types of buildings. Being modularized by design, the EEPSA ontology is expected to be easily customized. Furthermore, as it is demonstrated (in the evaluation section) that the EEPSA ontology modules are loosely coupled and have 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. Next, this ontology customization process



is illustrated with the H2020 RESPOND project<sup>49</sup> use case. Furthermore, a more exhaustive EEPISA ontology customization is performed in section 7.2.1.

#### 4.6.1 EEPISA ontology customization illustrative example: Residential Buildings

Peak energy demand has a negative impact on energy grid capital, operational cost and environmental aspects. This is mainly caused by the carbon-intense generation plants that are deployed to satisfy these energy peak demands [155]. Demand side management activities such as load curtailment or load reallocation, have a huge potential to match energy demand with energy supply side. This is particularly true for the residential sector, which is still a largely untapped sector. Since renewable energy sources are increasingly penetrating the energy production side, their dependence on the weather and climatic conditions largely influences their management and exploitation. In order to tackle these issues, Demand Response (DR) programs are introduced into the smart grids so that reliable and economical operation of power systems are ensured. DR can be understood as technologies or programs that concentrate on shifting energy use to help balancing energy supply and demand [156]. The RESPOND project aims to deploy an interoperable energy automation, monitoring and control solution to deliver DR programs at a dwelling, building and district level.

One of the DR actions considered in the RESPOND project consists in leveraging the thermal inertia of a room to minimize the use of heating systems. In this case, a model predicting the temperature of the room is needed, in order to decide when and how to activate or deactivate the corresponding heating system. Let us consider that the data analyst facing the development of such a model is aware of the EEPISA and wants to leverage it to receive support throughout the different KDD phases. However, the use case building is not tertiary, but residential. Therefore, the current EEPISA ontology, which is focused on tertiary buildings, may not be suitable to meet the problem requirements, and its customization will be required.

The FoI4EEPISA ontology module does not describe home appliances, which are relevant for the problem at hand. Therefore, the FoI4EEPISA needs to be replaced by an ontology module that adequately covers these appliances. Towards this goal the Ontological Resource Reuse Process proposed by the NeOn Methodology is applied, looking for existing ontologies that already describe these concepts. The DogOnt ontology describes appliances and separates them into “white” and “brown” goods depending on their energy consumption profile. Although the *dogOnt:WhiteGoods* class (representing household appliances) and its subclasses are relevant, they contain unnecessary axioms for the matter at hand (e.g. the states or functionalities of each appliance) so that they are not reused as they are. Furthermore there are some other relevant appliances such as tumble-dryers that are not defined. Therefore, the white goods hierarchy defined by DogOnt is re-engineered and extended with a set of classes (e.g. *foi4rbeepsa:Freezer* and *foi4rbeepsa:TumbleDryer*) in order to satisfy the use case requirements. Such

---

<sup>49</sup><http://project-respond.eu/>

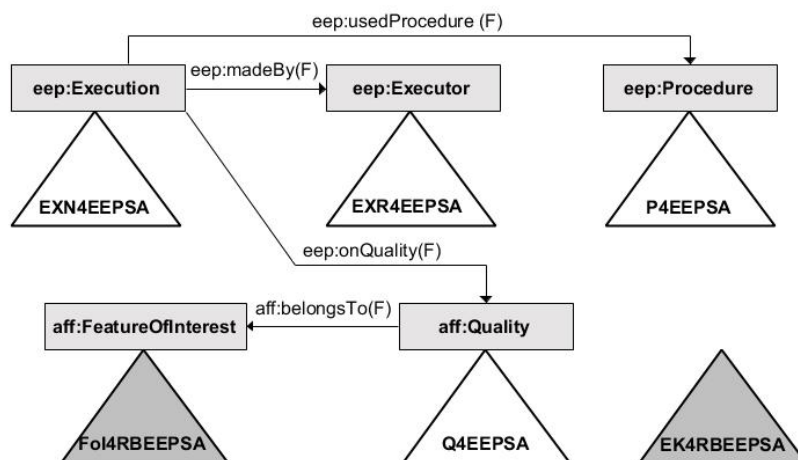


Figure 4.14: Overview of the ontology modules replaced by the EEPSA ontology's customization for residential buildings domain.

an extension leads to the creation of the new FoI4RBEEPSA (FoI for Residential Building EEPSA) ontology module<sup>50</sup>, which will replace the FoI4EEPSA ontology module in the new customized EEPSA ontology for residential buildings.

This ontology module customization task is performed in another EEPSA ontology module that in its current state is not suitable for addressing the use case requirements: EK4EEPSA. As a consequence, a new ontology modules is generated: EK4RBEEPSA<sup>51</sup>. Finally, the new RBEEPSA ontology, which is the EEPSA ontology's customization for residential buildings, imports all the original EEPSA ontology modules, except for the two ontology modules that could not satisfy the use case requirements (i.e. FoI4EEPSA and EK4EEPSA). Instead, the new ontology will import the new FoI4RBEEPSA and EK4RBEEPSA ontology modules. The RBEEPSA ontology<sup>52</sup> is depicted in Figure 4.14.

## 4.7 Documentation

When discovering an ontology, one of the first activities consists in reading its documentation to understand the ontology domain and determine whether it describes this domain appropriately or not. This is why nowadays, most ontologies have comprehensive web pages describing their theoretical backgrounds and features. This is, to a great extent, due to the proliferation of tools for the automatic generation of HTML documentation from ontologies. These tools minimize the efforts of writing proper documentation, and enable the interactive exploration of the ontology with the use of hyperlinks and/or Javascript mecha-

<sup>50</sup><https://w3id.org/rbeepsa/foi4rbeepsa>

<sup>51</sup><https://w3id.org/rbeepsa/ek4rbeepsa>

<sup>52</sup><https://w3id.org/rbeepsa>

nisms. Furthermore, a good documentation increases the understandability and potential usability of ontologies, both by experts in semantics and by people who are not necessarily experts in semantics and languages like OWL or RDF [157].

One of the first tools generating documentation for ontologies, RIF (Rule Interchange Format) rules and combinations of both of them was Parrot [158]. Parrot could retrieve ontology and RIF rule files from the Web, although it also supported the direct file upload for generating the documentation. LODE (Live OWL Documentation Environment) [159] is an online service that automatically generates a human-readable description of any OWL ontology (or, more generally, an RDF vocabulary), taking into account both ontological axioms and annotations. This documentation is presented to the user as an HTML page with embedded links to ease the browsing and navigation. WIDOCO (a Wizard for DOCUMENTing Ontologies) [160] creates a documentation with diagrams, human readable descriptions of the ontology terms and a summary of changes with respect to previous versions of the ontology. The documentation consists of a set of linked enriched HTML pages that can be further extended by users. WIDOCO builds on top of LODE and extends it with properties to qualify terms in the ontology.

The documentation of the EEPSA ontology and its components is generated with WIDOCO. Furthermore, the documentation has been extended with hand-made sections such as the alignments with other ontologies or ontology usage examples.

### 4.7.1 Ontology metadata

W3C's Data on the Web Best Practices [141] 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. There are different guidelines available for describing ontology metadata [161, 162]. The ODPs and ontology modules presented in this thesis were annotated following guidelines described by Garijo and Poveda-Villalón [162] as it was considered the most complete guideline among the ones reviewed. Let be the following prefixes and namespaces for the associated metadata:

- @prefix bibo: <http://purl.org/ontology/bibo/>
- @prefix cc: <http://creativecommons.org/ns#>
- @prefix dc: <http://purl.org/dc/terms/>
- @prefix owl: <http://www.w3.org/2002/07/owl#>
- @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
- @prefix vann: <http://purl.org/vocab/vann/>
- @prefix vs: <http://www.w3.org/2003/06/sw-vocab-status/ns#>

Regarding the description of the EEPSA ontology modules and ODPs, the following metadata terms are used:

- *vann:preferredNamespaceUri* for the ontology's main URI.
- *vann:preferredNamespacePrefix* for the preferred prefix used to refer to the ontology.
- *dc:title* for the ontology's title.
- *dc:description* for describing the ontology.
- *owl:versionInfo* for the version number of the ontology.
- *dc:created* for the date when the ontology was created.
- *dc:modified* for the date when the ontology was modified last.
- *dc:issued* for the date when the ontology was published.
- *dc:creator* for the people who created the ontology.
- *dc:contributor* for the people who contributed to the development of the ontology.
- *cc:license* for specifying the property right associated with the ontology.
- *bibo:status* for indicating the status of the ontology.

As for describing the classes, properties and individuals of the ontology, the following four metadata terms were used:

- *rdfs:label* for a readable label of the term.
- *rdfs:comment* for the textual definition of the term.
- *rdfs:isDefinedBy* for specifying the source used to define the term.
- *vs:term\_status* for indicating the usage status of the term. This metadata's range consists in a set of fixed values (i.e. archaic, testing, stable and unstable).

The aforementioned metadata comprise all the terms recommended by the selected metadata guidelines [162]. Furthermore, there are also some terms that, even though labelled as optional, were considered relevant and helpful for understanding the ontology and foster its reuse (e.g. *dcterms:issued* or *sw:term\_status*). Lastly, some concepts were also described with *rdfs:seeAlso* to refer to sources where a further explanation of the concept can be found.

### 4.7.2 EEP SA ontology documentation

For URI stability and manageability purposes, the W3C Permanent Identifier Community Group’s<sup>53</sup> [w3id.org](https://w3id.org)<sup>54</sup> redirection service is used. The purpose of this initiative is to provide a secure, permanent URL re-direction service for Web applications. The EEP SA ontology owns a [w3id](https://w3id.org) PROJECT-ID called “[eep sa](https://w3id.org)” and redirections to actual EEP SA GitHub pages are defined using Apache `htaccess` documents.

Next, the canonical URIs for the different EEP SA ontology components are shown

- EEP SA ontology: <https://w3id.org/eep sa>
- AffectedBy ODP: <https://w3id.org/affectedBy>
- EEP ODP: <https://w3id.org/eep>
- RC ODP: <https://w3id.org/rc>
- FOI4EEP SA ontology module: <https://w3id.org/eep sa/foi4eep sa>
- Q4EEP SA ontology module: <https://w3id.org/eep sa/q4eep sa>
- P4EEP SA ontology module: <https://w3id.org/eep sa/p4eep sa>
- EXR4EEP SA ontology module: <https://w3id.org/eep sa/exr4eep sa>
- EXN4EEP SA ontology module: <https://w3id.org/eep sa/exn4eep sa>
- EK4EEP SA ontology module: <https://w3id.org/eep sa/ek4eep sa>

Each component of the EEP SA ontology is available in TTL, RDF/XML, JSON-LD and HTML formats. A server content negotiation mechanism is used to serve the adequate version.

With regards to the ODPs, they are also available in the ODP repository<sup>55</sup>, which collects and makes ODPs available on the web, allowing users to download, propose, and discuss them.

Furthermore, the EEP SA ontology is registered on the Linked Open Vocabularies<sup>56</sup> repository.

## 4.8 Ontology Evaluation

There are many evaluation metrics for assessing ontologies in existing literature [163, 164]. Most of them, focus on structural notions without taking into

<sup>53</sup><https://www.w3.org/community/perma-id/>

<sup>54</sup><https://w3id.org/>

<sup>55</sup><http://ontologydesignpatterns.org>

<sup>56</sup><https://lov.linkeddata.es/dataset/lov/vocabs/eep sa>

Table 4.1: Summary of ontology design correctness evaluation by OOPS!

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

account the semantics, leading to incomparable measurement results [165]. And even though these are valid metrics, they may not be enough to determine the quality of an ontology. Likewise, it is also difficult to determine whether an ontology module is actually good or not. This is not only caused because metrics are not comprehensive enough to a variety of ontology modules but also because it is unclear which metrics fare well with particular types of ontology modules.

In order to avoid biased evaluations, next, the EEP SA ontology modules and the proposed ODPs are assessed from three perspectives: design correctness, structural metrics, and modularity quality.

#### 4.8.1 Design correctness metrics

The design correctness is evaluated using OOPS! (Ontology Pitfall Scanner) [166], which detects some of the most common pitfalls appearing within ontology developments. OOPS! is available online<sup>57</sup> 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 4.1 summarizes the number of pitfalls detected in each EEP SA ontology module and ODP. Detailed results are shown in Appendix C.

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 EEP SA ontology. Therefore, these pitfalls are ignored.

<sup>57</sup><http://oops.linkeddata.es/>

Table 4.2: Summary of ontology structural metrics by Protégé’s Ontology Metrics tab.

Ontology	Axioms	Class	OP	DP	Annotation	DL Expr
AffectedBy	62	3	3	0	31	ALERIF+
EEP(*)	80	6	8	0	40	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	114	16	2	0	36	ALI
EK4EEPSA	81	25	4	0	32	ALC

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 them.

#### 4.8.2 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<sup>58</sup> that displays entity and axiom counts for the active ontology. Table 4.2 summarizes the structural metrics for the different EEPSA ontology modules and ODPs. In this table, column represents the number of object properties, DP the number of datatype properties, LD Expr the DL Expressivity, and in ontologies marked with an asterisk (\*), imported axioms are not considered. These metrics are further detailed in Appendix C.

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 thesis, 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.

<sup>58</sup><http://protegeproject.github.io/protege/views/ontology-metrics>

### 4.8.3 Modularity quality metrics

The quality of the EEPSA ontology modules and the developed three ODPs is assessed based on the guidelines proposed by Khan and Keet [167]. This work creates a comprehensive list of module evaluation metrics as well as a definition of 14 types of ontology modules:

- T1: Ontology design patterns modules, when an ontology is modularised by identifying a part of the ontology for general reuse.
- T2: Subject domain modules, when a large domain is divided by subdomains present in the ontology.
- T3: Isolation branch modules, when a subset of entities from an ontology is extracted but entities with weak dependencies to the signature are not to be included in the module.
- T4: Locality modules, when a subset of entities from an ontology is extracted, including all entities that are dependent on the subset.
- T5: Privacy modules, when some information is hidden from an ontology.
- T6: Domain coverage modules, when a large ontology is partitioned by its graphical structure and placement of entities in the taxonomy.
- T7: Ontology matching modules, when an ontology is modularised for ontology matching into disjoint modules so that there is no repetition of entities.
- T8: Optimal reasoning modules, when an ontology is split into smaller modules to aid in overall reasoning over the ontology.
- T9: Axiom abstraction modules, when an ontology is modularised to have fewer axioms, to decrease the horizontal structure of the ontology.
- T10: Entity type abstraction modules, when an ontology is modularised by removing a certain type of entity e.g. data properties or object properties.
- T11: High-level abstraction modules, when an ontology is modularised by removing lower-level classes and only keeping higher-level classes.
- T12: Weighted abstraction modules, when an ontology is modularised by a weighting decided by the developer
- T13: Expressiveness sub-language modules, when an ontology is modularised by using a sub-language of a core ontology language.
- T14: Expressiveness feature modules, when an ontology is modularised by using limited language features

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<sup>59</sup> (Tool for Ontology Module Metrics) and results are available both online<sup>60</sup> and in Appendix C.

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 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.

## 4.9 Ontology Versioning

The EEPSA ontology presented in this chapter is the latest version available at the moment of writing this dissertation (version 2.0). The EEPSA ontology was developed with an iterative life cycle, which means that the ontology was progressively extended to cover new requirements. As a matter of fact, different EEPSA ontology versions exist, used for different EEPSA components because version 2.0 was not available at the moment of developing such EEPSA components.

The EEPSA ontology version 2.0 is a restructuring of the previous versions, based on ODPs and ontology modules in order to ease its maintainability, extensibility and reuse among others. This ontology version is used in the EEPSA ETL process (explained in section 5.4) and as a base to be customized in the poultry farm use case (explained in Chapter 7).

Next, the main features of the previous EEPSA ontology versions used in the different EEPSA components are briefly introduced.

---

<sup>59</sup><http://www.thezfiles.co.za/Modularity/TOMM.zip>

<sup>60</sup><https://github.com/iesnaola/ee psa/tree/master/Evaluation/TOMM>

### 4.9.1 EEPsa ontology version 1.2

This ontology version reuses and re-engineers some of the ontologies reviewed in section 3.1. The suite of imported modules by this EEPsa ontology’s version includes the tailor made bim4EEPsa module’s version 0.1<sup>61</sup>, which was devised to describe buildings and their spaces; the SSN ontology to cover observations and actuations; the measurements4EEPsa module version 0.1<sup>62</sup>, which was composed in order to cover measurement related concepts; the OWL-Time ontology to describe time-related concepts; and Basic Geo Vocabulary to represent spatially located things.

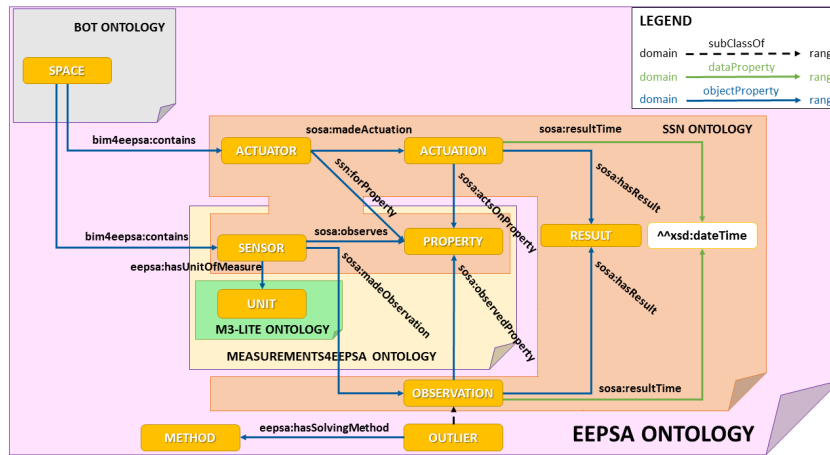


Figure 4.15: An overview of relevant classes and properties in the EEPsa ontology version 1.2.

The bim4EEPsa module imports BOT and extends it with some other generic classes such as *bim4eepsa:Door* and *bim4eepsa:Window*. With regards to the measurements4EEPsa module, it is composed of a set of subclasses of *sosa:Sensor*, *ssn:Property* and *qudt:Unit* extracted from the M3-lite ontology. Furthermore, the EEPsa ontology version 1.2 defines a class *eepsa:Outlier* for outliers, and methods to avoid the generation of future outliers such as *eepsa:DeviceRelocation* and *eepsa:DeviceShelter*.

This ontology version is available online at: [https://raw.githubusercontent.com/iesnaola/eepsa/master/EEPsa\\_previousVersions/eepsa-1.2.owl](https://raw.githubusercontent.com/iesnaola/eepsa/master/EEPsa_previousVersions/eepsa-1.2.owl). It was used for the SemOD framework (explained in section 5.3.1). Figure 4.15 shows an excerpt of the ontology’s relevant classes and properties.

<sup>61</sup>[https://raw.githubusercontent.com/iesnaola/eepsa/master/EEPsa\\_previousVersions/bim4eepsa-0.1.owl](https://raw.githubusercontent.com/iesnaola/eepsa/master/EEPsa_previousVersions/bim4eepsa-0.1.owl)

<sup>62</sup>[https://raw.githubusercontent.com/iesnaola/eepsa/master/EEPsa\\_previousVersions/measurements4eepsa-0.1.owl](https://raw.githubusercontent.com/iesnaola/eepsa/master/EEPsa_previousVersions/measurements4eepsa-0.1.owl)

### 4.9.2 EEPSA ontology version 1.3

This ontology version is the extension of the EEPSA ontology version 1.2 in order to address the requirements of the EEPSA’s support in the KDD Interpretation phase.

It leverages the forecasting4eepsa (Forecasting for EEPSA) ontology module<sup>63</sup>, which comprises the necessary terms to represent the predictive models, the procedures they implement, and the results obtained, by reusing and extending the SEAS Forecasting ontology<sup>64</sup> (which is a module of the SEAS Ontology that extends PEP). The SEAS Forecasting ontology defines the class *seas:Forecaster* whose individuals implement *seas:Forecasting* processes and make individuals of class *seas:Forecast*. This ontology is extended with class *f4eepsa:ForecastResult* to represent forecast results, as well as with class *f4eepsa:ForecastingInput* to represent inputs of forecasting processes. Furthermore, a prediction may contain many prediction results (one for each predicted instant); therefore, object property *f4eepsa:hasForecastResult* is defined. Figure 4.16 shows an overview of the forecasting4eepsa ontology module which is imported by the EEPSA ontology version 1.3.

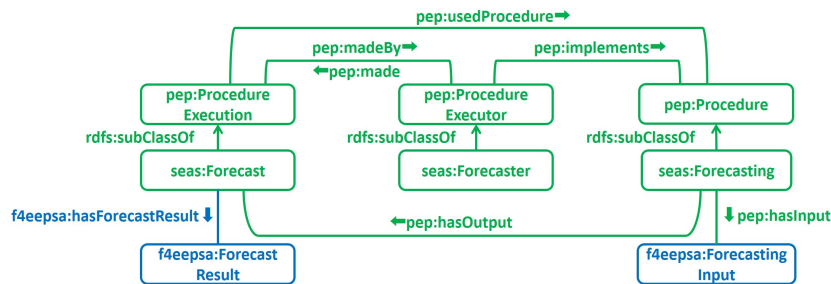


Figure 4.16: Overview of the Forecasting4eepsa ontology module.

Furthermore, the measurements4eepsa module is extended with knowledge representing HVAC systems and HVAC control strategies among others, leading to the version 0.3<sup>65</sup>. Class *m4eepsa:HVAC*, which is a subclass of *seas:Actuator*, is a simplified representation of a real-world HVAC system, and *m4eepsa:HVAC-ControlStrategy* is defined as subclass of *seas:Actuation* to represent HVAC control strategies made by HVAC systems. Furthermore, an HVAC control strategy makes different actuations (with object property *m4eepsa:hasActuation*) over time. Each actuation’s result (represented with an individual of class *sosa:Result*) is characterized mainly by a date time when the actuation takes place (with datatype property *sosa:resultTime*), a temperature that the space is aimed to have (with object property *m4eepsa:temperatureSetpoint*), and the number of AHUs<sup>66</sup> activated (with datatype property *m4eepsa:numberOfActiveAHUs*). Fig-

<sup>63</sup><https://w3id.org/forecasting4eepsa>

<sup>64</sup><https://w3id.org/seas/ForecastingOntology>

<sup>65</sup>[https://raw.githubusercontent.com/iesnaola/eepsa/master/EEPSA\\_previousVersions/measurements4eepsa-0.3.owl](https://raw.githubusercontent.com/iesnaola/eepsa/master/EEPSA_previousVersions/measurements4eepsa-0.3.owl)

<sup>66</sup>AHU (Air Handling Unit) is an HVAC system component used to regulate and circulate air. There may be more than one AHUs associated to a single HVAC system, usually in charge

Figure 4.17 shows an overview of measurements4eepsa module's extension.

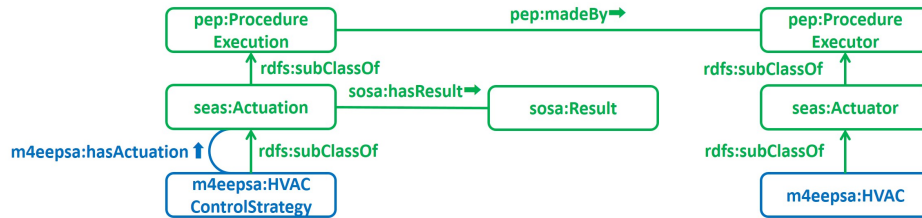


Figure 4.17: Overview of Measurements4eepsa ontology module's extension.

Last but not least, terms describing thermal comfort regulations are also added. A class *eepsa:ThermalComfortRegulation* is defined representing HVAC control strategies that fulfill a set of regulations and guidelines that ensure occupants' comfort with the thermal environment. This class has subclasses such as *eepsa:INSHTForSedentarySituation* and *eepsa:HSESituation*, which describe HVAC control strategies fulfilling thermal comfort regulations for workplaces defined by different entities.

This ontology version is available online at: [https://raw.githubusercontent.com/iesnaola/eepsa/master/EEPSA\\_previousVersions/eepsa-1.3.owl](https://raw.githubusercontent.com/iesnaola/eepsa/master/EEPSA_previousVersions/eepsa-1.3.owl). It was used for the EROSO framework (explained in section 5.6).

## Chapter 5

# The EEPSA

EEPSA aims at supporting data analysts throughout the KDD process towards the creation of predictive models to solve energy efficiency and thermal comfort problems in tertiary buildings, by leveraging Semantic Technologies. In the previous chapter, the development of the EEPSA ontology<sup>1</sup>, which is the cornerstone of the EEPSA data analyst was described. Next, the description of the EEPSA's use of Semantic Technologies in the different KDD phases is shown.

### 5.1 Semantic Annotation

In the context of the EEPSA, the semantic annotation process is understood as the alignment of resources (or parts of them) with a description of some of their properties and features represented with ontology terms. This way, resources are provided with explicit and unambiguous semantics. Without this explicit semantic assignment, different users could refer to the same resource with different meanings, or they could even refer to different resources with the same meaning. Furthermore, a semantically annotated dataset improves semantic interoperability<sup>2</sup>, providing both humans and machines with a shared meaning of terms [169].

This semantic annotation phase can be accomplished by manually editing an RDF model with the help of an adapted GUI (Graphical User Interface) or a data wrangling tool, or else with a properly programmed automatic middleware.

In this phase, all the relevant concepts related to the problem at hand are semantically annotated with the corresponding terms of the EEPSA ontology. These concepts span the building and building spaces (features of interest from FoI4EEPSA<sup>3</sup> and space types from EK4EEPSA<sup>4</sup>) and their qualities (from Q4-

---

<sup>1</sup><https://w3id.org/eepsa>

<sup>2</sup>Semantic interoperability is concerned with ensuring that the exchanged information has the same meaning for both message sender and receiver [168].

<sup>3</sup><https://w3id.org/eepsa/foi4eepsa>

<sup>4</sup><https://w3id.org/eepsa/ek4eepsa>

EEPSA<sup>5</sup>), as well as the agents (executors from EXR4EEPSA<sup>6</sup>) that apply specific plans or methods (procedures from P4EEPSA<sup>7</sup>) to produce events (executions from EXN4EEPSA<sup>8</sup>) related to those qualities. Whether the annotated data is stored natively or viewed as an RDF model by means of a middleware, this data will be accessible via SPARQL queries.

## 5.2 Data Selection

In this thesis it is conjectured that Semantic Technologies could be further exploited to assist data analysts in the Data Selection phase, rather than just using tools for data visualization purposes as it is reviewed in the related work section. Furthermore, relevance analysis may have performance issues in large and heterogeneous datasets [170], and they are not capable of suggesting new relevant attributes that are not present at the current dataset. EEPSA proposes the use of ontologies and assists the data analyst by suggesting the sets of data and variables that will potentially contribute in the development of an accurate predictive model.

In this regard, the EEPSA ontology captures knowledge specific to space types in buildings within the EK4EEPSA ontology module. This knowledge is represented in the form of OWL axioms so that it allows a reasoner inferring relevant information of the problem at hand.

First of all, the axioms captured in EK4EEPSA allow the classification of the building space at hand into one or more space types. This classification is inferred according to the building elements defined in the building space at hand, and the description of the building spaces within the EK4EEPSA ontology module. For example, a space with windows towards the outside is inferred to be a naturally enlightened space (*ek4eepsa:NaturallyEnlightenedSpace*), due to the axioms:

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix aff: <https://w3id.org/affectedBy#> .
@prefix bot: <https://w3id.org/bot#> .
@prefix foi4eepsa: <https://w3id.org/eepsa/foi4eepsa#> .
@prefix ek4eepsa: <https://w3id.org/eepsa/ek4eepsa#> .
@prefix vs: <http://www.w3.org/2003/06/sw-vocab-status/ns#> .

ek4eepsa:NaturallyEnlightenedSpace rdf:type owl:Class ;
  owl:equivalentClass [ owl:intersectionOf ( bot:Space
    [ rdf:type owl:Class ;
      owl:unionOf ( [ rdf:type owl:Restriction ;
        owl:onProperty bot:hasElement ;
```

<sup>5</sup><https://w3id.org/eepsa/q4eepsa>

<sup>6</sup><https://w3id.org/eepsa/exr4eepsa>

<sup>7</sup><https://w3id.org/eepsa/p4eepsa>

<sup>8</sup><https://w3id.org/eepsa/exn4eepsa>

```

    owl:someValuesFrom foi4eepsa:ExternalWindow ]
      [ rdf:type owl:Restriction ;
        owl:onProperty bot:hasElement ;
        owl:someValuesFrom foi4eepsa:Skylight ] ) ] ) ;
  rdf:type owl:Class ] ;
  rdfs:subClassOf bot:Space ,
[ rdf:type owl:Restriction ;
  owl:onProperty aff:influencedBy ;
  owl:someValuesFrom
    ek4eepsa:NaturallyEnlightenedSpaceIndoorTemperature
] ;
  rdfs:comment "A space enlightened with a source of light
from the exterior."@en ;
  rdfs:isDefinedBy <https://w3id.org/eepsa/ek4eepsa> ;
  rdfs:label "Naturally Enlightened Space"@en ;
  vs:term_status "stable"^^xsd:string .

```

Listing 5.1: Definition of a Naturally Enlightened Space in EK4EEPSA.

Furthermore, the EK4EEPSA ontology module defines a set of qualities that affect each space type's indoor temperature. These definitions are founded on the AffectedBy ODP, and they enable the inference of variables that data analysts need to take into account for an accurate temperature prediction of the space at hand. For example, an individual of class *ek4eepsa:NaturallyEnlightenedSpace* has its indoor temperature affected by the space humidity, occupancy and cloud coverage among others. These qualities are represented in the following axioms:

```

@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix aff: <https://w3id.org/affectedBy#> .
@prefix q4eepsa: <https://w3id.org/eepsa/q4eepsa#> .
@prefix ek4eepsa: <https://w3id.org/eepsa/ek4eepsa#> .
@prefix vs: <http://www.w3.org/2003/06/sw-vocab-status/ns#> .

ek4eepsa:NaturallyEnlightenedSpaceIndoorTemperature rdf:type
owl:Class ;
  rdfs:subClassOf q4eepsa:IndoorTemperature ,
  [ owl:intersectionOf ( [ rdf:type owl:Restriction ;
    owl:onProperty aff:affectedBy ;
    owl:someValuesFrom q4eepsa:CloudCover
  ]
  [ rdf:type owl:Restriction ;
    owl:onProperty aff:affectedBy ;
    owl:someValuesFrom q4eepsa:IndoorHumidity
  ]
  [ rdf:type owl:Restriction ;
    owl:onProperty aff:affectedBy ;
    owl:someValuesFrom q4eepsa:Occupancy
  ]
  [ rdf:type owl:Restriction ;
    owl:onProperty aff:affectedBy ;

```

```

        owl:someValuesFrom q4eepsa:SunPositionDirection
    ]
    [ rdf:type owl:Restriction ;
      owl:onProperty aff:affectedBy ;
      owl:someValuesFrom q4eepsa:SunPositionElevation
    ]
  ) ;
  rdf:type owl:Class
] ;
rdfs:comment "Temperature within a naturally enlightened
space."@en ;
rdfs:isDefinedBy <https://w3id.org/eepsa/ek4eepsa> ;
rdfs:label "Naturally Enlightened Space Indoor
Temperature"@en ;
vs:term_status "stable"^^xsd:string .

```

Listing 5.2: Definition of qualities affecting the indoor temperature of a Naturally Enlightened Space in EK4EEPSA.

Thanks to the aforementioned axioms, a reasoner can infer the qualities affecting the indoor temperature of a given space semantically annotated with EEPSA ontology terms. After that, data analysts need to know which of those qualities are available to use and which not. This can be discovered by instantiating and running the parameterizable SPARQL query shown in Listing 5.3. Wild card *\$SPACE\_TEMP* needs to be replaced with the corresponding space's indoor temperature URI, and wild card *\$SPACE* with the space's URI.

```

PREFIX aff: <http://w3id.org/affectedBy#>

SELECT DISTINCT ?affectingQuality
WHERE {
  { $SPACE_TEMP aff:affectedBy ?affectingQuality. }
MINUS
  { ?affectingQuality aff:belongsTo $SPACE. }
}

```

Listing 5.3: SPARQL query for retrieving qualities that affect but are not observed within the a given space.

Consequently, in the Data Selection phase, once the target building space is semantically annotated, the data analysts gets to know which type of space it is and the variables that may be relevant for predicting its indoor temperature accurately. Moreover, thanks to the parameterizable SPARQL query execution, the analyst knows which of those relevant qualities are available for the space at hand. It is worth emphasizing that this knowledge is captured in the EEPSA ontology and obtained via inferences, so that data analysts are not required to have a deep domain knowledge.



## 5.3 Preprocessing

In this KDD phase, different techniques are considered to deal with different types of data quality issues. This thesis focuses on the role of Semantic Technologies to, on the one hand, detect and classify outliers, and on the other, to deal with missing values.

### 5.3.1 Outlier detection and classification

The SemOD (Semantic Outlier Detection) framework [171] guides data analysts through the detection of outliers in WSNs and their classification according to their potential causes. This framework is also aimed at raising awareness of the potential of Semantic Technologies in both outlier detection and classification tasks.

The framework is intended for novice data analysts as well as data analysts with a lack of knowledge in the domain at hand, therefore outliers are detected and classified in a (semi-) automatic way and with no previous knowledge required. Additionally, the SemOD framework can be valuable for expert data analysts who many times overlook potential causes of outliers when trying to detect them in WSNs.

The SemOD framework is composed of three modules: the EEP SA ontology, the SemOD Method and the SemOD Query.

#### 5.3.1.1 The EEP SA ontology

The necessary semantic annotation task in the SemOD framework makes use of the EEP SA ontology version 1.2<sup>9</sup>, and the required information about sensors, measurements, and the context in which they have been measured is annotated with proper terms contained in this ontology version. Afterwards, due to the OWL axioms within the ontology, a reasoner can infer circumstances that make sensors susceptible to suffer from outliers. That is, when sensors are under those circumstances, their observations are likely to be outliers. For example, a temperature sensor is defined to be susceptible to suffer from outliers caused by illuminance, rainfall and solar radiation as shown in Listing 5.4:

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix m3-lite: <http://purl.org/iot/vocab/m3-lite#> .
@prefix sosa: <http://www.w3.org/ns/sosa/> .
@prefix eepsa: <https://raw.githubusercontent.com/iesnaola/
  eepsa/master/EEPSA_previousVersions/eepsa-1.2.owl#> .
```

<sup>9</sup>[https://raw.githubusercontent.com/iesnaola/eepsa/master/EEPSA\\_previousVersions/eepsa-1.2.owl](https://raw.githubusercontent.com/iesnaola/eepsa/master/EEPSA_previousVersions/eepsa-1.2.owl)

```

eepsa:TemperatureSensor rdf:type owl:Class ;
  rdfs:subClassOf sosa:Sensor ,
    [ owl:intersectionOf ( [ rdf:type owl:Restriction ;
      owl:onProperty eepsa:susceptibleToOutliersCausedBy ;
      owl:hasValue m3-lite:Illuminance]
    [ rdf:type owl:Restriction ;
      owl:onProperty eepsa:susceptibleToOutliersCausedBy ;
      owl:hasValue m3-lite:Rainfall]
    [ rdf:type owl:Restriction ;
      owl:onProperty eepsa:susceptibleToOutliersCausedBy ;
      owl:hasValue m3-lite:SolarRadiation] ) ;
  rdf:type owl:Class ] .

```

Listing 5.4: Definition of a Temperature sensor's susceptibilities.

As mentioned in section 3.3.1, several circumstances make WSNs prone to errors. For example, an indoor temperature sensor located in a poorly isolated external wall can be conditioned by external meteorological conditions such as wind speed or solar radiation. When exposed to rain, a wet outdoor sensor will not measure the same humidity as a dry sensor due to the evaporation of water from its surface. A sensor placed next to a light bulb might have its illuminance observations affected when the bulb is switched on. A sensor might not make accurate measurements if power supply levels are not enough. These are just some of the circumstances affecting sensors. Each of these types of outliers is represented in the EEPsa ontology version 1.2 as a subclass of class *eepsa:Outlier*, and each of them is linked to a proposed method to offset the problem, by means of object property *eepsa:hasSolvingMethod*<sup>10</sup>. Summarizing, this first module of the SemOD framework supports data analysts identifying circumstances that make sensors susceptible to outliers.

### 5.3.1.2 The SemOD Method

In order to detect outliers caused by each of those circumstances, the corresponding SemOD Method must be applied. SemOD methods provide data analysts with purposely defined steps and a set of resources towards the (semi-) automatic generation of a SemOD Query to detect outliers caused by a certain circumstance. These resources and steps have been designed by experts in such a way that no previous knowledge regarding the domain or Semantic Technologies are required to exploit them.

The SemOD Method that detects outliers measured by outdoor temperature sensors has been developed as a starting point of the research. This SemOD Method focuses on detecting temperature outliers caused when sensors receive direct solar radiation. Under this circumstance, sensors tend to get hot and can measure much higher temperatures than real ones, resulting in potential outliers. This SemOD Method is composed of three steps:

<sup>10</sup>Not to be confused with the SemOD Method, which aims at guiding the data analyst towards the detection and classification and outliers.

**1<sup>st</sup> step: Sensor's Sun Exposure Constraint generation.** For a temperature measurement to be affected by solar radiation, this SemOD Method specifies that two conditions must happen at the same time. On the one hand, sensor in charge of measuring temperature has to be placed in a place where, during measurement time, is exposed to receive direct solar radiation. And on the other, there must be no obstacles such as clouds on the sun beam lights' way to the sensor.

According to experts, time periods when an object might be exposed to direct solar radiation depends mainly on the object's location, orientation and the time of the year. When the orientation and location of an object are semantically annotated, thanks to EEPsa ontology version 1.2's OWL axioms, a reasoner can infer the periods in which the object might be exposed to the sun. Each of these periods of time is described by means of *eepsa:startingTime*, *eepsa:endingTime* and *eepsa:hasMonth* datatype properties. For example, a sensor *:sensor01* located in Spain and oriented towards the north-west is inferred to be an individual of class *eepsa:NorthWestOrientedObject*. As a part of its definition, any individual belonging to this class will have an *eepsa:hasSunExposurePeriod* object property with values such as *eepsa:periodFebruaryNW*. Therefore, retrieving attribute values of *eepsa:periodFebruaryNW*, it can be concluded that in February, *:sensor01* is exposed to sun between 18:00 and 19:00. Axioms shown in Listing 5.5 enable these inferences.

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix m3-lite: <http://purl.org/iot/vocab/m3-lite#> .
@prefix s4bldg: <https://w3id.org/def/saref4bldg#> .
@prefix m4eepsa : <http://w3id.org/measurements4eepsa#> .
@prefix eepsa: <https://raw.githubusercontent.com/iesnaola/
  eepsa/master/EEPSA_previousVersions/eepsa-1.2.owl#> .

eepsa:NorthWestOrientedObject rdf:type owl:Class ;
  owl:equivalentClass [ owl:intersectionOf ( [ rdf:type
    owl:Restriction ;
      owl:onProperty m3-lite:hasDirection ;
      owl:hasValue m4eepsa:northWestOrientation ]
    [ rdf:type owl:Restriction ;
      owl:onProperty wgs84_pos:location ;
      owl:hasValue <http://es.dbpedia.org/resource/Spain> ] ) ];
  rdfs:subClassOf s4bldg:PhysicalObject ,
    [ owl:intersectionOf ( [ rdf:type owl:Restriction ;
      owl:onProperty eepsa:hasSunExposurePeriod ;
      owl:hasValue eepsa:periodFebruaryNW ]
    [ rdf:type owl:Restriction ;
      owl:onProperty eepsa:hasSunExposurePeriod ;
      owl:hasValue eepsa:periodAugustNW ]
    (...) ] ;
  rdfs:label "North West Oriented Object"@en .

eepsa:periodFebruaryNW rdf:type owl:NamedIndividual ;
```

```

eepsa:endingTime "19:00:00" ;
eepsa:hasMonth 4 ;
eepsa:startingTime "18:00:00" ;
rdfs:comment "Individual representing sun exposure period
in April for objects oriented towards the North West."@en .

```

Listing 5.5: Axioms enabling the discovery of an object's sun exposure periods.

Keeping sun exposure periods generic for every object is not feasible because these periods vary depending on the location. For example, in December an object located in Tromsø (Norway) may have a smaller period of sun exposure compared with an object located in Santiago (Chile). The EEPsa ontology version 1.2 captures sun exposure periods that may be acceptable for locations in Spain. However, even in Spain there might be places where these values might not be completely accurate. In case the periods' starting and ending times need to be refined, the data analyst is always free to do so during the next stage of the SemOD framework, that is, in the SemOD Query Execution.

The SemOD Method presents a constraint pattern describing an object's sun exposure period as presented in Listing 5.6. This constraint pattern is composed of a month (integer) and two time values (in *hh:mm:ss* format), so that it retrieves resources that happen during the month and between the two times values. In order to instantiate this pattern, wild cards *\$MONTH\_VALUE*, *\$STARTING\_TIME* and *\$ENDING\_TIME* need to be replaced with values contained in the ontology assertions.

```

(month(?date) = $MONTH_VALUE &&
?time >= xsd:time($STARTING_TIME) &&
?time <= xsd:time($ENDING_TIME))

```

Listing 5.6: Constraint pattern describing an object's sun exposure times.

In order to retrieve sun exposure periods of an object, the SemOD Method proposes the parameterizable SPARQL query shown in Listing 5.7. Wild card *\$OBJECT* needs to be replaced with the corresponding object's URI.

```

PREFIX eepsa: <https://raw.githubusercontent.com/iesnaola/
eepsa/master/EEPSA_previousVersions/eepsa-1.2.owl#>

SELECT *
WHERE
{
    $OBJECT eepsa:hasSunExposurePeriod ?period.
    ?period eepsa:startingTime ?startingTime;
           eepsa:endingTime ?endingTime;
           eepsa:hasMonth ?monthValue. }

```

Listing 5.7: Parameterizable SPARQL query retrieving an object's sun exposure times.

Values obtained by executing this SPARQL query are used to instantiate the constraint pattern shown in Listing 5.6. This has to be instantiated as many

times as exposure periods the object has. Each instantiation of the constraint pattern has to be linked with the next one using the OR operator. An excerpt of the instantiation produced for *:sensor01* is shown in Listing 5.8.

```
(month(?date) = 2 &&
 ?time >= xsd:time("18:00:00") &&
 ?time <= xsd:time("19:00:00") ) ||
(month(?date) = 3 &&
 ?time >= xsd:time("17:00:00") &&
 ?time <= xsd:time("20:00:00") ) || ...
```

Listing 5.8: Constraint pattern describing *:sensor01*'s sun exposure times.

These constraints address a sensor's sun exposure times. As previously stated, to determine if a sensor measures an outlier due to direct solar radiation's effect, it is also necessary that during these periods of time solar radiation hits the sensor.

**2<sup>nd</sup> step: Sunshine Constraint generation.** In order to resolve whether the sensor receives direct solar radiation or not, the SemOD Method requires information coming from one of these two qualities: illuminance or solar irradiance. That is, it is possible to determine whether there is sunny weather or not using measurements of any of these two qualities. Experts have established that if threshold values of  $70\text{W}/\text{m}^2$  for solar irradiance and  $15,000\text{lx}$  for illuminance are exceeded, it can be considered that it is sunshine.

SemOD Method defines two sources of information to retrieve values for these qualities. Firstly, the sensor that is measuring temperature and secondly, Open Data. Sensor information is considered to be more adequate to create the constraints because most of times, Open Data will provide information for a nearby location but not for the exact sensor location, which can skew results. Therefore, when information coming from these two sources is available, it is preferable to use the data coming from the sensor itself.

The parameterizable SPARQL query shown in Listing 5.9 can be used to determine whether the temperature sensor at hand observes additional qualities such as solar irradiance or illuminance. Wild card *\$SENSOR* has to be replaced with the target sensor's URI being analysed (in the proposed example, *:sensor01*'s URI). Likewise, this query could be used to retrieve all sensors measuring solar irradiance or illuminance, if wild card *\$SENSOR* is replaced with a query variable such as *?sensor* and ASK is replaced with SELECT \*.

```
PREFIX ssn: <http://www.w3.org/ns/ssn/>
PREFIX sosa: <http://www.w3.org/ns/sosa/>
PREFIX m3-lite: <http://purl.org/iot/vocab/m3-lite#>
PREFIX eepsa: <https://raw.githubusercontent.com/iesnaola/
  eepsa/master/EEPSA_previousVersions/eepsa-1.2.owl#>

ASK
WHERE {
```

```

{ $SENSOR ssn:hasSubSystem ?sensor
  ?sensor sosa:observedProperty
            eePSa:SolarIrradiance }
UNION
{ $SENSOR ssn:hasSubSystem ?sensor
  ?sensor sosa:observedProperty
            m3-lite:illuminance   } }

```

Listing 5.9: Parameterizable SPARQL query to determine whether a sensor measures solar irradiance or illuminance qualities.

If a sensor measures any of these qualities, it can be derived whether sun hits the sensor or not, using the threshold values that experts have previously established. This information is used to complete previous constraints. For example, *:sensor01* measures temperature and solar irradiance, so that the constraints created in Listing 5.8 could be completed adding new information regarding sunshine, resulting in Listing 5.10.

```

(month(?date) = 2 &&
 ?time >= xsd:time("18:00:00") &&
 ?time <= xsd:time("19:00:00") &&
 xsd:integer(?solarIrradianceVal) > xsd:integer(70) )
|| ...

```

Listing 5.10: Constraint pattern describing *:sensor01*'s sun exposure times and sunshine levels of those periods.

In case the sensor at hand neither has illuminance nor solar radiation measuring capabilities, SemOD Method recommends to retrieve this information from Open Data.

**3<sup>rd</sup> step: SemOD Query generation.** The resulting constraints from the previous step (Listing 5.10) have to replace the wild card *\$PREVIOUSLY\_GENERATED\_CONSTRAINTS* in the FILTER clause of the predefined SemOD Query pattern shown in Listing 5.11. These constraints also need to be casted into their corresponding data types. Furthermore, the graph where the query is going to be performed needs to be specified in the FROM clause, replacing the *\$RDF\_GRAPH* wild card. Wild cards *\$QUALITY* and *\$UNIT\_OF\_MEASUREMENT* need also to be specified with the quality and unit URI used to derive sun information (i.e. solar irradiance or illuminance).

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX sosa: <http://www.w3.org/ns/sosa/>
PREFIX m3-lite: <http://purl.org/iot/vocab/m3-lite#>
PREFIX eePSa: <https://raw.githubusercontent.com/iesnaola/
  eePSa/master/EEPSA_previousVersions/eePSa-1.2.owl#>

CONSTRUCT {?obs1 rdf:type
  eePSa:OutlierCausedBySolarRadiation}
FROM <$RDF_GRAPH>

```

```

WHERE {
  ?sensor1 sosa:observedProperty
           m3-lite:Temperature .
  ?sensor2 sosa:observedProperty $QUALITY;
           ee psa:hasUnitOfMeasure $UNIT_OF_MEASUREMENT .
  ?obs1 sosa:isObservedBy ?sensor1;
        ee psa:obsTime ?time;
        ee psa:obsDate ?date;
  ?obs2 sosa:isObservedBy ?sensor2;
        ee psa:obsTime ?time;
        ee psa:obsDate ?date;
        sosa:hasSimpleResult ?illuminanceVal .
FILTER(
  $PREVIOUSLY_GENERATED_CONSTRAINTS ) }

```

Listing 5.11: SemOD Query pattern for detecting temperature outliers caused by solar radiation.

Finally, the SemOD Query is generated and ready to be executed. Listing 5.12 shows a snippet of a SemOD Query that has been generated for the *:sensor01* example. As previously stated, proposed SemOD Query is intended to be generic enough and adequate for every location in Spain. However, values used in constraints might need to be fine-tuned in some cases. Data analysts are free to do so in this step of the SemOD Method.

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX sosa: <http://www.w3.org/ns/sosa/>
PREFIX m3-lite: <http://purl.org/iot/vocab/m3-lite#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ee psa: <https://raw.githubusercontent.com/iesnaola/
  ee psa/master/EEPSA_previousVersions/ee psa-1.2.owl#>

CONSTRUCT {?obs1 rdf:type
  ee psa:OutlierCausedBySolarRadiation}
FROM <myGRAPH>
WHERE {
  :sensor01 sosa:observedProperty
           m3-lite:Temperature .
  :sensor01 sosa:observedProperty
           m3-lite:Illuminance;
           ee psa:hasUnitOfMeasure m3-lite:Lux .
  ?obs1 sosa:isObservedBy ?sensor1;
        ee psa:obsTime ?time;
        ee psa:obsDate ?date .
  ?obs2 sosa:isObservedBy ?sensor2;
        ee psa:obsTime ?time;
        ee psa:obsDate ?date;
        sosa:hasSimpleResult ?illuminanceVal .
FILTER(
  (month(?date) = 2 &&
   ?time > xsd:time("18:00:00") &&
   ?time < xsd:time("19:00:00") &&
   xsd:integer(?illuminanceVal) > xsd:integer(15000) )

```

```

    ||
    (... )
}

```

Listing 5.12: SemOD Query for detecting temperature outliers caused by solar radiation in :sensor01.

When executed in the next module, this query will retrieve temperature measurements likely to be outliers because of the sensor being hit by solar radiation.

### 5.3.1.3 SemOD Query execution

Generated SemOD Query has to be executed over the SPARQL endpoint that hosts sensor measurements to detect measurements suspected to be outliers because of receiving direct solar radiation. These measurements will also be classified as individuals of class *ee psa:OutlierCausedBySolarRadiation*. Depending on the needs of the use case, it is up to the data analyst what to do with these detected outliers (e.g. remove them from the dataset, analyse them, etc.).

This query is generated in a (semi-) automatic manner and with no previous knowledge required for data analysts. Furthermore, not only does detect outliers, but also classifies them according to their potential cause.

The SemOD framework exploits Semantic Technologies to detect outliers. This outlier detection approach is different to the one proposed by Gao et al. [109], which relies on nearby sensors to determine whether an observation is an outlier or not. The SemOD framework avoids the dependence with nearby sensors by annotating the context of sensors and observations to determine the existence of outliers. Furthermore, the SemOD framework guides data analysts to detect the cause of those outliers, which to the extent of our knowledge is a novelty. Knowing the provenance of an outlier may enable further decision-making, for example, decisions related to the management of sensors and actuators. Moreover, the ontology captures potential approaches to avoid future outlier problems. It is worth noting that these guidance is done leveraging a set of resources that abstract the data analyst from the underlying Semantic Technologies, so that neither a deep domain knowledge nor expertise in these technologies is required.

## 5.3.2 Missing values handling

Even though there are some approaches showing how to make use of Semantic Technologies to detect and fill missing values, this thesis conjectures that the potential of Semantic Technologies in this task is yet to be unlocked. EEPSA tries to raise awareness of the prominent impact Semantic Technologies could have in handling and imputing missing values. This is a preliminary work that shows promising results which should ideally be further researched and developed.

Time series-specific imputation methods have special characteristics that enable them to exploit the extra information available in the dataset, such as the



relation between nearby observations [172]. However, this characteristic that produces better informed estimations of missing values does not come without a drawback. In fact, these methods could struggle as the sequence of adjacent missing values widens, since most of them rely on the nearest values in terms of time proximity.

In this thesis a set of experiments are performed to have an initial insight on the potential contribution of the Semantic Technologies for bridging the gap of time series-specific imputation methods in section 6.1.3.

## 5.4 Transformation

Most approaches aiming at generating features from LOD, choose general domain knowledge bases such as DBpedia or YAGO, which are more focused on containing general domain information. Although it has been shown that the addition of features coming from these knowledge bases may enrich the dataset at hand and contribute to a potential improvement of predictions [120, 121, 122, 123], further improvements may be conceivable if domain-specific knowledge bases are exploited instead. This is why EEPSA focuses on the exploitation of knowledge bases containing domain-specific facts. More specifically, weather information is exploited, which is may be relevant for the energy efficiency and thermal comfort problems addressed in this thesis.

Nowadays, the data available in Open Data repositories does not normally reach the 5 stars quality defined by the Linked Data star rating system<sup>11</sup>. This means that repositories host data with different formats and a plethora of data structures. Meteorological repositories are no exception and suffer from the same data heterogeneity. As a matter of fact, each meteorological agency may describe weather station metadata and registered measurements with their own data structures and publish them in different file formats. For example, a meteorological repository may publish information in JSON format, while another uses XML files. Furthermore, these public files may even be badly formed or contain invalid characters. Moreover, a meteorology agency may describe weather station location coordinates in geodetic coordinates (also known as WGS84) while another may use UTM (Universal Transverse Mercator) coordinates. Therefore, developing a single universal tool which retrieves data from any meteorological repository may be infeasible.

The EEPSA includes an ETL (Extract, Transform, Load) process for weather stations regulated by Euskalmet<sup>12</sup> (the Basque Meteorology Agency). This agency controls over 100 weather stations installed all across the Basque Country in Spain. Figure 5.1 shows the EEPSA's ETL process for the KDD Transformation phase.

Firstly, the “E” (Extract) part of the ETL process extracts weather station metadata (e.g. weather station's location or installed sensors) and/or the obser-

<sup>11</sup><https://5stardata.info>

<sup>12</sup><http://www.euskalmet.euskadi.eus>

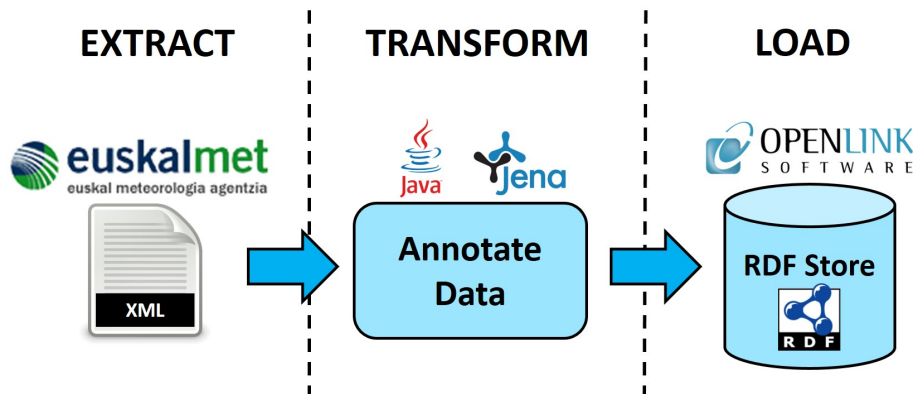


Figure 5.1: EEPSA’s ETL process for the Transformation phase.

vations they register, which are published in Open Data Euskadi<sup>13</sup> (the Basque Open Data portal). In this portal, weather station metadata is publicly available in XML<sup>14</sup>, JSON and XLSX formats, while historical observation data of each weather station is available only in XML files. The former data is updated daily, and the latter, every three months. Furthermore, historical observations are distributed in different XML files, where each of them contains observations registered by a weather station during a month. That is, an XML file contains observations registered by a given weather station during January 2019, while another XML file contains observations registered by the same weather station during February 2019.

Once the aimed information is extracted from the corresponding file, the “T” (Transform) part of the ETL process, which is based on the the Apache Jena framework<sup>15</sup>, semantically annotates the extracted information with appropriate ontological terms. Listing 5.13 shows an excerpt of a semantically annotated Euskalmet weather station.

```
@prefix : <http://www.tekniker.es/euskalmetWeatherStations#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix aemet: <http://aemet.linkeddata.es/ontology/> .
@prefix bot: <https://w3id.org/bot#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#> .
@prefix eep: <https://w3id.org/eep#> .
@prefix exr4eepsa: <https://w3id.org/eepsa/exr4eepsa#> .
@prefix q4eepsa: <https://w3id.org/eepsa/q4eepsa#> .

:weatherStation_Euskalmet_b093 rdf:type aemet:WeatherStation ;
  dbo:owner <http://es.dbpedia.org/page/Euskalmet> ;
  dbo:province <http://dbpedia.org/page/Biscay> ;
```

<sup>13</sup><http://opendata.euskadi.eus>

<sup>14</sup>Some weather station metadata contain invalid XML characters that need to be treated.

<sup>15</sup><https://jena.apache.org/>

```

dc:identifier "B093" ;
foaf:name "Puerto de Ondarroa" ;
geo:lat "43.32579"^^xsd:float ;
geo:long "-2.4157028"^^xsd:float ;
bot:containsElement :weatherStation_Euskalmet_B093_BB ,
                    :weatherStation_Euskalmet_B093_12 ,
                    :weatherStation_Euskalmet_B093_14 ,
                    :weatherStation_Euskalmet_B093_21 ,
                    :weatherStation_Euskalmet_B093_50 ,
(...)

:weatherStation_Euskalmet_B093_21 rdf:type
    exr4eepsa:Sensor ;
eep:forQuality
    :weatherStation_Euskalmet_B093_OutdoorTemperature .

:weatherStation_Euskalmet_B093_OutdoorTemperature rdf:type
    q4eepsa:OutdoorTemperature .

```

Listing 5.13: RDF representation excerpt of an Euskalmet weather station.

Semantically annotated information can then be stored in a virtual RDF model, or materialized in RDF files. For each weather station, an RDF model or RDF file containing its metadata can be generated. Likewise for the observations registered by weather stations during a month. It is worth mentioning that Euskalmet weather stations register observations every 10 minutes (e.g. at 01:00, at 01:10, at 01:20, and so on) and in the context of the EEPSA, having hourly measurements is considered to be enough. Therefore, these RDFs contain measurements registered every hour on the hour (i.e. at 01:00, at 02:00, and so on).

Once the RDF models or files are generated, the “L” (Load) part of the ETL process stores them in a Virtuoso Open Source Server<sup>16</sup>. This RDF Store has a SPARQL endpoint which is available online<sup>17</sup> and enables data analysts retrieving the desired information via SPARQL queries. On the one hand, weather station metadata can be retrieved, and on the other, their observations, which may lead to generate new meteorological variables in the already existing use case’s data pool.

Listing 5.14 shows a parameterizable GeoSPARQL<sup>18</sup> query that retrieves closest weather stations to a pair of given coordinates (replacing the wild cards *\$LAT* and *\$LONG* with sought coordinates in WGS84 format) that measure a certain quality (replacing the wild card *\$QUALITY* with the sought quality’s URI). Furthermore, this GeoSPARQL query’s FILTER clause can be modified to query weather stations measuring multiple qualities, by concatenating the “?quality rdf:type *\$QUALITY*” pattern.

<sup>16</sup><https://virtuoso.openlinksw.com/>

<sup>17</sup><http://193.144.237.227:8890/sparql>

<sup>18</sup>GeoSPARQL defines an extension to the SPARQL query language for processing geospatial data.

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX bot: <https://w3id.org/bot#>
PREFIX aemet: <http://aemet.linkeddata.es/ontology/>
PREFIX eep: <https://w3id.org/eep#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT DISTINCT ?stationID ?stationName
(bif:st_distance((bif:st_point(xsd:float(?lat),
xsd:float(?long))),
(bif:st_point(xsd:float($LAT), xsd:float($LONG))))
AS ?distanceToBuilding
FROM <http://tekniker.es/euskalmetWeatherStations>
WHERE {
?weatherStation rdf:type aemet:WeatherStation;
foaf:name ?stationName;
geo:lat ?lat;
geo:long ?long;
dc:identifier ?stationID.
bot:containsElement ?sensor.
?sensor eep:forQuality ?quality.
?quality rdf:type ?qType.

FILTER (
?qType = $QUALITY )
}
ORDER BY ?distanceToBuilding

```

Listing 5.14: Parameterizable GeoSPARQL query for retrieving nearby weather stations measuring a certain quality.

Listing 5.15 shows a parameterizable SPARQL query that retrieves measurements of a certain quality (replacing wild card *\$QUALITY* with the sought quality's URI) made by a certain weather station (replacing wild card *\$WEATHER\_STATION* with the sought weather station's URI) during a given time interval (replacing wild cards *\$START\_DATETIME* and *\$END\_DATETIME* with the sought interval's starting and ending instants in datetime format).

```

PREFIX bot: <http://w3id.org/bot#>
PREFIX eep: <http://w3id.org/eep#>
PREFIX rc: <http://w3id.org/rc#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX qudt: <http://qudt.org/1.1/schema/qudt#>
PREFIX time: <http://www.w3.org/2006/time#>

SELECT ?dateTime ?value ?unit
FROM <http://tekniker.es/euskalmetWeatherStations>
WHERE {
$WEATHER_STATION bot:containsElement ?sensor .
?observation eep:madeBy ?sensor ;
eep:onQuality ?quality ;

```

```

        rc:hasTemporalContext ?time ;
        rc:hasResult ?result .
    ?quality rdf:type ?qType .
    ?result qudt:numericValue ?value ;
            qudt:unit ?unit .
    ?time time:inXSDDateTimeStamp ?dateTime

FILTER (
    ?qType = $QUALITY
    && ?dateTime > xsd:dateTime($START_DATETIME)
    && ?dateTime < xsd:dateTime($END_DATETIME) )
}

```

Listing 5.15: SPARQL query for retrieving measurements of a quality made by a weather station.

This ETL process which is available online<sup>19</sup>, is developed in Java and it is modularized with views to ease its maintainability and extension. This extensibility for other meteorological repositories or other RDF Stores is expected to be done by end users.

Thanks to this ETL process, the Euskalmet weather stations information is accessible in a SPARQL endpoint which is available online at: <http://193.144.237.227:8890/sparql>.

The EEPsA’s Transformation phase proposes generating new features by exploiting meteorological (Linked) Open Data, rather than general domain knowledge bases proposed in existing work, which is expected to further improve performance of predictive models. Thanks to the proposed ETL process, new meteorological repositories can be exploited by simply extending the “E” part. Furthermore, since the usage of this ETL process ensures the same semantic structure, the proposed parameterizable GeoSPARQL and SPARQL queries can be leveraged to access and retrieve data.

## 5.5 Data Mining

The application of data mining tasks to generate predictive models, are very dependent on the problem itself and the available data. This task involves the proper selection of data mining algorithms and their parameters. Furthermore, even in the same domain (e.g. energy efficiency in buildings), the same data mining algorithm with the same settings, may have a different performance depending on the input dataset. Therefore, incorporating semantics into algorithms to directly influence their results is a complex task and, at the moment of writing this dissertation, still an untapped field.

In the EEPsA, data enhanced in the previous KDD phases is retrieved and integrated in the data analysis environment, mainly by means of SPARQL queries.

<sup>19</sup>[https://github.com/iesnaola/EEPSA\\_ETL](https://github.com/iesnaola/EEPSA_ETL)

Furthermore, the assistance in the previous KDD phases is expected to influence this phase. On the one hand, facilitating data analysts to have the most relevant attributes for the matter at hand. On the other, by improving the quality, enriching and enlarging the available data).

## 5.6 Interpretation

Most times, tertiary buildings are complex buildings which 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 environment occupant's thermal comfort. Therefore, facility managers may require assistance to set optimal HVAC control strategies in this type of buildings.

EROSO (thERmal cOmfort SOLUTION) [173, 174] is a framework that combines KDD processes and Semantic Technologies for ensuring thermal comfort in a certain type of tertiary buildings, namely in workplaces. Specifically, EROSO supports the KDD's Interpretation phase where Semantic Technologies are used to obtain an explanation of predictive model's temperature predictions. These predictions explanations are guided by the thermal comfort regulations they satisfy. Furthermore, this result interpretation supports facility managers in the task of selecting the optimal HVAC control strategies.

There is no EU law outlining a minimum and maximum temperature permitted in workplaces. The Directive 89/654/EEC - workplace requirements<sup>20</sup> states: "during working hours, the temperature in rooms containing workstations must be adequate for human beings, having regard to the working methods being used and the physical demands placed on the workers". Some EU countries do have some more specific guidelines though. According to UKs HSE (Health and Safety Executive)<sup>21</sup>, "the law does not state a minimum or maximum temperature, but the temperature in workrooms should normally be at least 16°C or 13°C if much of the work involves rigorous physical effort". In Spain there are more strict guidelines. The Ministry of Employment and Social Security, and INSHT (Spanish Work Security and Hygiene Institute) by means of the Real Decreto 486/1997<sup>22</sup> establishes comfort temperatures between 17°C and 27°C where sedentary work takes place, and between 14°C and 25°C for light work places. In addition, the RITE (Thermal Facility Regulation in Buildings) approved by the Real Decreto 1027/2007<sup>23</sup> establishes indoor conditions between 23°C and 25°C in summer, and between 21°C and 23°C in winter.

Figure 5.2 shows an overview of the EROSO framework, which begins with the execution of a predictive model (see (1) in Figure 5.2) to forecast the temperatures for the upcoming hours within a workplace. Once these predictions are obtained, an script (see (2) in Figure 5.2) semantically annotates and stores them

<sup>20</sup><https://osha.europa.eu/en/legislation/directives/2>

<sup>21</sup><http://www.hse.gov.uk>

<sup>22</sup><http://www.boe.es/buscar/pdf/1997/BOE-A-1997-8669-consolidado.pdf>

<sup>23</sup><https://www.boe.es/boe/dias/2007/08/29/pdfs/A35931-35984.pdf>

in an RDF Store. It also executes a set of predefined ontology-driven rules to classify these temperature predictions. Afterwards, facility managers use the EROSO graphic interface to execute ontology-driven queries and retrieve the HVAC activation strategies that ensure the thermal comfort regulation they want to have at their workplace (see (3) in Figure 5.2). Finally, facility managers select and implement the optimal HVAC control strategy in their workplace’s BMS (see (4) in Figure 5.2).

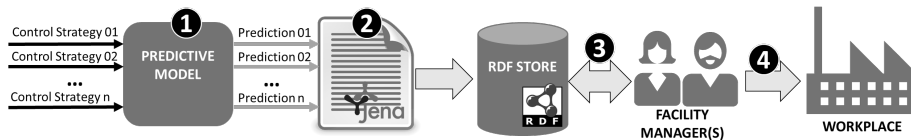


Figure 5.2: Overview of the EROSO framework.

### 5.6.1 Predictive model execution and semantic annotation

The predictive model used in EROSO receives, among other variables, an HVAC control strategy as input and it forecasts the expected temperature of the room if that HVAC control strategy is applied. This should ideally be the predictive model generated thanks to the EEPsA’s assistance in KDD phases, but it is not limited such a predictive model. The EROSO framework executes this predictive model  $N$  times, one for every different HVAC control strategy used as input. Therefore,  $N$  temperature predictions are generated.

Once these predictions are made, a script based on Apache Jena is triggered. This script annotates the predictive model itself and generated temperature predictions with the adequate EEPsA ontology’s version 1.3<sup>24</sup> terms. Listing 5.16 shows an excerpt of such an annotation. Individual *:predictiveProc20180215* represents the temperature forecasting procedure for date 2018-02-15 that has been implemented by the *:vectorLinearRegrModel* predictive model, and individual *:prediction20180214\_2300* represents the prediction made. This prediction, which has been generated on 2018-02-14 at 23:00 and forecasts temperature inside the *:openSpace* and it is related to three individuals of class *f4eepsa:ForecastResult* representing the prediction results for three specific time instants. The temperature prediction for *:openSpace* on 2018-02-15 at 15:00 is 22.5°C.

```

@prefix : <http://www.tekniker.es/openSpace#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix pep: <https://w3id.org/pep/> .
@prefix seas: <https://w3id.org/seas/> .
@prefix m3-lite: <http://purl.org/iot/vocab/m3-lite#> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix f4eepsa: <https://w3id.org/forecasting4eepsa> .
@prefix time: <http://www.w3.org/2006/time#> .
@prefix qudt: <http://qudt.org/1.1/schema/qudt#> .

```

<sup>24</sup>[https://raw.githubusercontent.com/iesnaola/eepsa/master/EEPSA\\_previousVersions/eepsa-1.3.owl](https://raw.githubusercontent.com/iesnaola/eepsa/master/EEPSA_previousVersions/eepsa-1.3.owl)

```

:predictiveProc20180215 rdf:type seas:Forecasting.
:vectorLinearRegrModel rdf:type seas:Forecaster;
    pep:implements :predictiveProc20180215.
:prediction20180214_2300 rdf:type seas:Forecast;
    pep:madeBy :vectorLinearRegrModel;
    seas:forecastsProperty m3-lite:Temperature;
    seas:forecasts :openSpace;
    prov:generatedAtTime "2018-02-14T23:00:00";
    f4eepsa:hasForecastResult :tempPredAt01pm;
    f4eepsa:hasForecastResult :tempPredAt02pm;
    f4eepsa:hasForecastResult :tempPredAt03pm.
:tempPredAt03pm rdf:type f4eepsa:ForecastResult;
    time:inXSDDateTimeStamp "2018-02-15T15:00:00";
    qudt:numericValue "22.5";
    qudt:unit m3-lite:DegreeCelsius.

```

Listing 5.16: RDF excerpt representing a predictive model and its predictions.

Since HVAC systems and their control strategies represent another main area of discourse addressed in EROSO, they are also semantically annotated. Listing 5.17 shows an excerpt of this semantic annotation result. Individual *:hvac\_Z013* represents an HVAC system acting on *:openSpace*. This HVAC is scheduled to make an HVAC control strategy which is composed of three actuations. One of those actuations (*:actuation\_20180215\_2300*) activates 6 AHUs with a flow temperature of 30°C on 2015-12-15 at 15:00.

```

@prefix : <http://www.tekniker.es/openSpace#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix pep: <https://w3id.org/pep/> .
@prefix seas: <https://w3id.org/seas/> .
@prefix m4eepsa : <http://w3id.org/measurements4eepsa#> .
@prefix qudt: <http://qudt.org/1.1/schema/qudt#> .
@prefix sosa: <http://www.w3.org/ns/sosa/> .

:hvac_Z013 rdf:type m4eepsa:HVAC;
    seas:actsOn :openSpace;
    pep:made :actuation_strat20180215.
:actuation_strat20180215 rdf:type m4eepsa:HVACControlStrategy;
    m4eepsa:hasActuation :actuation_20180215_2100;
    m4eepsa:hasActuation :actuation_20180215_2200;
    m4eepsa:hasActuation :actuation_20180215_2300.
:actuation_20180215_2300 rdf:type seas:Actuation;
    sosa:hasResult :actuation_20180215_2300_res.
:actuation_20180215_2300_res rdf:type sosa:Result;
    sosa:resultTime "2018-02-15T23:00:00";
    m4eepsa:numberOfActiveAHUs "6";
    m4eepsa:temperatureSetPoint :setPoint0084.
:setPoint0084 rdf:type sosa:Result;
    qudt:numericValue "30";
    qudt:unit m3-lite:DegreeCelsius.

```

Listing 5.17: RDF excerpt representing an HVAC system and its control strategy.



These semantic annotations, which are stored in an RDF Store, are the enablers of the rest of the EROSO components.

### 5.6.2 The ontology-driven rules

The same script that annotates aforementioned aspects, also executes a set of predefined ontology-driven rules. At the moment of writing this dissertation, there are five predefined rules representing the following workplace thermal comfort regulations: INSHT's guidelines for light workplaces, INSHT's guidelines for sedentary workplaces, HSE's guidelines, RITE's guidelines for wintertime and RITE's guidelines for summertime. These rules have been designed by domain experts and they represent knowledge regarding the thermal comfort domain. Namely, these rules classify HVAC control strategies used as predictive model input, according to the thermal comfort regulations they are forecasted to satisfy. For example, one of these predefined rules (shown in Listing 5.18) classifies HVAC control strategies forecasted to satisfy a thermal comfort regulation defined by INSHT during the working hours. This regulation determines that temperatures within a workplace where sedentary work is performed, have to be maintained between 17°C and 27°C during working hours.

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX pep: <https://w3id.org/pep/>
PREFIX qudt: <http://qudt.org/1.1/schema/qudt#>
PREFIX m3-lite: <http://purl.org/iot/vocab/m3-lite#>
PREFIX time : <http://www.w3.org/2006/time#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX eepsa: <https://raw.githubusercontent.com/iesnaola/
  eepsa/master/EEPSA_previousVersions/eepsa-1.3.owl#>
PREFIX f4eepsa: <https://w3id.org/forecasting4eepsa#>

CONSTRUCT {?hvacControlStrategy
  rdf:type eepsa:INSHTForSedentarySituation}
FROM <myGraph>
WHERE
{ ?prediction rdf:type seas:Forecast;
  pep:usedProcedure ?predictiveProcedure.
  ?predictiveProcedure pep:hasInput ?input.
  ?input f4eepsa:hasParameter ?hvacControlStrategy.
  { SELECT (COUNT(?predResult) AS ?count), ?prediction
    FROM <myGraph>
    WHERE{ ?prediction f4eepsa:hasForecastResult ?predResult.
      ?predResult qudt:numericValue ?temperatureVal.
      ?predResult time:inXSDDateTimeStamp ?dateTime.
      BIND(xsd:time(?dateTime) AS ?time).
      FILTER (
        xsd:double(?temperatureVal) >= 17 &&
        xsd:double(?temperatureVal) <= 27 &&
        ?temperatureUnit = m3-lite:DegreeCelsius &&
        xsd:time(?time) >= xsd:time(08:00) &&
        xsd:time(?time) <= xsd:time(17:00) ) }
    GROUP BY ?prediction }

```

```
FILTER (?count= 10) }
```

Listing 5.18: SPARQL Construct rule classifying HVAC control strategies forecasted to satisfy the thermal comfort regulation defined by INSHT. This rule is for a workplace where the working day starts at 8:00 and ends by 17:00.

The EROSO framework exploits rule-based knowledge instead of ontology class definitions, due to OWL2 DL's lack of expressivity to achieve the desired inferences. More specifically, SPARQL Construct rules have been used to describe this knowledge. Furthermore, every rule is parameterizable so that it can be applied to different workplaces with different working periods.

All predefined ontology-driven rules need to be parameterized once with the workplace and working periods information. Afterwards, all of them are automatically executed every time new predictions are stored in the RDF Store.

### 5.6.3 The ontology-driven queries

Once the ontology-driven rules are executed, the HVAC control strategies are classified in the RDF Store and they remain ready to be retrieved. EROSO has a single parameterizable SPARQL query to retrieve the HVAC control strategies according to the thermal comfort regulation aimed by the user. Listing 5.19 shows this parameterizable SPARQL query.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX pep: <https://w3id.org/pep/>
PREFIX seas: <https://w3id.org/seas/>
PREFIX time: <http://www.w3.org/2006/time#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX f4eepsa: <https://w3id.org/forecasting4eepsa#>

SELECT ?hvacControlStrategy
FROM <myGraph>
WHERE {
?hvacControlStrategy rdf:type $REGULATION
?predictiveProcedure pep:hasInput ?hvacControlStrategy.
?prediction pep:usedProcedure ?predictiveProcedure;
    seas:forecasts $LOCATION;
    f4eepsa:hasForecastResult ?predResult.
?predResult time:inXSDDateTimeStamp ?dateTime.
FILTER ( ?dateTime = xsd:dateTime($DATE)) }
```

Listing 5.19: SPARQL parameterizable query that retrieves HVAC control strategies forecasted to satisfy a specific thermal comfort regulation inside a location and on a date.

The execution of this SPARQL query is managed by a graphic interface that isolates facility managers from the underlying SPARQL query language in which they might not be experts, easing the interaction with the framework. The

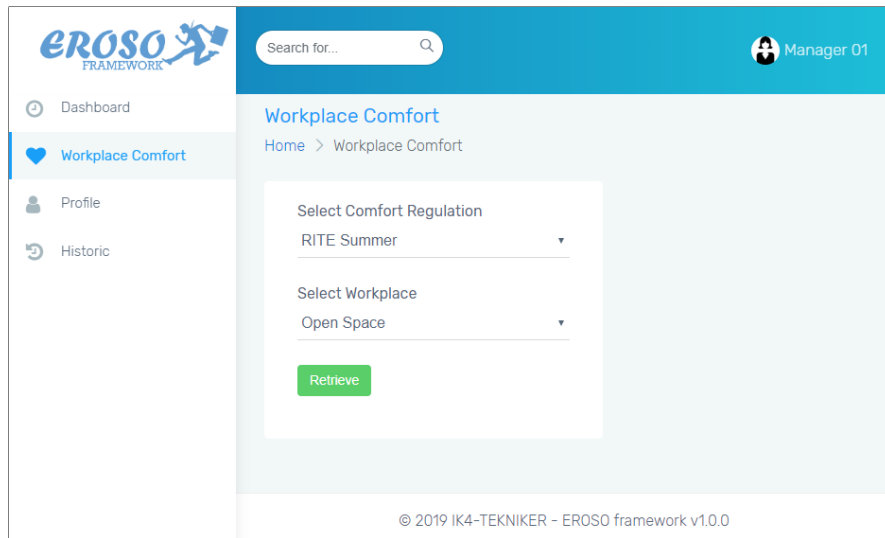


Figure 5.3: EROSO framework's interface.

graphic interface shows two dropdown lists: one with the thermal comfort regulations defined in EROSO, and the other with the locations managed by EROSO. When both the thermal regulation sought and the location where this regulation is sought are selected, after clicking the “Retrieve” button, the filling of the SPARQL query is triggered. On the one hand, the wild cards *\$REGULATION* and *\$LOCATION* are replaced by the chosen regulation's and location's corresponding URIs. On the other, *\$DATE* is replaced by the next day's date. Once the SPARQL query is complete, it is automatically executed. The HVAC control strategies forecasted to produce conditions that meet the thermal comfort selected in the dropdown list, are shown in the results section of the interface. Furthermore, for each HVAC control strategy, the user will visualize its temperature prediction both in a table (with numeric values) and plotted in a line chart (graphically). Afterwards, the facility manager chooses the optimal HVAC control strategy and implements it in the BMS. EROSO's graphic interface is shown in Figure 5.3.

The Interpretation phase is typically very resource-consuming even for domain experts, and an incorrect interpretation of the results may lead to wrong decisions. The EROSO framework exploits Semantic Technologies to ease the interpretation of data mining results in workplace thermal comfort problems.



## Chapter 6

# The EEP SA in an Office

The feasibility of the EEP SA as a whole and the different EEP SA components were tested in the IK4-Tekniker building, a technological centre constituted as a not-for-profit foundation located in Eibar (Basque Country, Spain). More specifically, a set of experiments were performed in the second floor of this building (from now on referred to as Open Space) shown in Figure 6.1. The Open Space is a single large room that acts as an office where over 250 people work on a daily basis. Regarding the usual work schedule, Monday to Thursday is split-shift and Fridays have reduced working hours.

The Open Space is equipped with different monitoring and actuating devices such as sensors developed in the European FP-7 Tibucon project<sup>1</sup> that observe temperature, humidity and illuminance at five minutes intervals. These Tibucon sensors are located both within the Open Space and outdoors and a sample of data registered by one of them is available online<sup>2</sup>. All the observations registered by Tibucon sensors are stored in a PostgreSQL<sup>3</sup> database. Furthermore, the Open Space is also equipped with eight AHUs<sup>4</sup> (Air Handling Units). It is also worth mentioning that the Open Space has a big thermal inertia due to its big dimensions, which means that it takes a long time to heat up or cool down.

IK4-Tekniker's facility manager needed a service suggesting how to activate HVAC systems in order to reach a minimum comfort temperature of 21°C at 08:00 a.m. (when the workday starts). Furthermore, the suggested HVAC control strategy had to be efficient from an energy expense point of view too. The EEP SA was therefore leveraged for developing a service based on a predictive model to satisfy the facility manager's requirements.

---

<sup>1</sup><http://www.tibucon.eu/>

<sup>2</sup><http://193.144.237.227:8890/DAV/home/dba/DataSample.csv>

<sup>3</sup><https://www.postgresql.org/>

<sup>4</sup>Air Handling Unit is an HVAC system component used to regulate and circulate air. There may be more than one AHUs associated to a single HVAC system, usually in charge of conditioning a specific space or zone.



Figure 6.1: IK4-Tekniker building's Open Space.

## 6.1 Experiments, Evaluation and Results

In this section, experiments performed to assess the EEPSA as a whole and the different EEPSA components are described. These experiments were used to evaluate the SemOD framework (explained in section 5.3.1), the predictive models generated after leveraging EEPSA's guidance through the different KDD phases (explained in Chapter 5) and the EROSO framework (explained in section 5.6). Furthermore, a set of experiments were performed aimed at demonstrating the potential of Semantic Technologies in the imputation of missing values.

### 6.1.1 The EEPSA instantiation

Prior to testing any of the EEPSA components, and instantiation of the EEPSA in the Open Space was necessary. To do so, the Semantic Annotation phase was performed. As previously stated, in an energy efficiency and thermal comfort problem in tertiary buildings there are three main information sources to be annotated: (i) the space at hand, (ii) the devices deployed in it, and (iii) the information gathered by those devices.

First of all an individual of class *bot:Building* was created to represent the IK4-Tekniker building (*:ik4tekniker*). Then, *:floor2* was created as an individual of class *bot:Storey*, and related with the building by means of the *bot:hasStorey* object property. The individual *:openSpace* belonging to class *bot:Space* was created to represent the Open Space and it was related with the *:floor2* individual by means of the *bot:hasSpace* object property. Building elements of the

Open Space were represented with individuals of classes such as *foi4eepsa:Door* or *foi4eepsa:Window* and they were related to the Open Space with the *bot:containsElement* object property. Regarding sensors and actuators deployed within the Open Space (including the ones located outdoors), they were represented with *exr4eepsa:Sensor* and *exr4eepsa:Actuator* classes respectively. A simplified RDF representation of the Open Space and its elements is available at Listing 6.1.

```

@prefix : <http://www.tekniker.es/openSpace#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foi4eepsa: <http://w3id.org/eepsa/foi4eepsa#> .
@prefix exr4eepsa: <http://w3id.org/eepsa/exr4eepsa#> .
@prefix bot: <https://w3id.org/bot#> .

:ik4tekniker rdf:type bot:Building ;
  bot:hasStorey :floor2 ;
  rdfs:comment "The IK4-Tekniker building" .

:floor2 rdf:type bot:Storey ;
  bot:hasSpace :openSpace ;
  rdfs:comment "The second storey of the IK4-Tekniker
  building" .

:openSpace rdf:type bot:BuildingSpace ;
  bot:containsElement :door1 ,
    :door2 ,
    :door3 ,
    :wall1 ,
    :wall2 ,
    :wall3 ,
    :window1 ,
    :tibuconIndoor1 ,
    :tibuconIndoor2 ,
    :tibuconIndoor3 ,
    :tibuconT17 ,
    :openSpaceHVAC ;
  rdfs:comment "Building space located at IK4-Tekniker
  building's second floor" .

:door1 rdf:type foi4eepsa:Door .

:door2 rdf:type foi4eepsa:Door .

:door3 rdf:type foi4eepsa:Door .

:wall1 rdf:type foi4eepsa:ExternalBuildingElement ,
  foi4eepsa:Wall .

:wall2 rdf:type foi4eepsa:ExternalBuildingElement ,
  foi4eepsa:Wall .

:wall3 rdf:type foi4eepsa:ExternalBuildingElement ,
  foi4eepsa:Wall .

```

```

:window1 rdf:type foi4eep sa:ExternalBuildingElement ,
          foi4eep sa:Window ;

:tibuconIndoor1 rdf:type exr4eep sa:Sensor .
:tibuconIndoor2 rdf:type exr4eep sa:Sensor .
:tibuconIndoor3 rdf:type exr4eep sa:Sensor .
:tibuconT17 rdf:type exr4eep sa:Sensor .

:openSpaceHVAC rdf:type exr4eep sa:Actuator .

```

Listing 6.1: Excerpt of an RDF representation of the Open Space.

Observations and actuations made by sensors and actuators were stored in a PostgreSQL Database. In order to semantically annotate this data with the appropriate EEP SA ontology terms, the Ontop tool<sup>5</sup> was used. Ontop is an OBDA (Ontology-Based Data Access) tool which enables mappings between relational database and an ontology [175]. It also enables to build a semantic layer, so that data can be queried with the SPARQL language while staying available as a relational database. These mappings can be implemented using the Ontop Protégé plugin. Nevertheless, Ontop's OWL 2 QL and RDFS inferencing capabilities are not enough to meet the EEP SA requirements (e.g. inferring property inclusions including property chains). Therefore, RDF assertions derived from mappings were materialized. Afterwards, these RDF assertions were complemented with other axioms to enable further inferences. A simplified RDF representation of an observation measured in the Open Space is available at Listing 6.2.

```

@prefix : <http://www.tekniker.es/openSpace#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix eep: <http://w3id.org/eep#> .
@prefix rc: <http://w3id.org/rc#> .
@prefix q4eep sa: <http://w3id.org/eep sa/q4eep sa#> .
@prefix p4eep sa: <http://w3id.org/eep sa/p4eep sa#> .
@prefix time: <http://www.w3.org/2006/time#> .
@prefix sosa: <http://www.w3.org/ns/sosa/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix qudt: <http://qudt.org/1.1/schema/qudt#> .
@prefix unit: <http://qudt.org/1.1/vocab/unit#> .

:obs_tibuconT17_20190102T2300_0T
  rdf:type exn4eep sa:Observation ;
  eep:madeBy :tibuconT17 ;
  eep:usedProcedure :tibuconSensingProcedure ;
  eep:onQuality :openSpace_OutdoorTemperature ;
  rc:hasTemporalContext :Instant20190102T2300 ;
  rc:hasGenerationTime "2019-01-02T23:00"^^xsd:dateTime ;
  rc:hasResult :res_tibuconT17_20190102T2300_0T
  rdfs:comment "tibuconT17's observation at

```

<sup>5</sup><http://ontop.inf.unibz.it/>



```

                20190102T2300 of openSpace_OutdoorTemperature" .
:tribuconSensingProcedure rdf:type p4eepsa:SensingProcedure .
:openSpace_OutdoorTemperature rdf:type
    q4eepsa:OutdoorTemperature .
:openSpace aff:influencedBy :openSpace_OutdoorTemperature .
:Instant20190102T230000 rdf:type time:Instant ;
    time:inXSDDateTimeStamp
        "2019-01-02T23:00:00Z"^^xsd:dateTimeStamp ;
    time:inXSDDate "2019-01-02"^^xsd:date ;
    time:month "01"^^xsd:integer ;
    time:hour "23"^^xsd:integer .
:res_tibuconT17_20190102T2300_OT rdf:type sosa:Result ;
    qudt:numericValue "8"^^xsd:float ;
    qudt:unit unit:DegreeCelsius .

```

Listing 6.2: Excerpt of an RDF representation of an observation measured in the Open Space.

Once the Open Space itself, the deployed devices and their observations were semantically annotated, they were loaded into Protégé and a Hermit 1.3.8.413 reasoner was executed to infer new data. All this data, including the Open Space representation and the inferred RDF assertions, was stored in a Virtuoso server 07.20.3217 version, running on an Ubuntu 14.04 Server. This RDF Store was private due to the sensitiveness of data.

## 6.1.2 SemOD framework

The SemOD framework was instantiated in the Open Space to detect temperature outliers potentially produced by a sensor's exposure to solar radiation. As mentioned before, the Open Space is equipped with Tibucon sensors which, similar to other temperature sensors, are prone to measure much higher temperatures than real ones when exposed to direct solar radiation. The SemOD framework was tested in three different Tibucon devices located outdoors and obtained results were compared with another statistical outlier detection technique.

### 6.1.2.1 Experiment design

In order to determine if an outlier detected by a given technique was an actual outlier or not, a reference dataset was used to make comparisons with. Namely, this dataset was composed of temperature observations made by an Euskalmet weather station located about 6km away from the IK4-Tekniker building and with a similar environment conditions. This weather station is equipped with a Rotronic sensor to measure temperature and these observations, which have a

Table 6.1: Tibucon sensors features and observations summary.

Sensor	Orientation	Observations	Actual Outliers	Time intervals
T17	Northwest	4,074	768	02/2016 - 07/2016
T7	Southwest	5,226	547	02/2016 - 11/2016
T23	Northwest	1,540	73	02/2017 - 05/2017

frequency of ten minutes, are available online in the Basque Open Data portal<sup>6</sup>. On average, temperature observations of Tibucon sensors located outside the IK4-Tekniker building had a deviation of 2.3°C compared with the weather station’s temperature observations. Keeping this in mind, a temperature difference of 5°C was set as a threshold to determine whether a temperature measured outside IK4-Tekniker was an actual outlier or not. That means that a temperature measured in IK4-Tekniker differing in more than 5°C compared with the reference one, was considered as an actual outlier.

Table 6.1 summarizes the features of the three Tibucon sensors located in the outside of the IK4-Tekniker building (i.e. Tibucon sensors T17, T7 and T23) in which experiments were performed. The “Observations” column determines the number of temperature observations available after sampling sensor data with a hourly frequency, the “Time Interval” column defines the time spans in which sensors made the registered observations, and the “Actual Outliers” column determines the number of temperature observations with more than 5°C of difference compared with the reference dataset.

Before using the SemOD framework, a statistical outlier detection technique was applied on the same 3 Tibucon sensors to obtain baseline results. Outliers detected by the SemOD framework were later evaluated by comparing them with baseline results. After testing different algorithms offered by Rapidminer, best results were obtained with the Detect Outlier (Densities) operator<sup>7</sup>.

In order to implement the SemOD framework (explained in section 5.3.1), once all required data was semantically annotated with proper EEPISA ontology version 1.2 terms, a HermiT 1.3.8.413 reasoner was executed to infer circumstances that make sensors susceptible to outliers. Due to the OWL axioms within the EEPISA ontology version 1.2 in which the SemOD framework is based, it was inferred that temperature sensors are susceptible to measure outliers caused by illuminance, rainfall and solar radiation among others. In these experiments, focus was placed on the outliers caused when sensors receive direct solar radiation.

The SemOD Method described in section 5.3.1.2 was applied to detect outliers potentially caused by this circumstance. The SemOD framework is based on the EEPISA ontology version 1.2, as it was the latest EEPISA ontology version available at the moment of the SemOD framework’s development. At the moment of writing this dissertation, parts of the SemOD framework already leverage the

<sup>6</sup>[http://opendata.euskadi.eus/contenidos/ds\\_meteorologicos/met\\_stations\\_ds\\_2017/opendata/2017/C075/C075\\_2017\\_1.xml](http://opendata.euskadi.eus/contenidos/ds_meteorologicos/met_stations_ds_2017/opendata/2017/C075/C075_2017_1.xml)

<sup>7</sup>[https://docs.rapidminer.com/latest/studio/operators/cleansing/outliers/detect\\_outlier\\_distances.html](https://docs.rapidminer.com/latest/studio/operators/cleansing/outliers/detect_outlier_distances.html)

latest EEPsa ontology version explained in Chapter 4. Listing 6.3 shows an excerpt of the SemOD query generated for sensor T17. The update of the rest of the SemOD framework to leverage the latest EEPsa ontology version is left as future work.

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX eep: <http://w3id.org/eep#>
PREFIX rc: <http://w3id.org/rc#>
PREFIX exn4eepsa: <http://w3id.org/eepsa/exn4eepsa#>
PREFIX q4eepsa: <http://w3id.org/eepsa/q4eepsa#>
PREFIX qudt: <http://qudt.org/1.1/schema/qudt#>
PREFIX unit: <http://qudt.org/1.1/vocab/unit#>
PREFIX time: <http://www.w3.org/2006/time#>

CONSTRUCT {?obs1 rdf:type
            exn4eepsa:TemperatureOutlierBySolarRadiation}
FROM <myGraph>
WHERE{
  ?obs1 eep:onQuality ?quality1 ;
        rc:hasTemporalContext ?instant1 .
  ?quality1 rdf:type q4eepsa:OutdoorTemperature .
  ?instant1 time:month ?month ;
            time:hour ?hour .
  ?obs2 rdf:type eep:onQuality ?quality2 ;
        rc:hasTemporalContext ?instant2 ;
        rc:hasResult ?res2 .
  ?quality2 rdf:type q4eepsa:Illuminance .
  ?instant2 time:month ?month ;
            time:hour ?hour .
  ?res2 qudt:numericValue ?val ;
        qudt:unit ?unit .
  :openSpace aff:influencedBy ?quality1 ,
              ?quality2 .

FILTER (
  ( ?month = 02 && ?hour >= 18 && ?hour <= 19
    && ?val > 15000 && ?unit = unit:Lux )
  || (...)
) }

```

Listing 6.3: SemOD Query excerpt for detecting temperature outliers caused by solar radiation based on the EEPsa ontology’s latest version available.

### 6.1.2.2 Evaluation and results discussion

Next, results obtained from experiments are discussed and evaluated. Table 6.2 summarizes these results.

The SemOD framework-enabled outlier detection technique (referred to as SemOD in this section) slightly improve specificity compared with Rapidminer’s Detect Outlier operator (from now on referred to as baseline technique), except for

Table 6.2: Summary of obtained results after applying baseline and SemOD outlier detection techniques.

Sensor	Applied Technique	Accuracy	Specificity	Sensitivity
T17	Baseline	85.7%	<b>99.6%</b>	<b>26%</b>
T17	SemOD	86.8%	<b>99.7%</b>	<b>31%</b>
T7	Baseline	91.7%	<b>99.4%</b>	<b>35.7%</b>
T7	SemOD	89.6%	<b>99.9%</b>	<b>15.7%</b>
T23	Baseline	94.3%	98.7%	17.8%
T23	SemOD	95.1%	100%	11%

the case of the T7 sensor. The remarkable outcome is that potential provenance of outliers detected by SemOD is known, while the other classical outlier detection techniques give no meaningful insight in this regard. As explained later on the case of sensor T23, this provenance provides valuable information for potential decision-making processes.

The analysis of the applied outlier detection techniques has focused on specific needs of this kind of problem: having a high specificity (detection of actual outliers, also known as recall) while not neglecting the sensitivity (normal data not being mistakenly classified as outliers).

In the case of sensor T17, both SemOD and baseline techniques have high specificity, being SemOD the one with the highest (99.7% against baseline's 99.6%). As for sensitivity, a considerable increase can be observed from 26% of outliers detected by the baseline, against the 31% detected by SemOD. Figure 6.2 plots Tibucon T17 sensor's actual outliers partitioned in four kinds: the ones detected by the SemOD, the ones detected by the baseline technique, the ones detected by both of them, and the ones undetected by any of them.

Out of the existing 768 actual outliers in the dataset, both SemOD and baseline techniques detect the same actual outliers in more than 75% of cases. It is remarkable that the baseline technique only detects outliers with high temperatures, while SemOD can detect outliers with fairly lower temperatures. For example, SemOD detects an 18°C outlier measured the 24<sup>th</sup> of April at 18:00 - a temperature value that may not seem an anomaly and could be considered as an ordinary temperature. However, this temperature was measured while sensor T17 was being hit by the sun, leading to the measurement of a much higher temperature in contrast with the 11.7°C observation made by the reference weather station. The SemOD detected this outlier in a straightforward way supported by the semantic annotation of the context, which is an essential part of the SemOD framework. Otherwise, such an outlier could be hardly detected.

Looking at Figure 6.2, it is also noticeable that most actual outliers are overlooked by both outlier detection techniques applied. It is worth mentioning that the applied SemOD Method only focused on the detection of outliers caused by sensor receiving solar radiation, and that many undetected outliers may be caused by other circumstances such as sensor's exposure to rainfall (specially during February and March).

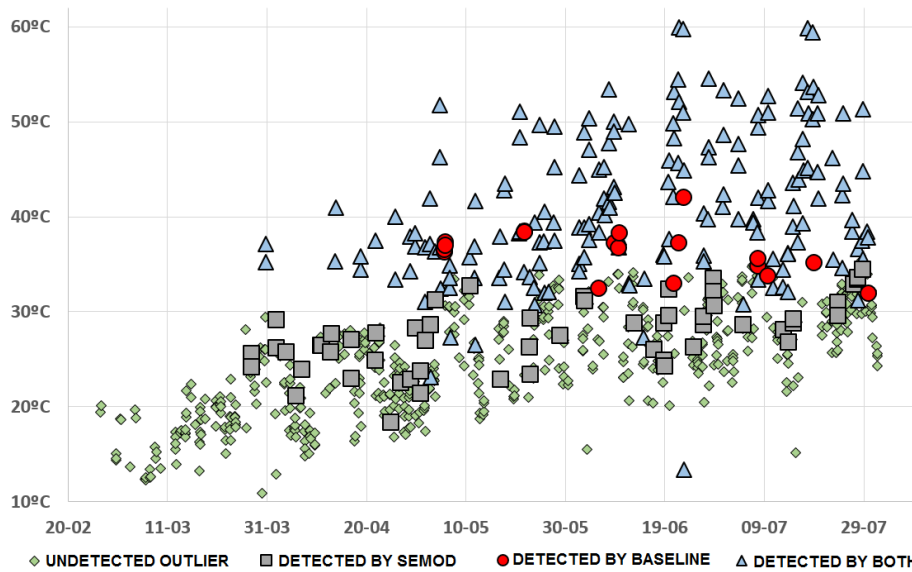


Figure 6.2: Overview of actual outliers measured by sensor T17 and their detection by different techniques.

Analysing the results of T7 sensor, SemOD obtains a slightly higher specificity than the baseline technique (99.9% against 99.4%). Nevertheless, the number of detected actual outliers dropped from baseline’s 35.7% to SemOD’s 15.7%. Results have to be interpreted by taking into consideration that SemOD only looks for outliers caused by direct solar radiation, whereas baseline looks for all of them. Consequently, it is reasonable that the baseline technique detects more actual outliers overall.

Looking at Table 6.1 obtained results for T7 might be somewhat unexpected at a first glance. The sensor is oriented to a direction that it is more exposed to sun compared with other directions (where IK4-Tekniker’s building is located, southwest-oriented objects have longer sun exposure times than northwest-oriented objects) and furthermore, it has a bigger number of observations (specifically 1,152 more observations) as it contains data from a longer period of time. In spite of these factors, less actual outliers occur comparing with sensor T17. The explanation of this case is that T7 sensor is better shielded from the sun thanks to the building’s architecture (the southwest-oriented side of the building is protected from the sun most of the year by a window overhang). Since T7 is actually less exposed to sun, less actual outliers are caused by solar radiation and that is why SemOD technique detects a low percentage (15.7%) of actual outliers. The rest of them presumably happen by other causes.

It is difficult to compare performance of different outlier detection techniques in T23 sensor’s dataset because it is very unbalanced (only 73 out of 1,540 observations, less than 5%, are actual outliers). SemOD improves accuracy when comparing with the baseline results. However, the most noteworthy thing is that sensor T23 was placed after applying SemOD Framework on T17 sensor and dis-

covering that many outliers were caused by direct solar radiation. T23 and T17 are placed few meters away from each other but T23 is strategically placed so that it is sheltered from sunshine the majority of the day. A direct comparison of both sensors' datasets cannot be made because observations for the same periods are not available. But T23 suffered from 3 times less outliers than T17 did during the same period of time the previous year. This supports our claim that spotting the potential provenance of outliers can aid in the decision-making.

### 6.1.3 Missing values

In order to see the potential of Semantic Technologies in missing values handling, a set of experiments was performed in the Open Space. More specifically, this experimentation was focused on the limitations of existing imputation methods in some specific scenarios.

#### 6.1.3.1 Experiment design

**Imputation methods.** A set of imputation methods is selected for the experimentation part. This set includes methods designed for both temporal and non-temporal data, as well as combinations of these methods. Additionally, new imputation methods have been created combining top performing imputation methods<sup>8</sup>. The list of imputation methods used follows:

1. Linear interpolation: Performs a linear interpolation between the last and first known points before and after the missing values.
2. Quadratic interpolation: Performs a quadratic interpolation between the last and first known points before and after the missing values.
3. Kalman filter-based imputation: The Kalman filter [176] is a model that can find the hidden state of a time series with white noise. This feature is exploited to impute values.
4. Kalman filter-based, polished regression imputation: Once an imputed time series is produced with the Kalman filter imputation, a regression based on similar timestamps reimputes the missing values.
5. Expectation-Maximization (EM) iterative-based, polished regression imputation: Once an imputed time series is produced with an iterative EM algorithm, a regression based solely on similar timestamps reimputes the missing values.
6. Linear interpolation-based, polished regression imputation: Analogous to the previous methods, the time series is firstly imputed using linear interpolation, then reimputed with regression.

---

<sup>8</sup>The top performing methods were selected from a preliminary experimentation which is not included in this thesis, as it is considered to be out of its scope.

In the first three methods, only the temporal component of data is taken into account<sup>9</sup>, that is, imputation is performed using only the different values of the same variable over time. The fourth and the sixth methods are temporal imputation methods that have a polished regression layer applied afterwards, therefore using all the information in the dataset to impute missing values. Finally, the fifth method does not exploit the temporal component of the data. That is, fourth, fifth and sixth methods are imputation methods whose main component is not the temporality of the data<sup>10</sup>. For further details on how these algorithms work, the reading of Garciarena’s study [177] is advised.

**Dataset benchmark.** These imputation methods have been applied on a dataset containing information collected by different sensors deployed in the Open Space, and some additional variables derived from existing data. For this experimentation, the variable to be imputed is “InTemp” (describing the indoor temperature of the Open Space). In total, the dataset has 10 variables and 1,200 observations. The data collection period starts on February 1<sup>st</sup>, 2016, ends on March 24<sup>th</sup>, 2016, and it does not suffer from missing values. The dataset is publicly available<sup>11</sup> and it is summarized in Table 6.3.

Table 6.3: Dataset benchmark summary.

Variable	Description	Source	MinValue	MaxValue
InTemp	Indoor temperature	Sensor	18.8	23.3
OutTemp	Outdoor temperature	Sensor	-0.4	21.7
OutHum	Outdoor humidity	Sensor	498.7	754.4
OutWind	Outdoor maximum wind speed	Sensor	0.0	19.6
HVAC	Air-Conditioning activation	Sensor	0	1
dT	Date and time of the observation	Sensor	01/02/2016	24/03/2016
month	Month of the observation	Derived	2	3
hour	Hour of the observation	Derived	0	23
minOcc	Minimum occupancy of the office	Derived	0	1
wDay	Type of working day	Derived	0	2

**Missing values.** The dataset used in this experimentation serves as a ground truth<sup>12</sup>, because it has no missing values. Therefore, in order to simulate a

<sup>9</sup>Henceforth, referred to as temporal imputation methods.

<sup>10</sup>Henceforth, referred to as non-temporal imputation methods.

<sup>11</sup><http://193.144.237.227:8890/DAV/home/dba/EKAW2018/Dataset.csv>

<sup>12</sup>In statistics and machine learning, it refers to data that is “known” to be correct.

missing values scenario and test the performance of the imputation methods, values have been artificially deleted. As stated in section 3.3.2, several types of missing data can be identified. For this experimentation the MCAR type is chosen. This type of missing value is the common choice when performing this kind of experiments, since most times patterns cannot be identified in real-world data [178].

The next step is to test whether the length of the missing values segments affects the performance of the imputation methods, to what extent, and from what segment length onwards. To the extent of our knowledge, no formal study in this matter has been performed at the moment of writing this dissertation. Therefore, a study for the use case dataset is made.

**Experiment methodology.** Since the dataset being treated in this experimentation part has no natural missing values, some artificial missingnesses have been injected. Therefore, performance of imputation methods can be evaluated based on the distance between the original and the imputed data value. This performance have been evaluated using different methods to compute distances. For the sake of simplicity, only results obtained using the Dynamic Time Warping [179] (DTW) measurement are shown, as they are representative of all the evaluated distances.

Missing values were only introduced in the variable where imputation would be interesting in this problem (i.e. InTemp). In every experiment run, 30% of the values in that variable were artificially deleted. To test how the length of the missing values segment affects the imputation method performance, segments of different lengths have been introduced in the data; namely segments with lengths of 1, 5, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 25, 30, 35, 40, and 45 adjacent missing values. In each experiment run, only one length of missing values sequence is introduced. However, the percentage of missing data has been kept constant, in order to maintain consistency across different runs. For example, an experiment introduced 72 missing values segments of length 5 in the dataset ( $5 \times 72 = 360$  missing values), and in another experiment run the dataset had 36 missing values segments of length 10 ( $10 \times 36 = 360$  missing values), keeping always 30% of missing values per experiment (30% of 1,200 observations = 360 observations).

Since the insertion of the missing segments is totally randomized, authors understand the necessity of repeating the experiment a significant amount of times in order to eradicate the stochastic component out of the experiments. Therefore, the experiment is repeated 30 times. The code developed for this experimentation is available on GitHub<sup>13</sup>.

### 6.1.3.2 Evaluation and results discussion

Figure 6.3 shows the mean DTW distances between the original dataset and the datasets with imputed values generated by the tested imputation methods.

<sup>13</sup><https://github.com/unaigarciarena/SemanticImputation>



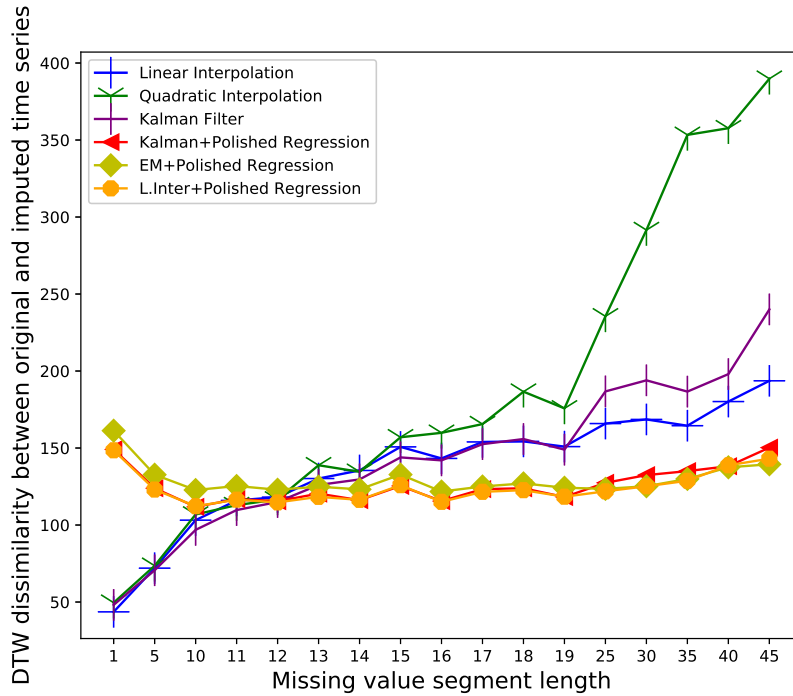


Figure 6.3: Mean DTW distance between the original and the datasets with imputed values, for all the tested imputation methods and missing segments lengths.

Furthermore, the figure captures the results for all the missing segment lengths tested in the experimentation.

Results show that, for segments of up to 12 adjacent missing values, temporal imputation methods produce considerably better results compared with the non-temporal imputation methods. However, when the missing values segment surpasses this length threshold, temporal imputation methods' performance drops (i.e. DTW distance augments) as the sequence length increases. As for the non-temporal imputation methods, their performance keeps consistent even when the missing values segment length is larger than 12. This was to be expected, as non-temporal imputation methods are not affected by the length of missing values segments, but by the total missing values on the dataset. However, in exchange for obtaining better results, these non-temporal imputation methods are more complex and time-consuming.

**Potential approaches.** Performed experiments showed that temporal imputation methods provide good quality results for missing value segments with a length of up to 12. From this length on, the larger the missing values segment, the bigger the drop in the methods' performance. In turn, non-temporal imputation

methods provide good results under these circumstances.

Non-temporal imputation methods based on regression models can lead to large time consumption if the number of observations to learn from is elevated. This time consumption aspect is presented as critical, specially when the time-series dataset comes from an on-line stream, with high frequency. And this is why finer strategies such as the polished regression emerge. In order to save time, this method calculates the euclidean distance between observations to find and use only similar observations to build a regression model. However, the calculation of the euclidean distances increases according to the number of variables, which means that it is not scalable. From a performance point of view, the bias introduced by the arbitrary selection of a distance in the polished regression algorithm can lead to non-optimal results. Furthermore, the implementation of a meaningful similarity between two data objects ought to consider contextual information [180].

A priori known information related to the problem could happen to be key in the design of new, improved imputation methods. Semantic Technologies and their capabilities to represent metadata can play a vital role towards this goal and open a range of possibilities to support the imputation of missing values.

Representing data with appropriate ontological terms can be fundamental to define new observations similarity criteria avoiding the use of simple metrics (such as the euclidean distance) and their consequent non-optimal imputation results. From a time consumption point of view, in contrast to calculating the euclidean distance in the polished regression method, the observation similarity calculation in this approach is expected to be kept constant.

Moreover, having data semantically annotated, enables further possibilities such as setting links to LOD (Linked Open Data) repositories, contributing to the enrichment of the context of the data. In addition, it will enable a more fine-grained selection of similar observations via SPARQL queries. Afterwards, different strategies could be implemented to impute missing values in a dataset. For example:

- Use the most similar data segment as-it-is to fill a missing values segment.
- Compute an appropriate function of the  $k$  most similar data observations.
- Use all the similar observations to build a regression model, following the polished regression method's idea.

Last but not least, if proposed Semantic Technologies are adequately complemented with tools that support the assistance of missing values handling, its usability and exploitation capabilities will be at hand. For example, a system could leverage these resources to recommend users the most suitable imputation method for their dataset.

In this thesis, a set of potential approaches where Semantic Technologies could contribute to handling missing values are described. Before deciding which

strategy could be implemented, extensive experimentation should be performed. For example, for the missing values imputation options, the option that offers a reasonable trade-off between time consumption and imputation quality could be selected for each use case. This experimentation is left as future work.

#### 6.1.4 EEPSA

The EEPSA data analyst assistant was instantiated in the Open Space as explained in section 6.1.1. In this section, focus is placed on the predictive models developed with the support of the EEPSA, and a set experiments are performed to evaluate them.

Keeping in mind the need of a predictive model forecasting the Open Space's temperature for the upcoming 24 hours, a baseline predictive model was developed first without the support of the EEPSA. This baseline model's results were later compared with the results of those predictive models developed after receiving EEPSA's assistance to determine if they improved and to what extent. Data spanning six months from 31<sup>st</sup> January 2016 to 1<sup>st</sup> August 2016 was sampled hourly and used to train predictive models.

In the Data Selection phase, qualities affecting indoor conditions of the Open Space were identified. According to the inferred axioms, individual *:openSpace* was an adjacent to the outside (*eepsa:AdjacentToOutsideSpace*) and a naturally enlightened (*eepsa:NaturallyEnlightenedSpace*) space. And as a result of the definition of these space subclasses, the Open Space's indoor temperature might be affected by the following qualities:

- m4eepsa:IndoorRelativeHumidity
- m4eepsa:IndoorTemperature
- m4eepsa:OutdoorRelativeHumidity
- m4eepsa:OutdoorTemperature
- m4eepsa:SpaceOccupancy
- m3-lite:CloudCover (\*)
- m3-lite:SolarRadiation (\*)
- m3-lite:SunPositionDirection (\*)
- m3-lite:SunPositionElevation (\*)
- m3-lite:WindSpeed (\*)

In order to know which of these qualities are currently available, it is necessary to instantiate and execute the SPARQL query shown in Listing 5.3 in the RDF Store where all the information related to the Open Space is stored.

```

PREFIX aff: <http://w3id.org/affectedBy#>
PREFIX : <http://www.tekniker.es/openSpace#>

SELECT DISTINCT ?affectingQuality
WHERE {
  { :openSpace_IndoorTemperature aff:affectedBy
    ?affectingQuality. }
MINUS
  { ?affectingQuality aff:belongsTo :openSpace. }
}

```

Listing 6.4: SPARQL query for retrieving qualities that affect but are not observed within the “:openSpace”.

After executing this SPARQL query in the RDF Store’s SPARQL endpoint, it was concluded that not all of these qualities were being observed in the Open Space. Namely, the qualities with an asterisk (\*) were not. Without all these qualities, predictions may not be as accurate as they could be.

The EEPSA Preprocessing phase deals with ensuring quality of available data. The EEPSA does so with the SemOD framework. This framework was applied on the *tibuconT17* sensor, which is located outdoors. Results showed that this sensor measured outliers nearly the 20% of times. This fact, together with the missing values the dataset suffered from, made the data analysts in charge of the problem decide that the outdoor temperature data had a poor quality. Since low quality data may lead to low quality results, it was decided that the information provided by this sensor (i.e. outdoor temperature of the Open Space) should be replaced by a higher quality data source.

After applying the aforementioned KDD Data Selection and Preprocessing tasks, it was concluded that, among the relevant qualities suggested by the EEPSA, the following could not be used because they were not being measured or because they had a very poor data quality to be confidently used. Namely, the set of qualities followed:

- m4eepsa:OutdoorTemperature
- m3-lite:CloudCover
- m3-lite:SolarRadiation
- m3-lite:SunPositionDirection
- m3-lite:SunPositionElevation
- m3-lite:WindSpeed

Within the KDD Transformation phase, the EEPSA focuses on the feature generation task in order to obtain qualities affecting the indoor temperature of a space shown in the previous list. Even though this task is intended for qualities that are not currently being measured, it can also be used for qualities that

Table 6.4: Closest Euskalmet weather stations to IK4-Tekniker building measuring outdoor temperature (results obtained after executing SPARQL query shown in Listing 6.5 the 28/01/2019).

stationID	stationName	distanceToBuilding
"C075"	"Eitzaga"	5.92932
"C0D3"	"Aixola (Embalse)"	7.03675
"C078"	"Altzola (Deba)"	8.02639
"C0BD"	"Iruzubieta"	11.217
"C0D2"	"San Prudentzio (Deba)"	12.0478

are being observed but for certain reason (e.g. inconsistent or noisy data) are low quality data. In the Open Space case, the outdoor temperature data was considered low quality data due to its outliers and missing values. Furthermore, the rest of the listed qualities were not being measured by the deployed devices. Therefore, the EEPSA feature generation task was applied to retrieve these qualities.

As explained in section 5.4, the EEPSA leverages weather stations regulated by Euskalmet. Among the aforementioned list of relevant qualities that are not available, these weather stations monitor outdoor temperature and wind speed information. Therefore, the EEPSA feature generation task was applied for both qualities<sup>14</sup>. For the sake of simplicity, only the outdoor temperature's case is described next.

The first step to retrieve the outdoor temperature variable from a nearby Euskalmet weather station, consisted in checking the weather stations nearby the Open Space measuring this quality. To do so, the data analyst instantiated the parameterizable GeoSPARQL query shown in Listing 5.14. The resulting GeoSPARQL query (shown in Listing 6.5) was later executed in the SPARQL endpoint of the RDF Store containing Euskalmet weather stations information<sup>15</sup>. The execution of this query returned a set of weather stations measuring outdoor temperature, sorted by proximity to the Open Space, as shown in Table 6.4. However, other factors than the distance could influence on the election of one or another weather station (e.g. the altitude where the sensing device is installed). This information was also represented as part of the Euskalmet weather stations' information. After evaluating all these factors, it was concluded that the weather station named "Eitzaga" was the most suitable one due to the similarity of its environment conditions.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX q4eepsa: <https://w3id.org/eepsa/q4eepsa#>
PREFIX bot: <https://w3id.org/bot#>
```

<sup>14</sup>At the moment of performing this experiment, the rest of qualities were not available in nearby weather stations.

<sup>15</sup><http://193.144.237.227:8890/sparql>

```

PREFIX aemet: <http://aemet.linkeddata.es/ontology/>
PREFIX eep: <https://w3id.org/eep#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT DISTINCT ?stationID ?stationName
(bif:st_distance((bif:st_point(xsd:float(?lat),
xsd:float(?long))),
(bif:st_point(xsd:float(43.19), xsd:float(-2.45)))))
AS ?distanceToBuilding
FROM <http://tekniker.es/euskalmetWeatherStations>
WHERE {
?weatherStation rdf:type aemet:WeatherStation;
foaf:name ?stationName;
geo:lat ?lat;
geo:long ?long;
dc:identifier ?stationID;
bot:containsElement ?sensor.
?sensor eep:forQuality ?quality.
?quality rdf:type ?qType.

FILTER (
?qType = q4eepsa:OutdoorTemperature )
}
ORDER BY ?distanceToBuilding
LIMIT 5

```

Listing 6.5: GeoSPARQL query for retrieving IK4-Tekniker building nearby weather stations measuring temperature.

Once the data analyst decided which was the suitable weather station to retrieve the data from, the parameterizable SPARQL query shown in Listing 5.15 was instantiated. This SPARQL query allows retrieving observations made by a weather station during a given period of time. The data analyst determined the weather station (i.e. “Eitzaga”’s URI), the quality (i.e. outdoor temperature) and the time span of the information sought (i.e. the time span between 31<sup>st</sup> January 2016 and 1<sup>st</sup> August 2016), which resulted in Listing 6.6. Afterwards, this SPARQL query was executed over the RDF Store where this information was previously saved<sup>16</sup>. The query returned the outdoor temperature values measured in the selected weather station for the specified period of time.

```

PREFIX : <http://www.tekniker.es/euskalmetWeatherStations#>
PREFIX bot: <https://w3id.org/bot#>
PREFIX eep: <https://w3id.org/eep#>
PREFIX rc: <http://w3id.org/rc#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX qudt: <http://qudt.org/1.1/schema/qudt#>
PREFIX time: <http://www.w3.org/2006/time#>
PREFIX q4eepsa: <https://w3id.org/eepsa/q4eepsa#>

SELECT ?dateTime ?value ?unit

```

<sup>16</sup>The ETL process was previously executed to extract, annotate and store this weather station’s observations

```

FROM <http://tekniker.es/euskalmetWeatherStations>
WHERE {
  :weatherStation_Euskalmet_c075 bot:containsElement ?sensor .
  ?observation eep:madeBy ?sensor ;
    eep:onQuality ?quality ;
    rc:hasTemporalContext ?time ;
    rc:hasResult ?result .
  ?quality rdf:type ?qType .
  ?result qudt:numericValue ?value ;
    qudt:unit ?unit .
  ?time time:inXSDDateTimeStamp ?dateTime

FILTER (
  ?qType = q4eepsa:OutdoorTemperature
  && ?dateTime > xsd:dateTime(2016-01-31T00:00:00Z)
  && ?dateTime < xsd:dateTime(2016-08-01T00:00:00Z) )
}

```

Listing 6.6: SPARQL query for retrieving observations of a quality made by a weather station.

The same process feature generation task was followed for the wind speed information, which was suggested to be also relevant for the matter at hand in the Data Selection phase.

The outdoor temperature and wind speed information retrieved from the Euskalmet weather station, alongside with the already existing data was used in the following Data Mining phase. In this case, the RapidMiner Studio 7.1 version<sup>17</sup> was used alongside with the Linked Open Data extension<sup>18</sup>. Within this extension, the operator SPARQL Data Importer was used to query the RDF Store and retrieve the information needed. The Series extension<sup>19</sup> was also used in order to work with the available data. This enabled having a richer dataset for developing a predictive model with a better performance (as shown in section 6.1.4.2) to forecast Open Space's temperature for the upcoming 24 hours.

#### 6.1.4.1 Experiment design

As previously stated, a baseline predictive model was developed without the support of the EEPsA throughout the KDD process. Different predictive models were built using different algorithms with different combinations of available variables and fine-tuning the parameters for their window sizes. Best results were obtained with a model built with Rapidminer's Vector Linear Regression algorithm<sup>20</sup> and containing a window of 553 features: the last 504 hours (21 days)

<sup>17</sup><https://docs.rapidminer.com/7.6/studio/releases/7.1/>

<sup>18</sup>[https://marketplace.rapidminer.com/UpdateServer/faces/product\\_details.xhtml?productId=rmx\\_lod](https://marketplace.rapidminer.com/UpdateServer/faces/product_details.xhtml?productId=rmx_lod)

<sup>19</sup>[https://marketplace.rapidminer.com/UpdateServer/faces/product\\_details.xhtml?productId=rmx\\_series](https://marketplace.rapidminer.com/UpdateServer/faces/product_details.xhtml?productId=rmx_series)

<sup>20</sup>[https://docs.rapidminer.com/studio/operators/modeling/predictive/functions/vector\\_linear\\_regression.html](https://docs.rapidminer.com/studio/operators/modeling/predictive/functions/vector_linear_regression.html)

Table 6.5: Predictive models and the variables used to build them.

Model	IT	OT	OH	WS	HVAC	OCC	Date
Baseline	3 Tibucon	1 Tibucon			OpenSpace		1 var
EEPSA#1	3 Tibucon	1 Tibucon	1 Tibucon		OpenSpace	2 vars	4 vars
EEPSA#2	3 Tibucon	Euskalmet	1 Tibucon		OpenSpace	2 vars	4 vars
EEPSA#3	3 Tibucon	1 Tibucon	1 Tibucon	Euskalmet	OpenSpace	2 vars	4 vars
EEPSA#4	3 Tibucon	Euskalmet	1 Tibucon	Euskalmet	OpenSpace	2 vars	4 vars

indoor temperature values, last 24 hours values for outdoor temperature, last 24 hours HVAC values, and another one for the date time.

For the EEPISA-enabled models, the available data pool became richer and larger thanks to the EEPISA’s assistance. Algorithm and variable selection and their window sizes were fine tuned to develop a model that accurately predicts Open Space’s indoor temperatures for the upcoming 24 hours. The most accurate model was built with Rapidminer’s Vector Linear Regression algorithm containing last 168 hours (7 days) indoor temperature values, last 24 hours values for outdoor temperature, outdoor humidity, outdoor wind speed and HVAC status, 2 features to describe current space occupancy, and 4 features describing the date (month, hour, day of the week and date time). Table 6.5 shows the input data used by some of the most accurate predictive models developed with and without the support of the EEPISA<sup>21</sup>. In the table, IT stands for indoor temperature, OT for outdoor temperature, OH for outdoor humidity, WS for wind speed and OCC for occupancy.

#### 6.1.4.2 Evaluation and results discussion

Performance of the forecasters was characterized by two statistical metrics: the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). Measures based on percentage errors (e.g. Mean Absolute Percentage Error, MAPE) were dismissed because of their disadvantage of being infinite or undefined if data is zero, and having extreme values when close to zero. Therefore, a percentage error makes no sense when measuring the accuracy of temperature forecasts on the Fahrenheit or Celsius scales [181]. Predicted indoor temperatures for the future 24 hours in the Open Space had a MAE of 0.80°C and a RMSE of 0.99°C for the baseline model, and a MAE of 0.57°C and a RMSE of 0.70°C for the EEPISA-enabled EEPISA#4 model. Figure 6.4 shows an overview of the Rapidminer process.

<sup>21</sup>Blank spaces mean that no variable was used, and “var(s)” is a contraction for variable(s)



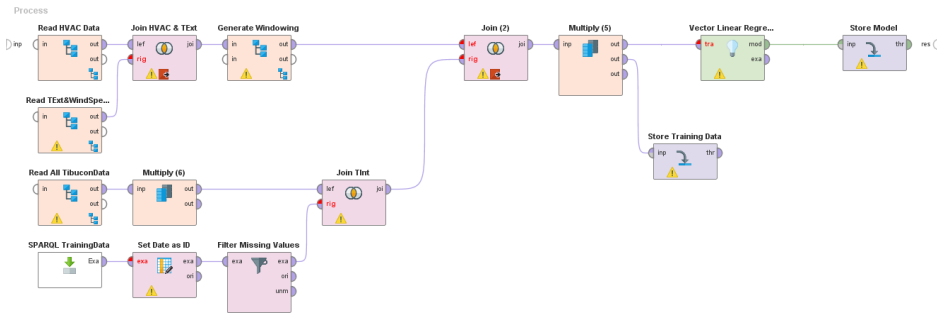


Figure 6.4: Rapidminer process of the baseline predictive model.

Results show that the predictive model EEP SA#4 reduced the MAE and RMSE by over 28% ( $0.23^{\circ}\text{C}$  in MAE and  $0.29^{\circ}\text{C}$  in RMSE) compared with the baselin, which could yield a more energy-efficient control [182]. Table 6.6 shows the MAE and RMSE obtained after applying different EEP SA-enabled models.

Table 6.6: MAE and RMSE obtained with different predictive models enabled by the EEP SA (best results were obtained with EEP SA #4).

Model	MAE (all days)	RMSE (all days)	MAE (reduced working hour)	RMSE (reduced working hour)
EEPSA #1	$0.63^{\circ}\text{C}$	$0.77^{\circ}\text{C}$	$0.67^{\circ}\text{C}$	$1.10^{\circ}\text{C}$
EEPSA #2	$0.60^{\circ}\text{C}$	$0.74^{\circ}\text{C}$	$0.57^{\circ}\text{C}$	$0.91^{\circ}\text{C}$
EEPSA #3	$0.61^{\circ}\text{C}$	$0.74^{\circ}\text{C}$	$0.64^{\circ}\text{C}$	$1.02^{\circ}\text{C}$
EEPSA #4 (*)	$0.57^{\circ}\text{C}$	$0.70^{\circ}\text{C}$	$0.56^{\circ}\text{C}$	$0.85^{\circ}\text{C}$

The Data Selection phase of the EEP SA suggested the incorporation of some variables such as wind speed to build the predictive model. The incorporation of this variable in the predictive model (which may be overlooked by a data analyst not expert in the domain), reduced MAE by 5%.

Thanks to the SemOD framework applied in the data preprocessing phase, anomalous temperature observations were detected in the data registered by the Tibucon sensor located outdoors. Furthermore, replacing the outdoor temperature data provided by the Tibucon sensor (considered to be low quality data) with a higher quality outdoor temperature source (a nearby weather station), MAE was reduced by 6%, and even by nearly 13% in some specific days (namely in days with reduced working hours).

Within the available data, a day that did not follow the expected work schedule was found. Specifically, the 23<sup>rd</sup> March 2016 (Wednesday) was a reduced hours workday, when typically it should have been a split shift schedule. This happened because in 2016, Easter started the 24<sup>th</sup> March. For the temperature prediction of this day, the EEP SA-enabled model reduced MAE by 44% ( $0.28^{\circ}\text{C}$ ) and RMSE by 45% ( $0.38^{\circ}\text{C}$ ) compared with the baseline model results. As long as more data is available, it should be analysed to which extent EEP SA-enabled models reduce prediction errors in days with atypical work schedule. This is left as future work.

### 6.1.5 EROSO framework

The EROSO framework was instantiated and tested in the Open Space, using the predictive model EEPsA#4 (explained in section 6.1.4.1). During the testing period, the forecasting process was automatically executed daily at 17:00. For each execution, 20 different HVAC control strategies were used as inputs of the predictive model, so 20 different temperature predictions were obtained. This way, the facility manager could decide which HVAC control strategy to implement in the Open Space, in order to ensure next day's thermal comfort. This EROSO instantiation was compared with an already existing solution implemented in the Open Space, known as OSCS (Open Space Comfort Solution).

#### 6.1.5.1 Experiment design

The OSCS also makes use of the predictive model EEPsA#4 and it also uses the same 20 HVAC control strategies as inputs to make temperature predictions. Currently, the OSCS seeks to comply with just one thermal comfort regulation, which consists in having a temperature over a predefined threshold of 21°C when the working day starts at 8:00<sup>22</sup>. Furthermore, the OSCS automatically selects the first HVAC control strategy found predicting the predefined comfort regulation. That is, even though 20 HVAC strategies are available to make forecasts, when a prediction fulfils the defined thermal comfort requirement, the forecasting process stops. The found strategy is then stored on a PostgreSQL database, alongside with the temperatures forecasted to produce during the next 24 hours. The OSCS offers a graphic interface where these stored temperatures are graphically depicted in a line chart (as shown in Figure 6.5). Nevertheless, many times facility managers have expressed their difficulties at trying to figure out in the line chart which temperature corresponds to a given instant.

#### 6.1.5.2 Evaluation and results discussion

In order to compare the EROSO implementation in the Open Space with the OSCS, the following criteria were evaluated:

- Usability: The quality of the interaction and facility managers' overall satisfaction with the system. It is measured via a survey and interviews.
- Extensibility: The ability of the system to be extended with additional functionalities or modifying existing ones (e.g. adding new thermal comfort regulations).
- Thermal comfort: The duration of periods when comfortable thermal situations occur during the working day.

---

<sup>22</sup>Due to the characteristics of the Open Space, it was assumed that once this temperature was achieved at the beginning of the working day, a comfortable thermal situation would be maintained throughout the working day. However, it has been proved that when certain outdoor conditions are given, this is not true.

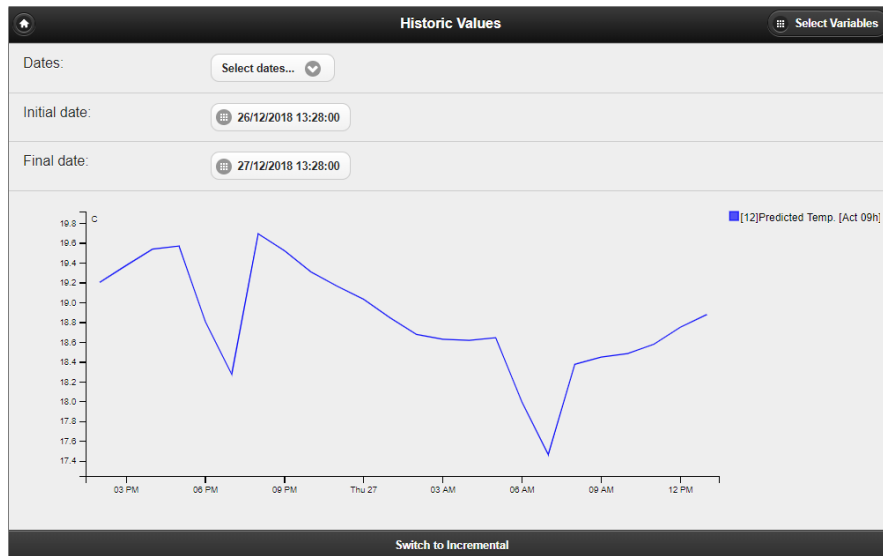


Figure 6.5: OSCE's interface.

**Usability.** IK4-Tekniker's facility manager and two workers were surveyed with the SUS (System Usability Scale) scale [183] after testing the EROSO framework. The average score obtained was 77.5 out of 100, so it can be concluded that overall interaction with EROSO framework is good. The feedback received in the interviews indicated that having different thermal comfort regulations available in the system, provides users with a bigger flexibility to choose the adequate HVAC control strategy for each situation. This aspect was highlighted by three interviewees, who foresee it very important when managing a workplace that may host different events (e.g. IK4-Tekniker building's Auditorium where at the moment of writing this dissertation EROSO is being implemented) or a space with changing requirements. The possibility of selecting different thermal comfort regulations in EROSO is enabled by Semantic Technologies.

**Extensibility.** Being based on Semantic Technologies, the EROSO framework is easier to extend or modify compared with the OSCE. For example, if a new predictive model is added to the system to make predictions, the OSCE needs to modify, compile and deploy the solution, as well as performing some tasks in the database to register the new predictive model. By contrast, EROSO just needs to add a new instance of the already existing class *seas:Forecaster* in its RDF Store. Furthermore, if the facility manager has other comfort needs or criteria, it would be enough to define a SPARQL construct rule and a class representing that regulation as a subclass of *eepta:ThermalComfortRegulation*. On the contrary, OSCE has just one comfort criterion (exceeding a threshold temperature at 8:00) and adding more comfort criteria would mean a modification of the source code, its recompilation and deployment.

Table 6.7: Comparison of mean discomfort duration per day (according to RITE regulation for winter days and INSHT regulation) suffered if HVAC control strategies proposed by EROSO and the OSCS were applied.

Framework	RITE winter regulation	INSHT regulation
EROSO	0 h 00 min	0 h 00 min
OSCS	2 h 48 min	0 h 00 min

**Thermal comfort.** The overall thermal comfort achieved by the HVAC control strategies proposed by EROSO and the OSCS have been compared. For that purpose, for each of the proposed HVAC control strategy, predicted temperatures for the Open Space have been recorded during 15 working days (from 5<sup>th</sup> to 25<sup>th</sup> February 2018). For each prediction, it has been calculated the amount of time that would not meet a certain thermal comfort regulation. That is, the amount of time when, if implementing the HVAC control strategies proposed by the different frameworks, the Open Space temperature would not be between the values defined by a certain regulation. For this experiment two thermal comfort regulations have been used: RITE (Spanish Buildings' Thermal Installation Regulation) for winter days (between 21°C and 23°C during working hours) and INSHT (Spanish Work Security and Hygiene Institute) for sedentary work (between 17°C and 27°C during working hours). Results show that EROSO does not propose any HVAC control strategy that does not predict a temperature fulfilling the aimed regulation's temperature requirements. Although the HVAC control strategies proposed by OSCS fulfil INSHT regulation, there is a mean duration of 2 hours and 48 minutes when they do not ensure a temperature that fulfils RITE's regulation. Table 6.7 summarizes this experiment's results. Thanks to the aforementioned flexibility enabled by the Semantic Technologies, EROSO users can seek different thermal regulations and the system recommends different HVAC control strategies accordingly. This flexibility is valuable because different thermal comfort regulations may be necessary even for the same space. For example when committing to RITE regulation, which defines different thermal requirements depending on the season of the year.

## Chapter 7

# The EEPISA in a Poultry Farm

World population is growing at exponential rates and, according to United Nation's 2017 Revision of World Population Prospects<sup>1</sup>, it is projected to reach a number of over 9.7 billion people by 2050. This growth poses issues that may affect the sustainability of demographic, social, and economic system. One of the main challenges related to this population growth consists in finding a way to feed all these people and the agriculture, which can be understood as the cultivation and breeding of animals and plants to provide food and other products to sustain and enhance life, plays a vital role in tackling this challenge.

As a consequence of the aforementioned population growth, world meat consumption is also expected to grow by 70% in the period of 2000-2030 and by 120% in the period of 2030-2050. The meat sector is one of the most important ones at a worldwide agriculture level and so it is in Europe. According to Eurostat<sup>2</sup> there are almost 7 million livestock farms in the EU, and the four main types of farms are the ones rearing pigs, bovine animals, poultry, and sheep and goats. However, in order to satisfy the foreseen meat demand, there is a dire need to increase meat production.

This meat production improvement cannot be done at whatever cost though, as maintaining the health and welfare status of animals at optimal levels is one of the farmers' main concerns. Comfort within farms is one of the main factors that influence the wellbeing and health of animals during their lifetime [184], hence it cannot be neglected. Providing a comfortable environment within farms not only enables maximizing each animal's profit, but also reduces mortality, which in turn allows to lessen the amount of wasted water and feed resources. Anyway, comfort requirements may vary depending on the species and their growth phase.

---

<sup>1</sup><https://esa.un.org/unpd/wpp/>

<sup>2</sup>[http://ec.europa.eu/eurostat/statistics-explained/index.php/Meat\\_production\\_statistics](http://ec.europa.eu/eurostat/statistics-explained/index.php/Meat_production_statistics)



Figure 7.1: Use case poultry farm.

In the context of the Internet of Food & Farm 2020 (from now on referred to as IoF2020) European H2020 project<sup>3</sup>, 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 (shown in Figure 7.1) 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. Figure 7.2 shows the default thermal comfort requirements for poultry flocks with an ordinary growth pace. These comfort requirements may vary if, for certain reasons, the poultry growth is slower or faster than expected. Furthermore, farm's building structure, thermal inertia, and outside climatic conditions have a direct effect on the indoor climatic conditions.

Currently, farmers' behaviour with respect to poultry welfare is reactive. That is, when an uncomfortable conditions occurs in the farm, they take measures to revert the situation and try to ensure a comfortable situation. Farmers could benefit from a system that lets them know if future farm indoor conditions will not meet animals comfort conditions. This system could lead to a paradigm change, making farmers more proactive with views to ensuring poultry welfare, and allowing them to act upon the farm before uncomfortable and harmful situations happen. 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 EEPsA (explained in Chapter 5).

---

<sup>3</sup><https://www.iof2020.eu/>

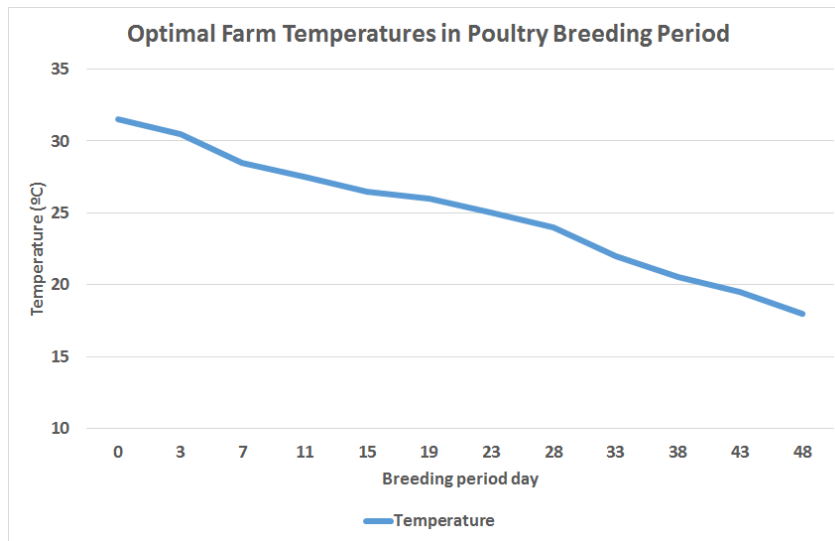


Figure 7.2: Optimal poultry farm temperatures through a breeding period.

## 7.1 Requirements

The EEPSA is designed to assist data analysts in energy efficiency and thermal comfort problems in tertiary buildings. Furthermore, one of its aims is to be applicable to similar problems in different types of buildings, so the EEPSA was designed bearing this reusability feature in mind. However, each use case may have its own requirements, and the EEPSA may need some additional customization tasks to be applicable to those use cases. Therefore, every time a new use case is faced, it is necessary to identify the use case requirements, and see to which extent EEPSA 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. Furthermore, variables that may not measured in tertiary buildings are necessary to be monitored in farms (e.g. ammonia levels). Likewise, in these contexts there is 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 qualities that affect farm's indoor climatic conditions. Summarizing, the system developed for the poultry farm use case at hand, should support data analysts in answering competency questions like the following:

- CQ01: How many breeding spaces are in the farm?
- CQ02: How many troughs are in a given space?
- CQ03: What is the stocking density in a given space?

- CQ04: What is the CO<sub>2</sub> level in a given space?
- CQ05: What are the devices installed within a farm?
- CQ06: What are the variables affecting the temperature of a breeding space?

This knowledge is important, and the EEPSA ontology<sup>4</sup> needs to properly capture it to address the aforementioned requirements. The FoI4EEPSA ontology module<sup>5</sup> describes buildings and building elements, but this knowledge is centred in tertiary buildings. Therefore, some terminology specific to poultry farms elements (e.g. breeding structures) is not covered. The Q4EEPSA ontology module<sup>6</sup> in charge of representing qualities of spaces, does not describe qualities that are typical in poultry farms, such as the stocking density representing the amount of kilos contained within a space. The EXR4EEPSA ontology module<sup>7</sup> in charge of representing sensors and actuators among others, lacks of the description of some agents monitoring or acting on farms conditions. As for the EXN4EEPSA<sup>8</sup>, which represents executions made by sensors, actuators and other agents, it does not represent circumstances that may compromise animals' welfare. Furthermore, representation of spaces such as breeding spaces and qualities affecting their environmental conditions are also missing in the EK4EEPSA ontology module<sup>9</sup>, as they are far from a tertiary building's casuistry. Therefore, the EEPSA ontology needs to be adequately extended and customized to tackle these issues that are currently uncovered.

The EEPSA ontology is not the only EEPSA component that may require a customization task. Although the set of resources and tools offered by the EEPSA are designed to be usable in different use cases, they may need to be extended or customized when the use case requires so. More specifically, EEPSA's ETL process (explained in section 5.4) and EROSO framework (explained in section 5.6) have to be customized.

In the KDD Transformation phase, the EEPSA implements an ETL process with a threefold objective: (i) to extract both weather stations metadata regulated by Euskalmet (Basque Meteorology Agency) and their observations from the Basque Open Data portal, (ii) to semantically annotate them using the adequate ontology terms, and (iii) to load them into a Virtuoso RDF Store. This RDF Store is expected to be accessed later on by the data analyst via SPARQL queries to retrieve a weather station's information or the observations registered by the desired weather station. However, the use case poultry farm is not within the territory covered by Euskalmet. This means that there is no weather station close to the target farm<sup>10</sup> and therefore none of Euskalmet weather stations can be confidently used to retrieve meteorological information.

---

<sup>4</sup><https://w3id.org/eepsa>

<sup>5</sup><https://w3id.org/eepsa/foi4eepsa>

<sup>6</sup><https://w3id.org/eepsa/q4eepsa>

<sup>7</sup><https://w3id.org/eepsa/exr4eepsa>

<sup>8</sup><https://w3id.org/eepsa/exn4eepsa>

<sup>9</sup><https://w3id.org/eepsa/ek4eepsa>

<sup>10</sup>The closest weather station is more than 250 km away.



Regarding the KDD Interpretation phase, the EROSO framework proposed by EEPISA contains a set of ontology-driven rules designed by domain experts. These rules classify temperature predictions according to the workplace thermal comfort regulations they satisfy. Nevertheless, poultry comfort needs differ from human thermal comfort needs in workplaces. Furthermore, the EROSO framework is geared towards the suggestion of optimal HVAC control strategies for ensuring users comfort. Meanwhile, the addressed farm requires a notification system to alert farmers from potential undesirable indoor conditions.

Summarizing, the EEPISA ontology, the EEPISA ETL process and the EROSO framework need to be customized to address the poultry farm use case at hand.

## 7.2 The EEPISA Customization

In this section, the customization process of the different EEPISA components is outlined. Thanks to the EEPISA's generic design, the rest of the EEPISA components can be reused as-they-are and additional customization tasks are unnecessary.

### 7.2.1 The EEPISA ontology customization

Bearing in mind the existing gap between the current EEPISA ontology and the poultry farm use case requirements, the NeOn Methodology's 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<sup>11</sup> [185] is an ontology repository for the agronomy domain which features ontology hosting, search, versioning, visualization, comment, and recommendation. Planteome<sup>12</sup> [186] 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<sup>13</sup> is a catalogue of data standards of different types and formats for the agri-food domain.

AGROVOC<sup>14</sup> is a thesaurus that organizes concepts related to the FAO<sup>15</sup> (Food and Agriculture Organization of the United Nations) including agriculture, food, nutrition, fisheries, forestry and environment. At the moment of writing this dissertation, AGROVOC consists of over 35,000 concepts and it is available in 23 different languages. Furthermore, there are ontologies covering

---

<sup>11</sup><http://agroportal.lirmm.fr>

<sup>12</sup><http://browser.planteome.org/amigo>

<sup>13</sup><http://vest.agrisemantics.org>

<sup>14</sup><http://aims.fao.org/en/agrovoc>

<sup>15</sup><http://www.fao.org>

different aspects of the agricultural domain. The Food ontology<sup>16</sup> [187] 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 [188] describes the crop production life cycle and the FTTO [189] (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 was performed.

The representation of poultry farms and related elements is tackled in the new FoI4PFEEPSA ontology module<sup>17</sup>. 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 concept as subclass of “buildings” while the latter concept is defined as subclass of “housing”. Such a fine-grained distinction between buildings and housings does not fit with the conceptualization of BOT reused in the EEPsA ontology. Therefore, these two AGROVOC terms are not reused. Instead, a new *foi4pfeepsa:PoultryHousing* class is created and defined as subclass of the new *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* for water dispensers, and *foi4pfeepsa:Trough* representing food dispensers. The representation of these concepts is important to ease other predictions such as bird density distribution within farms. It is worth mentioning that the FoI4PFEEPSA ontology module is aligned with AGROVOC in a separate file available online<sup>18</sup>.

The stocking density of a space represents the amount of kilos contained in such 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<sup>19</sup> 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<sup>20</sup> 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<sup>21</sup> with the definition of classes representing different levels of poultry discomfort: moderate

<sup>16</sup><https://www.bbc.co.uk/ontologies/fo/1.1.ttl>

<sup>17</sup><https://w3id.org/pfeepsa/foi4eepsa>

<sup>18</sup>[https://iesnaola.github.io/eepsa/PFEEPSA/FoI4PFEEPSA/alignments/foi4pfeepsa\\_Alignment\\_AGROVOC.owl](https://iesnaola.github.io/eepsa/PFEEPSA/FoI4PFEEPSA/alignments/foi4pfeepsa_Alignment_AGROVOC.owl)

<sup>19</sup><https://w3id.org/pfeepsa/q4pfeepsa>

<sup>20</sup><https://w3id.org/pfeepsa/exr4pfeepsa>

<sup>21</sup><https://w3id.org/pfeepsa/q4pfeepsa>

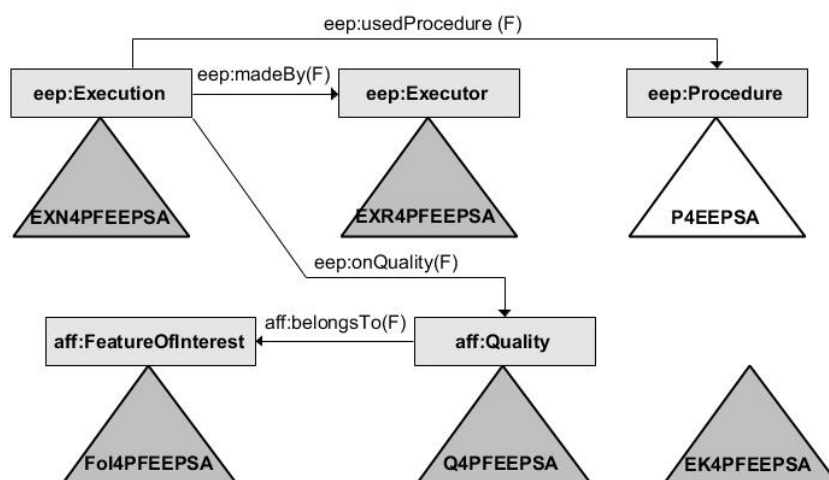


Figure 7.3: Overview of the ontology modules replaced by the EEPISA ontology's customization for Poultry Farm domain.

(*exn4pfeepsa:ModerateThermalDiscomfort*), high (*exn4pfeepsa:HighThermalDiscomfort*) and severe (*exn4pfeepsa: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 ontology module<sup>22</sup> extended the original EK4EEPSA ontology module, including the description of a class *ek4pfeepsa:BreedingSpace* and variables affecting its indoor conditions.

Figure 7.3 depicts the ontology modules that conform the EEPISA ontology's customization for poultry farms named the PFEEPSA (Poultry Farm Energy Efficiency Prediction Semantic Assistant) ontology<sup>23</sup>. An excerpt of the RDF model for the poultry farm at hand is available online<sup>24</sup>.

## 7.2.2 The EEPISA Transformation phase customization

In the KDD Transformation phase, the EEPISA leverages weather stations regulated by Euskalmet with an ETL process (explained in section 5.4). Euskalmet manages weather stations installed all over the Basque Country territory. However, the use case farm is not located in this territory, which means that there are no weather stations near the use case farm where meteorological data can be retrieved from. Therefore, the EEPISA's KDD Transformation phase support was extended for the set of weather stations managed by AEMET<sup>25</sup> (Spanish Meteorology Agency). AEMET regulates weather stations throughout the Spanish territory, including the use case poultry farm's region.

<sup>22</sup><https://w3id.org/pfeepsa/ek4pfeepsa>

<sup>23</sup><https://w3id.org/pfeepsa>

<sup>24</sup><https://raw.githubusercontent.com/iesnaola/pfeepsa/master/examples/poultryFarmExample.ttl>

<sup>25</sup><http://www.aemet.es>

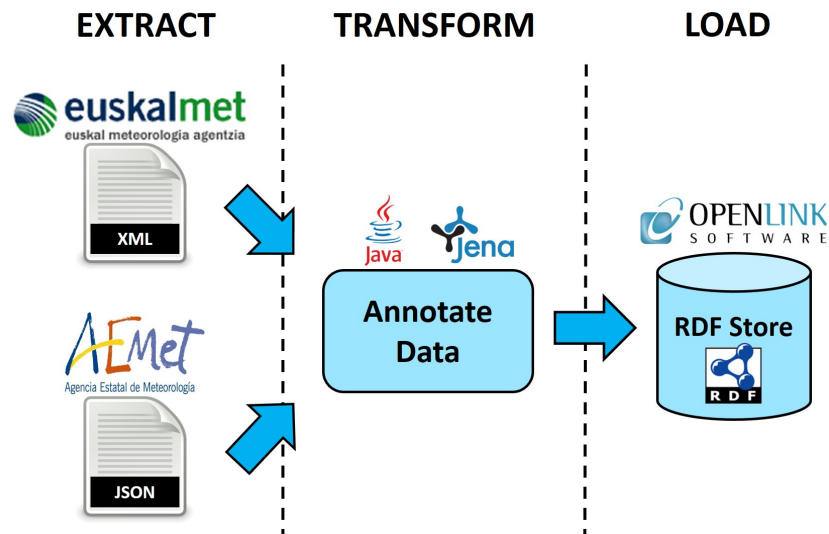


Figure 7.4: Overview of the EEP SA’s ETL process after its customization for AEMET weather stations.

The ETL process proposed by EEP SA is designed in a modular way so, parts of this process could be reused to develop an adaptation for the AEMET weather stations. In fact, the “T” (Transform) and “L” (Load) parts of the ETL process where reused as-they-are, and only the “E” (Extract) part was customized.

Meteorological Open Data repositories are heterogeneous in terms of formats and data structures, so developing a universal extractor (the “E” part of the ETL) is infeasible. Both the AEMET weather stations metadata and observations are delivered in JSON format by the AEMET Open Data portal<sup>26</sup>. Therefore, a Java based service in charge of accessing the corresponding JSON file, parsing it and structuring it in the adequate format is developed. It is worth mentioning that the coordinate system used by AEMET differs from the one used by Euskalmet, so that the latitude and longitude information had to be translated from UTM (Universal Transverse Mercator) to WGS84 (World Geodetic System 1984) coordinate system. Then, this information is sent to the “T” part of the ETL process, in charge of semantically annotating it. Finally, the “L” part loads it in an RDF store, where it will remain accessible for the data analyst. Figure 7.4 depicts the customization of the EEP SA’s ETL process for the KDD Transformation phase. Listing 7.1 shows an AEMET weather station semantically annotated.

```
@prefix : <http://www.tekniker.es/iof2020#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix aemet: <http://aemet.linkeddata.es/ontology/> .
@prefix dbo: <http://dbpedia.org/ontology/> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
```

<sup>26</sup><https://opendata.aemet.es>

```

@prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

:weatherStation_aemet_08103 rdf:type aemet:WeatherStation ;
  dbo:owner <http://es.dbpedia.org/page/AEMET> ;
  dbo:province <http://es.dbpedia.org/page/Huesca> ;
  dc:identifier "9784P" ;
  foaf:name "BIELSA" ;
  geo:lat "0.224463"^^xsd:float ;
  geo:long "42.630198"^^xsd:float .

```

Listing 7.1: RDF representation of an AEMET weather station.

EEPSA's parameterizable GeoSPARQL query (shown in Listing 5.14) for retrieving nearby weather station information and parameterizable SPARQL query (shown in Listing 5.15) for retrieving weather station measurements can be reused as they are.

### 7.2.3 The EEPSA Interpretation phase customization

In the KDD Interpretation phase, the EEPSA proposes the EROSO framework, which enables facility managers querying information derived from temperature predictions, to implement the HVAC control strategy that ensures a certain thermally comfortable situation in a given space. However, farmers responsible of the use case farm could not benefit from such assistance. Instead, they require a notification system that alerts them when the farm is expected to reach an uncomfortable thermal situation in the future. This way, they could anticipate and avoid these situations, with all the risks they involve to the animals' welfare.

The EROSO framework semantically annotates all the temperature predictions generated by a predictive model. Then, a set of predefined ontology-driven rules are executed to classify these predictions according to the thermal comfort criteria they fulfil. As it was mentioned before, birds' comfort requirements are different from humans, and additionally, these requirements may vary depending on their breeding stage and growth pace. Therefore, EROSO should be customized to let farmers know whether the temperature predictions fulfil animals comfort requirements or not. Thanks to the extensibility of the framework, this can be easily achieved by simply adding a SPARQL Construct rule. However, since farmers do not need to query different comfort strategies (unlike in the workplace thermal comfort problem), this classification process could be simplified and it was finally made in Java code. This way, predicted temperatures that are not compliant with animals' thermal comfort needs are classified into moderate, high and severe (represented with classes *exn4pfeepsa:ModerateThermalDiscomfort*, *exn4pfeepsa:HighThermalDiscomfort*, and *exn4pfeepsa:SevereThermalDiscomfort* respectively) depending on their level of disparity with optimal comfort temperatures. Therefore, EROSO's workflow for the use case poultry farm is slightly modified. In this occasion, temperature predictions are semantically annotated with PFEEPSA ontology terms and classified according to the alarming uncomfortable situations they generate. Afterwards, this annotated data is stored in a

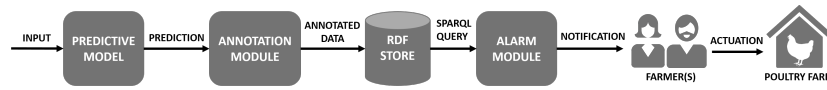


Figure 7.5: Overview of the EROSO framework customization for the poultry farm domain.

### RDF Store.

Furthermore, for this use case EROSO does not need an interface to query different temperature predictions. Instead, EROSO needs to be extended with a notification system. Every time an undesirable comfort situation occurs, a notification should be sent to the farmer in charge<sup>27</sup>. Receiving an alarm notification means that, under the current ventilation and windows actuation mode, poultry comfort requirements may not be fulfilled in the future. This way, the farmer will be able to act to avoid these situations and ensure an adequate environmental condition in each thermal zone.

Figure 7.5 shows an overview of the EROSO framework’s customization for the use case poultry farm.

## 7.3 Experiments

Due to the characteristics of the farm at hand, six different thermal zones can be identified (as shown in Figure 7.6). Furthermore, the right side of the farm (i.e. the right wall of thermal zones 02, 04, and 06) is equipped with three big ventilation systems and the wall adjacent to thermal zones 01 and 02, and thermal zones 05 and 06 contain a set of windows that are opened and closed to change indoor environmental conditions leveraging outdoor weather. Since conditions between thermal zones may differ considerably (e.g. at some periods, differences of over 8°C were registered), temperatures for each zone are predicted independently. That is, six different predictive models need to be developed. Each of those predictive models forecasted hourly temperatures for the upcoming 24 hours within its thermal zone. At the moment of writing this dissertation, predictive models for thermal zones 01, 02 and 04 were developed.

All predictive models were developed with the EEPISA’s assistance<sup>28</sup>, that is, following the steps and guidelines described in Chapter 5. First of all the semantic annotation phase was applied. Unlike in the case of the Open Space where the EEPISA ontology v1.2 was used, all the data regarding the poultry farm, installed sensors and actuators, and their observations and actuations was annotated using the PFEEPISA ontology (the EEPISA ontology’s customization for poultry farms) described in section 7.2.1. This semantic annotation led to the the data selection phase, where the data analyst assistant suggested that the most relevant variables

<sup>27</sup>Notifications could be sent via SMS or email, depending on the farmer’s preferences and the situation’s criticality.

<sup>28</sup>For the sake of simplicity, EEPISA’s application is summarized. A more detailed explanation of EEPISA’s implementation is provided in Chapter 5 and Chapter 6.

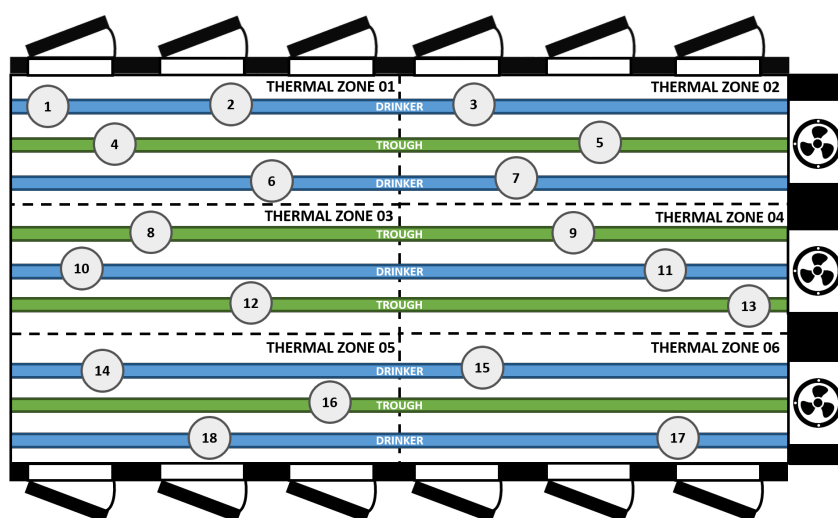


Figure 7.6: Use case poultry farm's thermal zone division.

for the problem at hand were indoor temperature, indoor humidity and stocking density among others. In the transformation phase, AEMET weather stations were exploited thanks to the customized ETL process (described in section 7.2.2) and the set of parameterizable SPARQL queries. This way, outdoor temperature and humidity measurements (which were not measured by any of the devices installed in the farm) were obtained to enlarge the existing data pool.

Predictive models were built using R<sup>29</sup> and R's SVM (Support Vector Machine) algorithm<sup>30</sup>. The decision to change from Rapidminer (which was used in the Open Space use case) to R was founded on the problems Rapidminer had when deploying developed predictive models into production. Furthermore, R has a bigger community and the CRAN package repository<sup>31</sup> with over 13,640 packages at the moment of writing this dissertation.

These predictive models were trained using data gathered through a whole breeding period of 42 days, namely from 6<sup>th</sup> November 2018 to 18<sup>th</sup> December 2018. It is worth clarifying that each predictive model used indoor temperature and indoor humidity observations gathered by sensors within the thermal zone the predictive model is targeting. For example, the predictive model forecasting temperature of thermal zone 02 was trained using temperature and humidity data gathered by the three sensors installed in the thermal zone 02 (i.e. sensors 3, 5 and 7 in Figure 7.6). Furthermore, due to different reasons (e.g. sensor failure), not all sensors were available during this breeding period and, therefore, data registered by some sensors was missing.

<sup>29</sup><https://www.r-project.org>

<sup>30</sup><https://www.rdocumentation.org/packages/e1071/versions/1.7-0/topics/svm>

<sup>31</sup><https://cran.r-project.org/web/packages/>

Table 7.1: MAE and RMSE obtained with predictive models developed for different thermal zones of the use case farm.

Thermal Zone	MAE	RMSE
Zone 01	0.34°C	0.42°C
Zone 02	0.51°C	0.61°C
Zone 04	0.66°C	0.75°C

## 7.4 Evaluation and Results discussion

Predictive models were tested with a 10-fold Cross Validation repeated 5 times. Performance of predictive models was characterized by their MAE and RMSE. Obtained MAEs range between 0.34° and 0.66°C, while RMSEs are between 0.42°C and 0.75°C. Table 7.1 summarize the performance of predictive models for the use case farm’s thermal zones 01, 02 and 04.

Predictive model for thermal zone 01 is the one with the best performance in terms of prediction error. In this thermal zone, indoor temperatures do not suffer from sudden abrupt changes and they are quite stable. Therefore, this is a reasonable result. As for thermal zone 02, the predictive model’s MAE is increased in 50% and RMSE in 45% compared with thermal zone 01. Such a considerable error increase can be caused because thermal zone 02 is closer to the farm’s ventilation system, and when this is activated, indoor temperatures may change faster than in thermal zone 01. As a matter of fact, the temperature changes in this thermal zone may be more frequent and abrupt, hindering predictive model’s prediction accuracy. Finally, thermal zone 04’s predictive model has the worst performance in terms of prediction error among the three predictive models. Its MAE and RMSE are nearly twice as the errors of predictive model of thermal zone 01. These results could be related also with the zone’s indoor temperature stability. As thermal zone 04’s sensors are even closer to farm’s ventilation system, temperatures may suffer from even more abrupt changes than in thermal zone 02, and therefore, this predictive model may be prone to have higher errors.

## 7.5 The PFEEPSA in Production

This section describes the transition of the developed predictive models into production. The architecture deployment of the software components taking part in the analytic process is made via Docker<sup>32</sup>, a platform to develop, deploy, and run applications with containers. Furthermore, these docker containers are deployed in two Ubuntu host machines. The components taking part in the analytic process are shown next:

<sup>32</sup><https://www.docker.com/>



- Apache Tomcat<sup>33</sup> version 8.5.24 as the application server. It executes the scheduled tasks.
- MongoDB<sup>34</sup> version 3.6.0 as a database. It stores the data used as input for the predictive model.
- Rserve<sup>35</sup> version 3.2.5 as the analytic engine. It executes the R-based predictive model.
- Openlink Virtuoso<sup>36</sup> Universal Server version 07.20.3217 as the RDF Store. It stores semantically annotated data.

Apache Tomcat executes a scheduled task with a periodicity of 1 hour. Firstly, this task retrieves the necessary data to be used as input of the predictive model from MongoDB and prepares it adequately. Then, this data is sent to Rserve to be used as input of the predictive model to be executed. Afterwards, the predictive model's output (i.e. the farm's temperature predictions) are compared with poultry's comfort requirement curve (which is updated as poultry growth pace changes), they are semantically annotated and stored in the Virtuoso RDF Store. Finally, this semantically annotated data can be further exploited to send timely notifications to farmers and allow them making the necessary actuations in the farm.

---

<sup>33</sup><http://tomcat.apache.org/>

<sup>34</sup><https://www.mongodb.com/>

<sup>35</sup><https://www.rforge.net/Rserve/>

<sup>36</sup><https://virtuoso.openlinksw.com/>



## Chapter 8

# Conclusions

Achieving occupants thermal comfort with an efficient use of heating and cooling systems from an energetic point of view, is still an unsolved problem in most buildings. This problem is specially recurrent in tertiary buildings which usually have complex rooms and spaces with big dimensions. Although the maturity of the IoT for monitoring the real-world has paved the way to solve such an issue, new difficulties have also arisen, related with the handling of the new heterogeneous data available. The main group of people suffering from such a problematic scenario are data analysts, who have to deal with this data when implementing KDD processes.

This thesis has leveraged Semantic Technologies for assisting data analysts through the whole KDD process, towards the achievement of energy efficiency and thermal comfort in tertiary buildings. To this end, a set of semantic resources have been carefully designed to support frameworks and tools that assist data analysts in different KDD phases, with the final goal of developing predictive models to solve the aforementioned problematic scenarios. This way, domain and expert knowledge can be exploited by data analysts with a lack of domain knowledge, thus avoiding the undesirable trial-and-error approach when developing predictive models.

In this thesis it has been shown that Semantic Technologies enable the support of data analysts in KDD processes. This support is not limited to certain KDD phases and it is extendible to the whole process. Although in this thesis focus has been placed on energy efficiency and thermal comfort in tertiary buildings, Semantic Technologies' capabilities enable the development of data analyst assistants for other domains and problems. The main drivers of these developments are capabilities of the Semantic Technologies to capture expert knowledge into a form which can be leveraged by non-experts and to exploit the data's underlying semantics. Certainly, although this thesis presents a promising approach, the full potential of Semantic Technologies in the support of KDD processes is still to be unlocked. Therefore, Semantic Technologies are worth further research efforts in this regard.

## 8.1 Contributions

This section outlines the major contributions of this thesis towards alleviating data analysts work in the development of predictive models. Figure 8.1 represents a summary of these contributions.

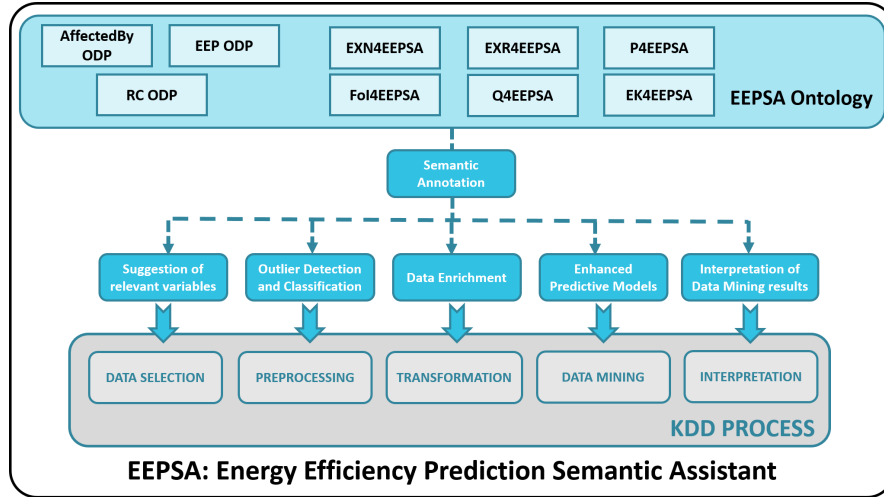


Figure 8.1: Summary of the major contributions of this thesis.

### 8.1.1 The AffectedBy, EEP and RC ODPs

These three ODPs (Ontology Design Patterns) have been proposed to enable the concise representation of the core scope of the problem presented in this thesis. Certainly, these ODPs are the cornerstone of additional semantic developments and support data analysts, as explained in section 4.3.

The AffectedBy ODP supports the discovery of relevant variables affecting qualities of a feature of interest. The EEP (Execution-Executor-Procedure) ODP extends the AffectedBy ODP to support data analysts making further queries to discover sensors or actuators that observe or act on a given quality. And the RC (Result-Context) ODP represents the results of observations and actuations as well as their contexts.

The careful design of these three ODPs' property axioms overcome weaknesses discovered in existing ODP-based ontologies such as the SSN ontology or the SEAS FeatureOfInterest ontology. Furthermore, these ODPs try to be minimal in the number of classes and properties offered but complete with respect to the considered CQs and include appropriate ontology axioms that allow proper inferences.

The scope of this three ODPs is isolated from energy efficiency and thermal comfort problems in tertiary buildings. Instead, they are aimed at supporting

data analysts in problems related with features of interest and their respective qualities, as well as observations and actuations, the sensors and actuators that generate them, and the procedures used. This is a very recurrent scenario nowadays due to the IoT's rapid spread and the abundance of sensing and actuating devices.

The proper documentation of these ODPs, their publication in well-known repositories (e.g. the ODP repository), and their alignment with related ODPs, domain ontologies and upper-level ontologies, promotes their reusability. This enhances their capabilities of being used as basic building blocks in ontologies covering a domain where the sensing and actuating scenarios are present.

### 8.1.2 The EEP SA ontology

This ontology have been proposed as a core ontology that supports a data analyst assistant towards the development of predictive models to solve energy efficiency and thermal comfort problems in tertiary buildings, as it is explained in Chapter 4.

The EEP SA (Energy Efficiency Prediction Semantic Assistant) ontology's backbone has been defined as a combination of the AffectedBy, EEP and RC ODPs, which provide appropriate concepts to represent scenarios where observations or actuations play a key role.

On top of these ODPs, six ontology modules have been developed aimed at comprising the set of suitable terms to support data analysts through energy efficiency and thermal comfort problems in tertiary buildings. Each ontology module specializes the knowledge in the scope of the stub classes defined in the ODPs, reusing existing resources as much as possible. More specifically, these ontology modules are FoI4EEP SA for representing building and building spaces; Q4EEP SA for representing qualities of these spaces; EXR4EEP SA for representing executors such as sensors and actuators; P4EEP SA for representing specific plans or methods; EXN4EEP SA for representing executions such as observations and actuations; and EK4EEP SA for representing different types of spaces and the variables affecting their indoor conditions.

Although the EEP SA ontology is focused on tertiary buildings, it has been designed to support similar use cases in different types of buildings. As a matter of fact, the high encapsulation of the EEP SA ontology modules enables its customization via the module replacement method to address similar scenarios where the development of an accurate predictive model is necessary.

The proposed ontological resources have been well documented and available online. Furthermore, they have been evaluated from three different viewpoints and results show that the EEP SA ontology components are correct from a design standpoint, they are light weighted and they have a high-quality. Furthermore, ODPs and ontology modules are aligned with other related ontologies as well as upper-level ontologies. All these tasks are aimed at fostering the EEP SA ontology's reusability and interoperability.

### 8.1.3 The EEP SA

This process has been proposed as a data analyst assistant for the different KDD phases, as it is explained in Chapter 5. Taking leverage of the EEP SA ontology, the EEP SA provides assistance through the whole KDD process, which to the extent of our knowledge, it was untackled so far. The contributions of the EEP SA can be summarized as follows:

**Suggestion of relevant variables.** In the Data Selection phase (section 5.2), EEP SA assists the data analyst by suggesting the sets of data and variables that will potentially contribute in the development of an accurate predictive model. This approach may be more suitable than the relevance analysis, which may have performance issues in typical large and heterogeneous datasets, and which may not be capable of suggesting new relevant attributes that are not present at the current dataset.

Furthermore, in this approach Semantic Technologies are exploited with objectives that are not limited to data visualization purposes as it happens with most of previous approaches. Namely, the EK4EEP SA ontology module captures expert knowledge regarding the qualities that may influence a given space's temperature, which facilitates the temperature prediction scenario.

**Outlier Detection and Classification.** In the Preprocessing phase the SemOD (Semantic Outlier Detection) framework (section 5.3.1) is proposed, which guides data analysts through the detection of outliers. This outlier detection is a crucial task in data analysis problems counting on data coming from sensors, as WSNs are prone to fail for various reasons. These failures can make sensors measure outliers that do not represent the reality and can lead to the development of inaccurate predictive models. In these situations, even expert data analysts with a deep domain knowledge may have difficulties identifying outliers. EEP SA solves these difficulties with the SemOD framework, which leverages a set of resources that abstract the data analyst from the underlying Semantic Technologies, therefore, neither a deep domain knowledge nor expertise in these technologies is required to use the SemOD framework. Furthermore, this outlier detection approach depends on the context rather than on other nearby sensors, being therefore applicable to isolated nodes. The SemOD framework not only assists in the outlier detection task, but more importantly, it supports the identification of the potential provenance of outliers, which to the extent of our knowledge is a novelty.

Results show that the SemOD framework enables the detection of outliers which could be hardly detected with a statistical outlier detection technique, due to its fairly ordinary value. This is achieved thanks to the exploitation of Semantic Technologies that enable the representation of the observation context. Furthermore, knowing the provenance of outliers has been proven to be a valuable information for decision-making processes related to determining how to prevent those outliers or act on them.

**Dataset enrichment.** In the Transformation phase (section 5.4), EEPsA leverages meteorological Open Data repositories to enrich the sets of data that will be used to develop the predictive models for the problem at hand. More specifically, the exploited meteorological variables are the ones suggested in the Data Selection phase. Since environmental conditions have a considerable influence on a building's indoor conditions, these meteorological repositories are motivating data sources for problems addressed in this thesis.

To do so, EEPsA proposes an ETL process that allows the extraction, semantic annotation and load of weather station and observed measurement information coming from Open sources. Furthermore a set of parameterizable SPARQL queries are proposed, enabling the exploitation of this information even by non-experts in Semantic Technologies. This ETL process is published online and it can be extended by users to retrieve information from different sources and load them in different RDF Stores.

**Enhanced Predictive Models.** The enrichment of the data in KDD phases previous to the Data Mining phase is expected to improve its quality and to enlarge it with additional relevant variables. Certainly, having a richer pool of data, data analysts are expected to have new possibilities of developing predictive models with better performance.

Results show that the new predictive models enabled by EEPsA can accomplish a reduction of temperature prediction errors. This reduction can vary depending on the scenario, the available data and the problem at hand among others. In a best case scenario, these error reduction can exceed the 25%, and in certain situation, even reach a reduction of 40%.

**Interpretation of Data Mining results.** In the Interpretation phase (section 5.6), the EROSO framework is proposed. This framework focuses on the interpretation of regression predictive model results, which to the extent of our knowledge, is an untackled field. The interpretation phase is typically very resource-consuming even for domain experts, and EROSO is aimed at filling this gap by exploiting Semantic Technologies to allow the optimal decision-making regarding the assurance of thermal comfort in workplaces.

The evaluation of the EROSO framework shows that the flexibility to select different thermal comfort criteria, along with the graphic interface, makes the framework useful for managing different spaces, specially those spaces hosting different activities and with changing thermal requirements.

## 8.2 Future work

The contributions presented in this thesis try to raise awareness of the possibilities of Semantic Technologies in the KDD process, and it could lay the foundations for future data analyst assistants that are not limited to energy effi-

ciency and thermal comfort aspects. Although some contributions are made, it also opens up new paths for research.

First of all, in order to keep the proposed EEP SA approach updated, some maintenance tasks are envisioned.

Regarding the EEP SA ontology and their components (i.e. the ODPs and the EEP SA ontology modules), from now on they should be managed with Ontology<sup>1</sup>. This is a system for collaborative ontology development process, and comprises other tools used in this thesis such as OOPS! for ontology correctness or WIDOCO for documentation purposes. Having all these tools centralized is expected to ease the maintenance and evolution of the ontological resources. Furthermore, in order to keep the EEP SA ontology interoperable, it should be aligned with related ontologies that at the moment of writing this dissertation are still under development, such as PRODUCT and OPM proposed by the W3C LBD community group.

The EEP SA data analyst assistant comprises different frameworks which leverage different versions of the EEP SA ontology. In order to homogenize the process, all components should be updated to leverage the same EEP SA ontology. More specifically, they should be based on the latest EEP SA ontology version presented in Chapter 4.

This thesis also conceives the extension of the proposed approach in different aspects.

The proposed SemOD framework shows its usability to detect temperature outliers when sensors are hit by solar radiation. But WSNs and sensors are prone to fail and suffer from outliers caused by many other various reasons. Therefore, the SemOD framework should be extended to let data analysts discover outliers caused by other causes, such as temperature outliers caused by sensors' exposure to rainfall.

As for the missing values, the experiments performed in this thesis showed that Semantic Technologies and their capabilities to represent metadata could contribute proposing new methods for their imputation. Furthermore, Semantic Technologies can also contribute in describing missing value segments, and assisting data analysts proposing suitable imputation techniques for different types of segments. As a matter of fact, this is a line that it is already being researched.

With regards to the poultry farm use case described in Chapter 7, it is still an ongoing work that needs to be finished. Apart from the development of predictive models for the remaining thermal zones (i.e. thermal zones 03, 05 and 06), the development of the alarm notification system is a high priority task. The alarm generation and management, which could be understood as part of the KDD Interpretation phase, constitutes a problem in other domains, so that its foundation in Semantic Technologies should facilitate its reusability.

Towards the facilitation of the EEP SA's usage, interaction with the system

---

<sup>1</sup><http://ontology.linkeddata.es/>



should be improved with the design and implementation of a set of GUIs. Last but not least, the EEPsA should not only be evaluated according to the performance of generated predictive models. Being a data analyst assistant, the EEPsA should also be evaluated measuring the user's satisfaction and guidance's benefits, both to novice and expert data analysts.

Last but not least, EEPsA is designed aimed at supporting data analysts in thermal comfort and energy efficiency problems in tertiary buildings. So far the thermal comfort part of the problem has been emphasized and in the future, the energy efficiency part of the problem should be equally addressed. Moreover, the EEPsA is aimed at being extended to other domains in the future, namely to the manufacturing and tribology domains.

Apart from the identified future work, the contributions presented in this thesis can also open up new paths for research.



# Bibliography

- [1] P. Waide and D. Gerundino, International standards to develop and promote energy efficiency and renewable energy sources, *Prepared for the G8 Plan of Action. IEA information paper.* (2007).
- [2] 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.
- [3] 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.
- [4] B.P. Haynes, The impact of office comfort on productivity, *Journal of Facilities Management* **6**(1) (2008), 37–51.
- [5] A. Hedge and D.E. Gaygen, Indoor environment conditions and computer work in an office, *Hvac&R Research* **16**(2) (2010), 123–138.
- [6] 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.
- [7] 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 9781466595996.
- [8] 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.
- [9] 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.

- [10] 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.
- [11] 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.
- [12] T. Berners-Lee, J. Hendler and O. Lassila, The semantic web, *Scientific american* **284**(5) (2001), 34–43.
- [13] J. Domingue, D. Fensel and J.A. Hendler, Introduction to the Semantic Web Technologies, in: *Handbook of Semantic Web Technologies*, J. Domingue, D. Fensel and J.A. Hendler, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 1–41. ISBN 978-3-540-92913-0. doi:10.1007/978-3-540-92913-0\_1.
- [14] D. Allemang and J. Hendler, *Semantic web for the working ontologist: effective modeling in RDFS and OWL*, Elsevier, 2011. ISBN 978-0-12-385965-5. doi:10.1016/C2010-0-68657-3.
- [15] F. Manola and E. Miller, *RDF Primer*, W3C Recommendation, W3C, 2004, <http://www.w3.org/TR/rdf-primer/>.
- [16] A. Seaborne and G. Carothers, *RDF 1.1 N-Triples*, W3C Recommendation, W3C, 2014, <http://www.w3.org/TR/n-triples/>.
- [17] N. Walsh, *Using Qualified Names (QNames) as Identifiers in XML Content*, Technical Report, W3C, 2004, <https://www.w3.org/2001/tag/doc/qnameids>.
- [18] G. Carothers and E. Prud'hommeaux, *RDF 1.1 Turtle*, W3C Recommendation, W3C, 2014, <http://www.w3.org/TR/turtle/>.
- [19] T. Heath and C. Bizer, *Linked data: Evolving the web into a global data space*, Synthesis lectures on the semantic web: theory and technology, Vol. 1, Morgan & Claypool Publishers, 2011, pp. 1–136. doi:10.2200/S00334ED1V01Y201102WBE001.
- [20] E. Prud'hommeaux and A. Seaborne, *SPARQL Query Language for RDF*, W3C Recommendation, W3C, 2008, <https://www.w3.org/TR/rdf-sparql-query/>.
- [21] D. Brickley and R. Guha, *RDF Schema 1.1*, W3C Recommendation, W3C, 2014, <http://www.w3.org/TR/rdf-schema/>.
- [22] D. McGuinness and F. van Harmelen, *OWL Web Ontology Language Overview*, W3C Recommendation, W3C, 2004, <https://www.w3.org/TR/owl-features/>.
- [23] B. Parsia, P. Patel-Schneider, P. Hitzler, S. Rudolph and M. Krötzsch, *OWL 2 Web Ontology Language Primer (Second Edition)*, W3C Recommendation, W3C, 2012, <https://www.w3.org/TR/owl2-primer/>.

- [24] R. Studer, V.R. Benjamins and D. Fensel, Knowledge engineering: principles and methods, *Data and knowledge engineering* **25**(1) (1998), 161–198, ISSN 0169-023X. doi:10.1016/S0169-023X(97)00056-6.
- [25] N. Guarino, Formal Ontology and Information Systems, in: *Formal ontology in information systems*, Vol. 46, N. Guarino, ed., IOS press, 1998.
- [26] O. Lassila and D. McGuinness, The role of frame-based representation on the semantic web, *Linköping Electronic Articles in Computer and Information Science* **6**(5) (2001), 2001.
- [27] 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.
- [28] 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 1-58113-146-1. doi:10.1145/319950.320008.
- [29] 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 978-3-540-46014-5. doi:10.1007/3-540-46014-4\_26.
- [30] 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.
- [31] 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.
- [32] B. Dave, A. Buda, A. Nurminen and K. Främling, A framework for integrating BIM and IoT through open standards, *Automation in Construction* **95** (2018), 35–45, ISSN 0926-5805. doi:10.1016/j.autcon.2018.07.022.
- [33] 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.
- [34] 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.
- [35] D. Oberle, How ontologies benefit enterprise applications **5**(6) (2014), 473–491. doi:10.3233/SW-130114.

- [36] 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.
- [37] 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).
- [38] 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.
- [39] 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 978-3-319-15326-1. doi:10.1007/978-3-319-15326-1\_2.
- [40] 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.
- [41] D. Bonino and F. Corno, Dogont - Ontology Modeling for Intelligent Domestic Environments, in: *International Semantic Web Conference*, Springer, 2008, pp. 790–803. doi:10.1007/978-3-540-88564-1\_51.
- [42] 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.
- [43] J.J.V. Díaz, M.R. Wilby, A.B.R. González and J.G. Muñoz, 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.
- [44] 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 978-3-319-19656-5. doi:10.1007/978-3-319-19656-5\_1.
- [45] 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.
- [46] 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..

- [47] 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.
- [48] 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.
- [49] 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.
- [50] 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 978-1-4503-4264-3. doi:10.1145/2993422.2993577.
- [51] 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.
- [52] 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.
- [53] 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.
- [54] 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.
- [55] 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.
- [56] 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.

- [57] 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.
- [58] 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).
- [59] 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.
- [60] 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>.
- [61] A. Haller, K. Janowicz, S. Cox, M. Lefrançois, K. Taylor, D.L. Phuoc, J. Lieberman, R. Garcia-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.
- [62] 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.
- [63] 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.
- [64] 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.
- [65] 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.
- [66] 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.



- [67] 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.
- [68] 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.
- [69] M. Alirezaie, K. Hammar and E. Blomqvist, SmartEnv as a Network of Ontology Patterns, *Semantic Web To be published* (2018). <http://www.semantic-web-journal.net/>.
- [70] 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.
- [71] 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/>.
- [72] R. Fikes and Q. Zhou, A Reusable Time Ontology, in: *Proceeding of the AAAI Workshop on Ontologies for the Semantic Web*, 2002.
- [73] 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.
- [74] 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 978-3-642-18472-7. doi:10.1007/978-3-642-18472-7\_8.
- [75] 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.
- [76] 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.
- [77] S. Cox and C. Little, Time Ontology in OWL, W3C Recommendation, W3C, 2017, <https://www.w3.org/TR/2017/REC-owl-time-20171019/>.
- [78] 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.
- [79] M. Perry and J. Herring, OGC GeoSPARQL-A geographic query language for RDF data, *OGC implementation standard* (2012).

- [80] J.M. Keil and S. Schindler, Comparison and evaluation of ontologies for units of measurement, *Semantic Web* (2018), 1–19.
- [81] 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. Nuzolese, S. Rudolph, M. Maleshkova, H. Paulheim, J.Z. Pan and M. Alam, eds, Springer International Publishing, 2018, pp. 196–201. ISBN 978-3-319-98192-.
- [82] P. Panov, S. Deroski and L. Soldatova, OntoDM: An Ontology of Data Mining, in: *2008 IEEE International Conference on Data Mining Workshops*, 2008, pp. 752–760, ISSN 2375-9232. doi:10.1109/ICDMW.2008.62.
- [83] Y. Li, M.A. Thomas and K.-M. Osei-Bryson, Ontology-based data mining model management for self-service knowledge discovery, *Information Systems Frontiers* **19**(4) (2017), 925–943, ISSN 1572-9419. doi:10.1007/s10796-016-9637-y.
- [84] C. Fürber and M. Hepp, Towards a Vocabulary for Data Quality Management in Semantic Web Architectures, in: *Proceedings of the 1st International Workshop on Linked Web Data Management, LWDM '11*, ACM, New York, NY, USA, 2011, pp. 1–8. ISBN 978-1-4503-0608-9. doi:10.1145/1966901.1966903.
- [85] H. Qiu, 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.
- [86] A. Zhou, D. Yu and W. Zhang, A research on intelligent fault diagnosis of wind turbines based on ontology and FMECA, *Advanced Engineering Informatics* **29**(1) (2015), 115–125, ISSN 1474-0346. doi:10.1016/j.aei.2014.10.001.
- [87] B. Steenwinckel, P. Heyvaert, D.D. Paepe, O. Janssens, S.V. Hautte, A. Dimou, F.D. Turck, S.V. Hoecke and F. Ongenaes, Towards Adaptive Anomaly Detection and Root Cause Analysis by Automated Extraction of Knowledge from Risk Analyses, in: *9th International Semantic Sensor Networks Workshop*, 2018.
- [88] A.-S. Dadzie and M. Rowe, Approaches to visualising linked data: A survey, *Semantic Web* **2**(2) (2011), 89–124. doi:10.3233/SW-2011-0037.
- [89] J. Han, M. Kamber and J. Pei, Chapter 3 - Data Preprocessing, in: *Data Mining (Third Edition)*, Third edition edn, J. Han, M. Kamber and J. Pei, eds, The Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann, Boston, 2012, pp. 39–82. ISBN 978-0-12-381479-1. doi:10.1016/B978-0-12-381479-1.00002-2.
- [90] T. Friedman and M. Smith, Measuring the Business Value of Data Quality, Technical Report, Gartner, 2011.

- [91] C. Fürber, *Data quality management with semantic technologies*, Springer, 2015. doi:10.1007/978-3-658-12225-6.
- [92] V. Kotu and B. Deshpande, Chapter 11 - Anomaly Detection, in: *Predictive Analytics and Data Mining*, V. Kotu and B. Deshpande, eds, Morgan Kaufmann, Boston, 2015, pp. 329–345. ISBN 978-0-12-801460-8. doi:10.1016/B978-0-12-801460-8.00011-2.
- [93] V.J. Hodge and J. Austin, A survey of outlier detection methodologies, *Artificial Intelligence Review* **22**(2) (2004), 85–126. doi:10.1023/B:AIRE.0000045502.10941.a9.
- [94] V. Chandola, A. Banerjee and V. Kumar, Anomaly detection: A survey, *ACM computing surveys (CSUR)* **41**(3) (2009), 15. doi:10.1145/1541880.1541882.
- [95] Y. Zhang, N. Meratnia and P. Havinga, Outlier Detection Techniques for Wireless Sensor Networks: A Survey, *IEEE Communications Surveys Tutorials* **12**(2) (2010), 159–170, ISSN 1553-877X. doi:10.1109/SURV.2010.021510.00088.
- [96] D. Wienand and H. Paulheim, Detecting Incorrect Numerical Data in DBpedia, in: *The Semantic Web: Trends and Challenges*, V. Presutti, C. d’Amato, F. Gandon, M. d’Aquin, S. Staab and A. Tordai, eds, Springer International Publishing, Cham, 2014, pp. 504–518. ISBN 978-3-319-07443-6. doi:10.1007/978-3-319-07443-6\_34.
- [97] H. Paulheim, Identifying Wrong Links between Datasets by Multi-dimensional Outlier Detection., in: *Third International Workshop on Debugging Ontologies and Ontology Mappings (WoDOOM 2014)*, 2014, pp. 27–38.
- [98] D. Kontokostas, P. Westphal, S. Auer, S. Hellmann, J. Lehmann, R. Cornelissen and A. Zaveri, Test-driven Evaluation of Linked Data Quality, in: *Proceedings of the 23rd International Conference on World Wide Web, WWW ’14*, ACM, New York, NY, USA, 2014, pp. 747–758. ISBN 978-1-4503-2744-2. doi:10.1145/2566486.2568002.
- [99] D. Kontokostas, A. Zaveri, S. Auer and J. Lehmann, TripleCheckMate: A Tool for Crowdsourcing the Quality Assessment of Linked Data, in: *Knowledge Engineering and the Semantic Web*, P. Klinov and D. Mourontsev, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 265–272. ISBN 978-3-642-41360-5. doi:10.1007/978-3-642-41360-5\_22.
- [100] A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann and S. Auer, Quality assessment for Linked Data: A Survey, *Semantic Web* **7**(1) (2015), 63–93. doi:10.3233/SW-150175.
- [101] C. Fürber and M. Hepp, Using semantic web resources for data quality management, in: *International Conference on Knowledge Engineering and Knowledge Management*, Springer, 2010, pp. 211–225. doi:10.1007/978-3-642-16438-5\_15.

- [102] C. Fürber and M. Hepp, Using SPARQL and SPIN for data quality management on the semantic web, in: *International Conference on Business Information Systems*, Vol. 47, Springer, 2010, pp. 35–46. doi:10.1007/978-3-642-12814-1\_4.
- [103] N. Khasawneh and C. Chan, Active User-Based and Ontology-Based Web Log Data Preprocessing for Web Usage Mining, in: *2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06)*, 2006, pp. 325–328. doi:10.1109/WI.2006.32.
- [104] X. Wang, H.J. Hamilton and Y. Bither, An ontology-based approach to data cleaning, Technical Report, University of Regina ,Department of Computer Science, 2005.
- [105] D. Perez-Rey, A. Anguita and J. Crespo, OntoDataClean: Ontology-Based Integration and Preprocessing of Distributed Data, in: *Biological and Medical Data Analysis*, N. Maglaveras, I. Chouvarda, V. Koutkias and R. Brause, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 262–272. ISBN 978-3-540-68065-9. doi:10.1007/11946465\_24.
- [106] Y. Wang and S. Yang, Outlier detection from massive short documents using domain ontology, in: *2010 IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS)*, Vol. 3, IEEE, 2010, pp. 558–562. doi:10.1109/ICICISYS.2010.5658426.
- [107] S. Brüggemann and F. Grüning, Using Ontologies Providing Domain Knowledge for Data Quality Management, in: *Networked Knowledge - Networked Media: Integrating Knowledge Management, New Media Technologies and Semantic Systems*, T. Pellegrini, S. Auer, K. Tochtermann and S. Schaffert, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 187–203. ISBN 978-3-642-02184-8. doi:10.1007/978-3-642-02184-8\_13.
- [108] A. Preece, P. Missier, S. Embury, B. Jin and M. Greenwood, An Ontology-based Approach to Handling Information Quality in e-Science, *Concurrency and Computation Practice and Experience* **20**(3) (2008), 253–264, ISSN 1532-0626. doi:10.1002/cpe.v20:3.
- [109] L. Gao, M. Bruenig and J. Hunter, Semantic-based detection of segment outliers and unusual events for wireless sensor networks, *International Conference on Information Quality* (2014), 127–144.
- [110] J. Luengo, S. García and F. Herrera, On the choice of the best imputation methods for missing values considering three groups of classification methods, *Knowledge and Information Systems* **32**(1) (2012), 77–108, ISSN 0219-3116. doi:10.1007/s10115-011-0424-2.
- [111] M. Luo, F. Wang and X. Hu, Estimating Missing Values of WSN using modified Frequent itemsets mining and NN search, in: *Proceedings of the 4th International Conference on Computer, Mechatronics, Control and Electronic Engineering*, 2015, pp. 673–678.
- [112] R. Kumar, D. Chaurasia, N. Chuahan and N. Chand, Predicting Missing Values in Wireless Sensor Network using Spatial-Temporal Correlation **7**(3) (2014), 20–25, ISSN 2250-3501.

- [113] A. Farhangfar, L.A. Kurgan and W. Pedrycz, A novel framework for imputation of missing values in databases, *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* **37**(5) (2007), 692–709.
- [114] A.C. Acock, Working with missing values, *Journal of Marriage and family* **67**(4) (2005), 1012–1028. doi:10.1111/j.1741-3737.2005.00191.x.
- [115] E. Acuña and C. Rodriguez, The Treatment of Missing Values and its Effect on Classifier Accuracy, in: *Classification, Clustering, and Data Mining Applications*, D. Banks, F.R. McMorris, P. Arabie and W. Gaul, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 639–647. ISBN 978-3-642-17103-1.
- [116] F.M. Shrive, H. Stuart, H. Quan and W.A. Ghali, Dealing with missing data in a multi-question depression scale: a comparison of imputation methods, *BMC Medical Research Methodology* **6**(1) (2006), 57, ISSN 1471-2288. doi:10.1186/1471-2288-6-57.
- [117] S. Egami, T. Kawamura and A. Ohsuga, Estimation of Spatial Missing Data for Expanding Urban LOD, in: *JIST (Workshops & Posters)*, 2016, pp. 82–85.
- [118] O. Lehmborg, D. Ritze, P. Ristoski, R. Meusel, H. Paulheim and C. Bizer, The Mannheim Search Join Engine, *Web Semantics: Science, Services and Agents on the World Wide Web* **35** (2015), 159–166, Semantic Web Challenge 2014, ISSN 1570-8268. doi:10.1016/j.websem.2015.05.001.
- [119] I. Guyon and A. Elisseeff, An Introduction to Feature Extraction, in: *Feature Extraction: Foundations and Applications*, I. Guyon, M. Nikravesh, S. Gunn and L.A. Zadeh, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 1–25. ISBN 978-3-540-35488-8. doi:10.1007/978-3-540-35488-8\_1.
- [120] V. Narasimha, P. Kappara, R. Ichise and O. Vyas, LiDDM: A Data Mining System for Linked Data, *Workshop on Linked Data on the Web. CEUR Workshop Proceedings* **813** (2011).
- [121] H. Paulheim and J. Fümkrantz, Unsupervised generation of data mining features from linked open data, *Proceedings of the 2nd international conference on web intelligence, mining and semantics* (2012), 31. doi:10.1145/2254129.2254168.
- [122] P. Ristoski, C. Bizer and H. Paulheim, Mining the web of linked data with rapidminer, *Web Semantics: Science, Services and Agents on the World Wide Web* **35** (2015), 142–151. doi:10.1016/j.websem.2015.06.004.
- [123] W. Cheng, G. Kasneci, T. Graepel, D. Stern and R. Herbrich, Automated feature generation from structured knowledge, in: *Proceedings of the 20th ACM international conference on Information and knowledge management*, ACM, 2011, pp. 1395–1404. doi:10.1145/2063576.2063779.
- [124] U. Sivarajah, M.M. Kamal, Z. Irani and V. Weerakkody, Critical analysis of Big Data challenges and analytical methods, *Journal of Business Research* **70** (2017), 263–286, ISSN 0148-2963. doi:10.1016/j.jbusres.2016.08.001.

- [125] W. Derguech, E. Bruke and E. Curry, An Autonomic Approach to Real-Time Predictive Analytics using Open Data and Internet of Things, *2014 IEEE 11th Intl Conf on Ubiquitous Intelligence and Computing and 2014 IEEE 11th Intl Conf on Autonomic and Trusted Computing and 2014 IEEE 14th Intl Conf on Scalable Computing and Communications and Its Associated Workshops* (2014), 204–211. doi:10.1109/UIC-ATC-ScalCom.2014.13.
- [126] P. Ristoski and H. Paulheim, Analyzing statistics with background knowledge from linked open data, in: *Workshop on Semantic Statistics*, 2013.
- [127] I. Tiddi, Explaining Data Patterns using Knowledge from the Web of Data, PhD dissertation, The Open University, 2016.
- [128] A. Vavpetič, V. Podpečan and N. Lavrač, Semantic subgroup explanations, *Journal of Intelligent Information Systems* **42**(2) (2014), 233–254, ISSN 1573-7675. doi:10.1007/s10844-013-0292-1.
- [129] I. Tiddi, M. d’Aquin and E. Motta, Dedalo: Looking for Clusters Explanations in a Labyrinth of Linked Data, in: *The Semantic Web: Trends and Challenges*, V. Presutti, C. d’Amato, F. Gandon, M. d’Aquin, S. Staab and A. Tordai, eds, Springer International Publishing, Cham, 2014, pp. 333–348. ISBN 978-3-319-07443-6. doi:10.1007/978-3-319-07443-6\_23.
- [130] M. d’Aquin and N. Jay, Interpreting data mining results with linked data for learning analytics: motivation, case study and directions, in: *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, ACM, 2013, pp. 155–164. doi:10.1145/2460296.2460327.
- [131] V. Svátek, J. Rauch and M. Ralbovský, Ontology-Enhanced Association Mining, in: *Semantics, Web and Mining*, M. Ackermann, B. Berendt, M. Grobelnik, A. Hotho, D. Mladenič, G. Semeraro, M. Spiliopoulou, G. Stumme, V. Svátek and M. van Someren, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 163–179. ISBN 978-3-540-47698-6. doi:10.1007/11908678\_11.
- [132] D. Dou, H. Wang and H. Liu, Semantic data mining: A survey of ontology-based approaches, in: *Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015)*, 2015, pp. 244–251. doi:10.1109/ICOSC.2015.7050814.
- [133] 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 978-3-540-24750-0. doi:10.1007/978-3-540-24750-0\_6.
- [134] 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.*
- [135] 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 978-3-642-24794-1. doi:10.1007/978-3-642-24794-1\_2.
- [136] 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 978-3-642-24794-1. doi:10.1007/978-3-642-24794-1\_5.
- [137] T. Schandl and A. Blumauer, PoolParty: SKOS thesaurus management utilizing linked data, in: *Extended Semantic Web Conference*, Springer, 2010, pp. 421–425.
- [138] M.A. Musen, The protégé project: a look back and a look forward, *AI matters* **1**(4) (2015), 4–12.
- [139] S. Chacon and B. Straub, *Pro git*, Apress, 2014.
- [140] E. Simperl, Reusing ontologies on the Semantic Web: A feasibility study, *Data & Knowledge Engineering* **68**(10) (2009), 905–925.
- [141] 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/>.
- [142] M. Fernández-López, M.C. Suárez-Figueroa and A. Gómez-Pérez, Ontology Development by Reuse, 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. 147–170. ISBN 978-3-642-24794-1. doi:10.1007/978-3-642-24794-1\_7.
- [143] P.-Y. Vandenbussche, G.A. Atemezing, M. Poveda-Villalón and B. Vatant, Linked Open Vocabularies (LOV): a gateway to reusable semantic vocabularies on the Web, *Semantic Web* **8**(3) (2017), 437–452. doi:10.3233/SW-160213.
- [144] A. Gyrard, A. Zimmermann and A. Sheth, Building IoT based applications for Smart Cities: How can ontology catalogs help?, *IEEE Internet of Things Journal* (2018).
- [145] A. Lozano-Tello and A. Gómez-Pérez, Ontometric: A method to choose the appropriate ontology, *Journal of Database Management (JDM)* **15**(2) (2004), 1–18.
- [146] 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 978-3-540-92673-3. doi:10.1007/978-3-540-92673-3\_10.
- [147] P. Hitzler, A. Gangemi and K. Janowicz, *Ontology Engineering with Ontology Design Patterns: Foundations and Applications*, Vol. 25, IOS Press, 2016.
- [148] N.F. Noy, Semantic integration: a survey of ontology-based approaches, *ACM Sigmod Record* **33**(4) (2004), 65–70. doi:10.1145/1041410.1041421.

- [149] 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.
- [150] 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>.
- [151] 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.
- [152] 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.
- [153] 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.
- [154] 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 978-3-642-01907-4. doi:10.1007/978-3-642-01907-4.4.
- [155] L.D. Collins and R.H. Middleton, Distributed demand peak reduction with non-cooperative players and minimal communication, *IEEE Transactions on Smart Grid* (2018), ISSN 1949-3053. doi:10.1109/TSG.2017.2734113.
- [156] P. Warren, A review of demand-side management policy in the UK, *Renewable and Sustainable Energy Reviews* **29** (2014), 941–951, ISSN 1364-0321. doi:10.1016/j.rser.2013.09.009.
- [157] 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.
- [158] C. Tejo-Alonso, D. Berrueta, L. Polo and S. Fernández, Metadata for Web Ontologies and Rules: Current Practices and Perspectives, in: *Metadata and Semantic Research*, E. García-Barriocanal, Z. Cebeci, M.C. Okur and A. Öztürk, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 56–67. ISBN 978-3-642-24731-6.
- [159] S. Peroni, D. Shotton and F. Vitali, The Live OWL Documentation Environment: A Tool for the Automatic Generation of Ontology Documentation, in: *Knowledge Engineering and Knowledge Management*, A. ten Teije, J. Völker, S. Handschuh, H. Stuckenschmidt, M. d’Acquin, A. Nikolov,



- N. Aussenac-Gilles and N. Hernandez, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 398–412. ISBN 978-3-642-33876-2.
- [160] 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 978-3-319-68204-4.
- [161] P.-Y. Vandenbussche and Bernard Vatant, Metadata Recommendations For Linked Open Data Vocabularies (2011).
- [162] D. Garijo and M. Poveda-Villalón, A checklist for complete vocabulary metadata, Technical Report, 2017. <https://w3id.org/widoco/bestPractices>.
- [163] J. Brank, M. Grobelnik and D. Mladeníć, A Survey of Ontology Evaluation Techniques, in: *Proc. of 8th Int. multi-conf. Information Society*, 2005, pp. 166–169.
- [164] 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 978-0-387-48438-9. doi:10.1007/978-0-387-48438-9\_8.
- [165] 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 978-3-540-72667-8.
- [166] M. Poveda-Villalón, A. Gómez-Pérez and M.C. Suárez-Figueroa, OOPS! (Ontology Pitfall Scanner!): An On-line Tool for Ontology Evaluation, *Int. J. Semant. Web Inf. Syst.* **10**(2) (2014), 7–34, ISSN 1552-6283. doi:10.4018/ijswis.2014040102.
- [167] 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 978-3-319-49004-5.
- [168] S. Pokraev, D. Quartel, M.W.A. Steen and M. Reichert, Semantic Service Modeling: Enabling System Interoperability, in: *Enterprise Interoperability*, G. Doumeingts, J. Müller, G. Morel and B. Vallespir, eds, Springer London, London, 2007, pp. 221–230. ISBN 978-1-84628-714-5. doi:10.1007/978-1-84628-714-5\_21.
- [169] L. Obrst, Ontologies for semantically interoperable systems, in: *Proceedings of the twelfth international conference on Information and knowledge management*, ACM, 2003, pp. 366–369. doi:10.1145/956863.956932.

- [170] L. Yu and H. Liu, Efficient feature selection via analysis of relevance and redundancy, *Journal of machine learning research* **5**(Oct) (2004), 1205–1224.
- [171] I. Esnaola-Gonzalez, J. Bermúdez, I. Fernandez, S. Fernandez and A. Arnaiz, Towards a Semantic Outlier Detection Framework in Wireless Sensor Networks, in: *Proceedings of the 13th International Conference on Semantic Systems*, Semantics2017, ACM, New York, NY, USA, 2017, pp. 152–159. ISBN 978-1-4503-5296-3. doi:10.1145/3132218.3132226.
- [172] S. Moritz, A. Sardá, T. Bartz-Beielstein, M. Zaefferer and J. Stork, Comparison of different methods for univariate time series imputation in R, *arXiv preprint arXiv:1510.03924* (2015).
- [173] I. Esnaola-Gonzalez, J. Bermúdez, I. Fernández and A. Arnaiz, EROSO: Semantic Technologies Towards Thermal Comfort in Workplaces, in: *Proceedings of the 21th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2018)*, C.F. Zucker, C. Ghidini, A. Napoli and Y. Toussaint, eds, Springer International Publishing, 2018, pp. 519–533. doi:10.1007/978-3-030-03667-6\_33.
- [174] I. Esnaola-Gonzalez, J. Bermúdez, I. Fernandez and A. Arnaiz, Supporting Predictive Models Results Interpretation for Comfortable Workplaces, in: *Proceedings of the ISWC 2018 Posters & Demonstrations, Industry and Blue Sky Ideas Tracks co-located with 17th International Semantic Web Conference (ISWC 2018)*, Vol. 2180, M. van Erp, M. Atre, V. Lopez, K. Srinivas and C. Fortuna, eds, CEUR, 2018.
- [175] D. Calvanese, B. Cogrel, S. Komla-Ebri, R. Kontchakov, D. Lanti, M. Rezk, M. Rodriguez-Muro and G. Xiao, Ontop: answering SPARQL queries over relational databases, *Semantic Web* **8**(3) (2016), 471–487. doi:10.3233/SW-160217.
- [176] R.E. Kalman, A new approach to linear filtering and prediction problems, *Journal of basic Engineering* **82**(1) (1960), 35–45.
- [177] U. Garciarena, An investigation of imputation methods for discrete databases and multi-variate time series, Masters thesis, University of the Basque Country, 2016. <http://hdl.handle.net/10810/19052>.
- [178] U. Garciarena and R. Santana, An extensive analysis of the interaction between missing data types, imputation methods, and supervised classifiers, *Expert Systems with Applications* **89** (2017), 52–65.
- [179] P. Tormene, T. Giorgino, S. Quaglini and M. Stefanelli, Matching incomplete time series with dynamic time warping: an algorithm and an application to post-stroke rehabilitation, *Artificial intelligence in medicine* **45**(1) (2009), 11–34.
- [180] G. Wu, E.Y. Wu and N. Panda, Formulating Context-dependent Similarity Functions, in: *Proceedings of the 13th Annual ACM International Conference on Multimedia*, MULTIMEDIA '05, ACM, New York, NY, USA, 2005, pp. 725–734. ISBN 1-59593-044-2. doi:10.1145/1101149.1101307.

- [181] R.J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*, OTexts, 2014.
- [182] F. Zamora-Martínez, P. Romeu, P. Botella-Rocamora and J. Pardo, Towards energy efficiency: Forecasting indoor temperature via multivariate analysis, *Energies* **6**(9) (2013), 4639–4659. doi:10.3390/en6094639.
- [183] J. Brooke et al., SUS-A quick and dirty usability scale, *Usability evaluation in industry* **189**(194) (1996), 4–7.
- [184] J. Hulzebosch, Effective heating systems for poultry houses, *World Poultry* **22**(2) (2005), 212–216.
- [185] 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.
- [186] 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.
- [187] 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.
- [188] 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.
- [189] T. Pizzuti, G. Mirabelli, M.A. Sanz-Bobi and F. Gómez-González, Food Track & Trace ontology for helping the food traceability control, *Journal of Food Engineering* **120** (2014), 17–30.



# Appendix A

## List of Abbreviations

This appendix provides a list of abbreviations used throughout this dissertation.

- AEC = Architecture, Engineering, and Construction.
- AHU = Air Handling Unit.
- BIM = Building Information Model.
- BMS = Building Management System.
- BOT = Building Topology Ontology.
- DOI = Digital Object Identifier.
- ETL = Extract, Load, Transform.
- DR = Demand Response.
- DTW = Dynamic Time Warping.
- EEPSA = Energy Efficiency Prediction Semantic Assistant.
- EK4EEPSA = Expert Knowledge for EEPSA.
- EXN4EEPSA = Execution for EEPSA.
- EXR4EEPSA = Executor for EEPSA.
- FM = Facilities Management.
- FMEA = Failure Mode and Effect Analysis.
- FoI4EEPSA = Feature of Interest for EEPSA.
- FTA = Fault Tree Analysis.
- gbXML = Green Building XML.

- GIS = Geographic Information Systems.
- GUI = Graphic User Interface.
- HVAC= Heating, Ventilation and Air Conditioning.
- IDE = Intelligent Domotic Environment.
- IFC = Industry Foundation Classes.
- IoT = Internet of Things.
- IRI = Internationalized Resource Identifier.
- INSHT = Instituto Nacional de Seguridad e Higiene en el Trabajo (Spanish Work Security and Hygiene Institute).
- KDD = Knowledge Discovery in Databases.
- LD = Linked Data.
- LOD = Linked Open Data.
- MAE = Mean Absolute Error.
- MCAR = Missing Completely At Random.
- MEP = Mechanical, Electrical and Plumbing.
- ODP = Ontology Design Pattern.
- ORSD = Ontology Requirements Specification Document.
- OSCS = Open Space Comfort Solution.
- OWL = Web Ontology Language.
- P4EEPSA = Procedure for EEPSA.
- Q4EEPSA = Quality for EEPSA.
- RDF = Resource Description Framework.
- RDFS = RDF Schema.
- RIF = Rule Interchange Format.
- RITE = Reglamento de Instalaciones Térmicas de los Edificios (Spanish Buildings' Thermal Installation Regulation).
- RMSE = Root Mean Squared Error.
- SPARQL = SPARQL Protocol and RDF Query Language.
- SPIN = SPARQL Inference Notation.
- SWRL = Semantic Web Rule Language.
- URI = Uniform Resource Identifier.

- URN = Uniform Resource Name.
- W3C = World Wide Web Consortium.
- XSLT = Extensible Stylesheet Language Transformation.
- WSN = Wireless Sensor Network.





## Appendix B

# Ontology Requirements Specification Document

This appendix shows the Ontology Requirements Specification Document of the EEPISA ontology.

<b>EEPSA Ontology Requirements Specification Document</b>	
<b>1 Purpose</b>	The purpose of this ontology is to provide a knowledge model for energy efficiency and thermal comfort problems in tertiary buildings domain.
<b>2 Scope</b>	The ontology has to focus on the indoor temperature prediction in tertiary buildings. It must not be restricted just to office buildings, being a valid model for tertiary buildings with other activities like the ones with entertainment purposes or stores. The level of granularity is directly related to the competency questions and terms identified.
<b>3 Implementation Language</b>	The ontology has to be implemented in OWL language.
<b>4 Intended End-Users</b>	User 1. Data analyst, not necessarily experts in the domain of application. User 2. Facility Manager. User 3. Ontology engineer.
<b>5 Intended Uses</b>	Use 1. Enrich data/Add semantic to data. Offer resources to which data can be linked. Use 2. Outlier detection. Infer whether a data object is an outlier or not. Use 3. New knowledge/data/relation inference. Inference of implicit knowledge and relationships between data. Use 4. Data addition support. Help data analysts deciding which data can be added to the KDD process.
<b>6 Ontology Requirements</b>	
<b>a. Non-Functional Requirements</b>	NFR1. The Ontology must be written following the CamelCase naming convention.
<b>b. Functional Requirements</b>	See Competency Questions section.
<b>7 Pre-Glossary of Terms</b>	
<b>Terms from Competency Questions</b>	<ul style="list-style-type: none"> <li>• Actuate</li> <li>• Actuating Procedure</li> <li>• Actuation</li> <li>• Actuator</li> <li>• Affect</li> <li>• Air Quality Sensor</li> <li>• Ammonia</li> <li>• Anemometer</li> <li>• Atmospheric Pressure</li> <li>• Belong</li> <li>• Blind actuator</li> <li>• Building</li> <li>• Calorimeter</li> <li>• Cloud Coverage</li> <li>• CO2</li> <li>• Door</li> <li>• Door Actuator</li> <li>• Electric Consumption</li> <li>• Environment Sensors</li> <li>• Feature of Interest</li> <li>• ...</li> </ul>

Objects
<p>Actuator</p> <ul style="list-style-type: none"> <li>• Blind Actuator</li> <li>• Door Actuator</li> <li>• HVAC</li> <li>• Smart Plug</li> <li>• Light Actuator</li> <li>• Smart Plug</li> <li>• Thermostat</li> <li>• Window Actuator</li> </ul> <p>Environment Sensor</p> <ul style="list-style-type: none"> <li>• Air Quality Sensor <ul style="list-style-type: none"> <li>○ Ammonia Sensor</li> <li>○ CO2 Sensor</li> </ul> </li> <li>• Anemometer</li> <li>• Atmospheric Pressure Sensor</li> <li>• Cloud Coverage Sensor</li> <li>• Humidity Sensor</li> <li>• Light Sensor</li> <li>• Movement Sensor</li> <li>• Precipitation Sensor</li> <li>• Solar Radiation Sensor</li> <li>• Sun Sensor</li> <li>• Temperature Sensor</li> </ul> <p>Execution</p> <ul style="list-style-type: none"> <li>• Actuation</li> <li>• Missing Value</li> <li>• Observation <ul style="list-style-type: none"> <li>○ Forecast</li> <li>○ Imputation</li> <li>○ Outlier</li> </ul> </li> </ul> <p>Executor</p> <ul style="list-style-type: none"> <li>• Actuator</li> <li>• Imputation Model</li> <li>• Predictive Model</li> <li>• Sensor</li> </ul> <p>Feature of Interest</p> <ul style="list-style-type: none"> <li>• Door <ul style="list-style-type: none"> <li>○ External Door</li> </ul> </li> <li>• External Building Element <ul style="list-style-type: none"> <li>○ External Door</li> <li>○ External Wall</li> <li>○ External Window</li> <li>○ Skylight</li> </ul> </li> <li>• Roof</li> <li>• Wall <ul style="list-style-type: none"> <li>○ External Wall</li> </ul> </li> <li>• Window <ul style="list-style-type: none"> <li>○ External Window</li> </ul> </li> </ul> <p>Meteorological Quality</p>

<ul style="list-style-type: none"> <li>• Atmospheric Pressure</li> <li>• Cloud Coverage</li> <li>• Outdoor Humidity</li> <li>• Outdoor Temperature</li> <li>• Precipitation Level</li> <li>• Solar Radiation</li> <li>• Sun Position Direction</li> <li>• Sun Position Elevation</li> <li>• Wind Chill</li> <li>• Wind Direction</li> <li>• Wind Speed</li> </ul> <p>Observation</p> <ul style="list-style-type: none"> <li>• Forecast</li> <li>• Imputation</li> <li>• Outlier <ul style="list-style-type: none"> <li>○ Outlier Caused by Device Error</li> <li>○ Outlier Caused by Sensor Location</li> </ul> </li> </ul> <p>Outlier Caused by Device Error</p> <ul style="list-style-type: none"> <li>• Outlier Caused by Power Supply</li> <li>• Outlier Caused by Sensor Malfunction</li> </ul> <p>Outlier Caused by Sensor Location</p> <ul style="list-style-type: none"> <li>• Illuminance Outlier Caused by Light Beam</li> <li>• Temperature Outlier Caused by Rain</li> <li>• Temperature Outlier Caused by Solar Radiation</li> </ul> <p>Procedure</p> <ul style="list-style-type: none"> <li>• Actuating Procedure</li> <li>• Imputation Procedure</li> <li>• Predictive Procedure</li> <li>• Sensing Procedure</li> </ul> <p>Resource Consumption/Generation Quality</p> <ul style="list-style-type: none"> <li>• Electric Consumption</li> <li>• Electric Generation</li> <li>• Gas Consumption</li> <li>• Heat Consumption</li> <li>• Water Consumption</li> </ul> <p>Quality</p> <ul style="list-style-type: none"> <li>• Actuatable Quality <ul style="list-style-type: none"> <li>○ Thermal Comfort Quality</li> </ul> </li> <li>• Observable Quality <ul style="list-style-type: none"> <li>○ Meteorological Quality</li> <li>○ Occupancy</li> <li>○ Orientation</li> <li>○ Resource Consumption/Generation Quality</li> <li>○ Thermal Comfort Quality</li> </ul> </li> </ul> <p>Sensor</p> <ul style="list-style-type: none"> <li>• Environment Sensor</li> <li>• Utility Meter</li> </ul> <p>Thermal Comfort Quality</p> <ul style="list-style-type: none"> <li>• Ammonia</li> <li>• CO2</li> </ul>
---

	<ul style="list-style-type: none"><li>• Illuminance</li><li>• Indoor Humidity</li><li>• Indoor Temperature</li></ul>
	Utility Meter
	<ul style="list-style-type: none"><li>• Calorimeter</li><li>• Electricity Meter</li><li>• Gas Meter</li><li>• Generation Meter</li><li>• Smart Plug</li><li>• Water Meter</li></ul>

## B.1 EEP SA ontology Requirements

In this section, the requirements for the EEP SA ontology are described in the form of Competency Questions (CQs).

CQs tackled by ODPs are summarized in:

- Table B.1 for AffectedBy ODP,
- Table B.2 for EEP ODP, and
- Table B.3 for RC ODP<sup>1</sup>.

Likewise, CQs tackled by ontology modules are summarized in:

- Table B.4 for FoI4EEP SA ontology module,
- Table B.5 for Q4EEP SA ontology module,
- Table B.6 for P4EEP SA ontology module,
- Table B.7 for EXR4EEP SA ontology module,
- Table B.8 for EXN4EEP SA ontology module, and
- Table B.9 for EK4EEP SA ontology module

Table B.1: Requirements addressed by the AffectedBy ODP.

ID	CQ	Answer
CQ01	What are the qualities that influence a feature of interest?	Room03Temperature, Room03Occupancy
CQ02	What are the qualities that affect a given quality of a feature of interest?	Room03Occupancy
CQ03	Which feature of interest does a given quality belong to?	Room03

<sup>1</sup>CQ15 is addressed by combining EEP ODP with RC ODP.

Table B.2: Requirements addressed by the EEP ODP.

ID	CQ	Answer
CQ04	What are the observations/actuactions performed by a given procedure?	Observation01, Actuation35
CQ05	What are the observations/actuactions performed by a given sensor/actuator?	Observation12, Actuation03
CQ06	What are the procedures implemented by a given sensor/actuator?	Procedure01
CQ07	What are the features of interest on a given observation/actuation?	Room03
CQ08	What are the qualities sensed/actuated by a given observations/actuactions?	Room03Humidity
CQ09	What are the features of interest of a given sensor/actuator?	Room03
CQ10	What are the qualities sensed/actuated by a given sensor/actuator?	Room03Humidity

Table B.3: Requirements addressed by the RC ODP.

ID	CQ	Answer
CQ11	Which is the value of an observation/actuation?	1.5kWh
CQ12	When was an observation/actuation generated?	2019-01-03
CQ13	For what time interval or instant is valid an observation/actuation?	TimeInterval61
CQ14	For what spatial location is valid an observation/actuation?	Room03Location
CQ15	Which is the temperature value of room 03 on 2018-11-20 at 16:00?	16°C

Table B.4: Requirements addressed by the FoI4EEPSA ontology module.

ID	CQ	Answer
CQ16	Which building does a given space belong to?	Building16
CQ17	How many spaces does a building have?	53
CQ18	In which storey is a given space located?	Storey02
CQ19	Is a given storey located in an underground storey?	Yes
CQ20	Which space does a given door belong to?	Space12
CQ21	Is a given door adjacent to outdoors?	No
CQ22	How many windows does a given space have?	2
CQ23	Is a given window adjacent to outdoors?	Yes
CQ24	Which building does a given wall belong to?	Building05
CQ25	Is a given wall adjacent to outdoors?	Yes
CQ26	Does a given space have a skylight?	Yes
CQ27	Which is the intended use of the building?	Commercial
CQ28	Which is the expected occupancy type for the building?	Occupancy B (Educational)
CQ29	When was the building built?	1979
CQ30	Which is the gross floor area of the building?	50,000 m <sup>2</sup>

Table B.5: Requirements addressed by the Q4EEPSA ontology module.

ID	CQ	Answer
CQ31	Which are the actuatable qualities?	Thermal Comfort Qualities
CQ32	Which are the observable qualities?	Thermal Comfort Qualities, Meteorological Qualities,...
CQ33	Which are the thermal comfort qualities?	Indoor Humidity, Indoor Temperature,...
CQ34	Which are the meteorological qualities?	Solar Radiation, Cloud Cover,...
CQ35	Which are the resource consumption qualities?	Electric Consumption, Gas Consumption...



Table B.6: Requirements addressed by the P4EEPSA ontology module.

ID	CQ	Answer
CQ36	What are the actuating procedures?	Procedures to act on events
CQ37	What are the predictive procedures?	Procedures to predict events
CQ38	What are the imputation procedures?	Procedures to impute events
CQ39	What are the sensing procedures	Procedures to sense events

Table B.7: Requirements addressed by the EXR4EEPSA ontology module.

ID	CQ	Answer
CQ40	Which type of sensor is a given sensor?	Anenometer
CQ41	Is a given executor a temperature sensor?	Yes
CQ42	Which are the sensors?	Environment sensor, Utility meter
CQ43	Which are the environment sensors?	Precipitation sensor, Humidity sensor,...
CQ44	Which are the utility meters?	Electricity meter, Water meter,...
CQ45	Which type of actuator is a given actuator?	Blind actuator
CQ46	Is a given executor a window actuator?	Yes
CQ47	Which are the actuators?	Door actuator, light actuator,...
CQ48	Is a given executor a predictive model?	No
CQ49	Is a given executor an imputation method?	Yes

Table B.8: Requirements addressed by the EXN4EEPSA ontology module.

ID	CQ	Answer
CQ50	Which executions are actuations?	Actuation01, Actuation03
CQ51	Which executions are observations?	Imputation01, Observation04
CQ52	Which observations are forecasted?	Forecasting01, Forecasting02
CQ53	Which observations are imputed?	Imputation01, Imputation02
CQ54	Which observations are outliers?	Outlier01, Outlier02
CQ55	Which is the cause of an outlier?	OutlierCausedByDeviceError
CQ56	Is a given execution a missing value?	No
CQ57	Which are the executions of a given collection?	Execution01, Execution02
CQ58	Which collection's member is a given execution?	Collection03

Table B.9: Requirements addressed by the EK4EEPSA ontology module.

ID	CQ	Answer
CQ59	What is a space adjacent to outdoor?	A space in contact with the exterior
CQ60	What is a bad insulated space?	A space with bad insulation
CQ61	What is a below ground level space?	A space located in a storey below ground
CQ62	What is a naturally enlightened space?	A space enlightened with an external source of light
CQ63	Which types of spaces are in a building?	Bad insulated spaces
CQ64	Which are the qualities affecting an adjacent to outdoor space's temperature?	Solar radiation, wind speed,...
CQ65	Which are the qualities affecting a bad insulated space's temperature?	Outdoor temperature, outdoor humidity,...
CQ66	Which are the qualities affecting an underground space's temperature?	Atmospheric pressure, occupancy,...
CQ67	Which are the qualities affecting a naturally enlightened space's temperature?	Cloud cover, sun position,...

## Appendix C

# Evaluation of the Ontology

This appendix shows the results of the evaluation of the ODPs and EEPsA ontology modules developed in this thesis.

### C.1 Design Correctness Metrics

In this section, the design correctness metrics obtained with OOPS! (Ontology Pitfall Scanner) are shown.

<b>Results for P04: Creating unconnected ontology elements.</b> <span style="float: right;">1 case   Minor</span>
<p>Ontology elements (classes, object properties and datatype properties) are created isolated, with no relation to the rest of the ontology.</p> <ul style="list-style-type: none"> <li>This pitfall appears in the following elements:           <ul style="list-style-type: none"> <li>&gt; <a href="http://purl.org/vocommons/voaf#Vocabulary">http://purl.org/vocommons/voaf#Vocabulary</a></li> </ul> </li> </ul>
<b>Results for P08: Missing annotations.</b> <span style="float: right;">1 case   Minor</span>
<p>This pitfall consists in creating an ontology element and failing to provide human readable annotations attached to it. Consequently, ontology elements lack annotation properties that label them (e.g. <code>rdfs:label</code>, <code>lemon:LexicalEntry</code>, <code>skos:prefLabel</code> or <code>skos:altLabel</code>) or that define them (e.g. <code>rdfs:comment</code> or <code>dc:description</code>). This pitfall is related to the guidelines provided in [5].</p> <ul style="list-style-type: none"> <li>The following elements have neither <code>rdfs:label</code> or <code>rdfs:comment</code> (nor <code>skos:definition</code>) defined:           <ul style="list-style-type: none"> <li>&gt; <a href="http://purl.org/vocommons/voaf#Vocabulary">http://purl.org/vocommons/voaf#Vocabulary</a></li> </ul> </li> </ul>
<b>Results for P10: Missing disjointness.</b> <span style="float: right;">ontology*   Important</span>
<p>The ontology lacks disjoint axioms between classes or between properties that should be defined as disjoint. This pitfall is related with the guidelines provided in [6], [2] and [7].</p> <p>*This pitfall applies to the ontology in general instead of specific elements.</p>
<b>Results for P13: Inverse relationships not explicitly declared.</b> <span style="float: right;">2 cases   Minor</span>
<p>This pitfall appears when any relationship (except for those that are defined as symmetric properties using <code>owl:SymmetricProperty</code>) does not have an inverse relationship (<code>owl:inverseOf</code>) defined within the ontology.</p> <ul style="list-style-type: none"> <li>This pitfall appears in the following elements:           <ul style="list-style-type: none"> <li>&gt; <a href="https://w3id.org/affectedBy#belongsTo">https://w3id.org/affectedBy#belongsTo</a></li> <li>&gt; <a href="https://w3id.org/affectedBy#affectedBy">https://w3id.org/affectedBy#affectedBy</a></li> </ul> </li> </ul>

Figure C.1: Design correctness metrics for the AffectedBy ODP.






<b>Results for P04: Creating unconnected ontology elements.</b>	<b>1 case   Minor</b> 
<p>Ontology elements (classes, object properties and datatype properties) are created isolated, with no relation to the rest of the ontology.</p> <ul style="list-style-type: none"> <li>This pitfall appears in the following elements:           <ul style="list-style-type: none"> <li>&gt; <a href="http://purl.org/vocommons/voaf#Vocabulary">http://purl.org/vocommons/voaf#Vocabulary</a></li> </ul> </li> </ul>	
<b>Results for P08: Missing annotations.</b>	<b>4 cases   Minor</b> 
<p>This pitfall consists in creating an ontology element and failing to provide human readable annotations attached to it. Consequently, ontology elements lack annotation properties that label them (e.g. <code>rdfs:label</code>, <code>lemon:LexicalEntry</code>, <code>skos:prefLabel</code> or <code>skos:altLabel</code>) or that define them (e.g. <code>rdfs:comment</code> or <code>dc:description</code>). This pitfall is related to the guidelines provided in [5].</p> <ul style="list-style-type: none"> <li>The following elements have neither <code>rdfs:label</code> or <code>rdfs:comment</code> defined:           <ul style="list-style-type: none"> <li>&gt; <a href="https://w3id.org/affectedBy#Quality">https://w3id.org/affectedBy#Quality</a></li> <li>&gt; <a href="https://w3id.org/affectedBy#FeatureOfInterest">https://w3id.org/affectedBy#FeatureOfInterest</a></li> <li>&gt; <a href="http://purl.org/vocommons/voaf#Vocabulary">http://purl.org/vocommons/voaf#Vocabulary</a></li> <li>&gt; <a href="https://w3id.org/affectedBy#belongsTo">https://w3id.org/affectedBy#belongsTo</a></li> </ul> </li> </ul>	
<b>Results for P10: Missing disjointness.</b>	<b>ontology*   Important</b> 
<p>The ontology lacks disjoint axioms between classes or between properties that should be defined as disjoint. This pitfall is related with the guidelines provided in [6], [2] and [7].</p> <p>*This pitfall applies to the ontology in general instead of specific elements.</p>	
<b>Results for P11: Missing domain or range in properties.</b>	<b>1 case   Important</b> 
<p>Object and/or datatype properties without domain or range (or none of them) are included in the ontology.</p> <ul style="list-style-type: none"> <li>This pitfall appears in the following elements:           <ul style="list-style-type: none"> <li>&gt; <a href="https://w3id.org/affectedBy#belongsTo">https://w3id.org/affectedBy#belongsTo</a></li> </ul> </li> <li><b>Tip:</b> Solving this pitfall may lead to new results for other pitfalls and suggestions. We encourage you to solve all cases when needed and see what else you can get from OOPS!</li> </ul>	
<b>Results for P13: Inverse relationships not explicitly declared.</b>	<b>7 cases   Minor</b> 
<p>This pitfall appears when any relationship (except for those that are defined as symmetric properties using <code>owl:SymmetricProperty</code>) does not have an inverse relationship (<code>owl:inverseOf</code>) defined within the ontology.</p> <ul style="list-style-type: none"> <li>This pitfall appears in the following elements:           <ul style="list-style-type: none"> <li>&gt; <a href="https://w3id.org/eep#usedProcedure">https://w3id.org/eep#usedProcedure</a></li> <li>&gt; <a href="https://w3id.org/eep#onQuality">https://w3id.org/eep#onQuality</a></li> <li>&gt; <a href="https://w3id.org/eep#implements">https://w3id.org/eep#implements</a></li> <li>&gt; <a href="https://w3id.org/eep#hasFeatureOfInterest">https://w3id.org/eep#hasFeatureOfInterest</a></li> <li>&gt; <a href="https://w3id.org/eep#forQuality">https://w3id.org/eep#forQuality</a></li> <li>&gt; <a href="https://w3id.org/eep#forFeatureOfInterest">https://w3id.org/eep#forFeatureOfInterest</a></li> <li>&gt; <a href="https://w3id.org/affectedBy#belongsTo">https://w3id.org/affectedBy#belongsTo</a></li> </ul> </li> </ul>	

Figure C.2: Design correctness metrics for the EEP ODP.






<b>Results for P04: Creating unconnected ontology elements.</b>	1 case   Minor 
<p>Ontology elements (classes, object properties and datatype properties) are created isolated, with no relation to the rest of the ontology.</p> <ul style="list-style-type: none"> <li>This pitfall appears in the following elements: <ul style="list-style-type: none"> <li><a href="http://purl.org/vocommons/voaf#Vocabulary">http://purl.org/vocommons/voaf#Vocabulary</a></li> </ul> </li> </ul>	
<b>Results for P08: Missing annotations.</b>	4 cases   Minor 
<p>This pitfall consists in creating an ontology element and failing to provide human readable annotations attached to it. Consequently, ontology elements lack annotation properties that label them (e.g. <code>rdfs:label</code>, <code>lemon:LexicalEntry</code>, <code>skos:prefLabel</code> or <code>skos:altLabel</code>) or that define them (e.g. <code>rdfs:comment</code> or <code>dc:description</code>). This pitfall is related to the guidelines provided in [5].</p> <ul style="list-style-type: none"> <li>The following elements have neither <code>rdfs:label</code> or <code>rdfs:comment</code> (nor <code>skos:definition</code>) defined: <ul style="list-style-type: none"> <li><a href="https://w3id.org/eep#Execution">https://w3id.org/eep#Execution</a></li> <li><a href="http://www.w3.org/2006/time#TemporalEntity">http://www.w3.org/2006/time#TemporalEntity</a></li> <li><a href="http://www.w3.org/2003/01/geo/wgs84_pos#SpatialThing">http://www.w3.org/2003/01/geo/wgs84_pos#SpatialThing</a></li> <li><a href="http://purl.org/vocommons/voaf#Vocabulary">http://purl.org/vocommons/voaf#Vocabulary</a></li> </ul> </li> </ul>	
<b>Results for P10: Missing disjointness.</b>	ontology*   Important 
<p>The ontology lacks disjoint axioms between classes or between properties that should be defined as disjoint. This pitfall is related with the guidelines provided in [6], [2] and [7].</p> <p>*This pitfall applies to the ontology in general instead of specific elements.</p>	
<b>Results for P11: Missing domain or range in properties.</b>	2 cases   Important 
<p>Object and/or datatype properties without domain or range (or none of them) are included in the ontology.</p> <ul style="list-style-type: none"> <li>This pitfall appears in the following elements: <ul style="list-style-type: none"> <li><a href="https://w3id.org/rc#hasResult">https://w3id.org/rc#hasResult</a></li> <li><a href="https://w3id.org/rc#hasSimpleResult">https://w3id.org/rc#hasSimpleResult</a></li> </ul> </li> <li><b>Tip:</b> Solving this pitfall may lead to new results for other pitfalls and suggestions. We encourage you to solve all cases when needed and see what else you can get from OOPS!</li> </ul>	
<b>Results for P13: Inverse relationships not explicitly declared.</b>	3 cases   Minor 
<p>This pitfall appears when any relationship (except for those that are defined as symmetric properties using <code>owl:SymmetricProperty</code>) does not have an inverse relationship (<code>owl:inverseOf</code>) defined within the ontology.</p> <ul style="list-style-type: none"> <li>This pitfall appears in the following elements: <ul style="list-style-type: none"> <li><a href="https://w3id.org/rc#hasTemporalContext">https://w3id.org/rc#hasTemporalContext</a></li> <li><a href="https://w3id.org/rc#hasSpatialContext">https://w3id.org/rc#hasSpatialContext</a></li> <li><a href="https://w3id.org/rc#hasResult">https://w3id.org/rc#hasResult</a></li> </ul> </li> </ul>	

Figure C.3: Design correctness metrics for the RC ODP.




<b>Results for P04: Creating unconnected ontology elements.</b>	<b>2 cases   Minor</b> 
Ontology elements (classes, object properties and datatype properties) are created isolated, with no relation to the rest of the ontology.	
<ul style="list-style-type: none"> <li>• This pitfall appears in the following elements:           <ul style="list-style-type: none"> <li>&gt; <a href="https://w3id.org/affectedBy#FeatureOfInterest">https://w3id.org/affectedBy#FeatureOfInterest</a></li> <li>&gt; <a href="http://purl.org/vocommons/voaf#Vocabulary">http://purl.org/vocommons/voaf#Vocabulary</a></li> </ul> </li> </ul>	
<b>Results for P08: Missing annotations.</b>	<b>5 cases   Minor</b> 
This pitfall consists in creating an ontology element and failing to provide human readable annotations attached to it. Consequently, ontology elements lack annotation properties that label them (e.g. <code>rdfs:label</code> , <code>lemon:LexicalEntry</code> , <code>skos:prefLabel</code> or <code>skos:altLabel</code> ) or that define them (e.g. <code>rdfs:comment</code> or <code>dc:description</code> ). This pitfall is related to the guidelines provided in [5].	
<ul style="list-style-type: none"> <li>• The following elements have neither <code>rdfs:label</code> or <code>rdfs:comment</code> (nor <code>skos:definition</code>) defined:           <ul style="list-style-type: none"> <li>&gt; <a href="https://w3id.org/affectedBy#FeatureOfInterest">https://w3id.org/affectedBy#FeatureOfInterest</a></li> <li>&gt; <a href="https://w3id.org/bot#Interface">https://w3id.org/bot#Interface</a></li> <li>&gt; <a href="https://w3id.org/bot#Zone">https://w3id.org/bot#Zone</a></li> <li>&gt; <a href="http://purl.org/vocommons/voaf#Vocabulary">http://purl.org/vocommons/voaf#Vocabulary</a></li> <li>&gt; <a href="https://w3id.org/bot#Element">https://w3id.org/bot#Element</a></li> </ul> </li> </ul>	
<b>Results for P10: Missing disjointness.</b>	<b>ontology*   Important</b> 
The ontology lacks disjoint axioms between classes or between properties that should be defined as disjoint. This pitfall is related with the guidelines provided in [6], [2] and [7].	
*This pitfall applies to the ontology in general instead of specific elements.	

Figure C.4: Design correctness metrics for the FoI4EEPSA ontology module.




<b>Results for P04: Creating unconnected ontology elements.</b>	<b>2 cases   Minor</b> 
Ontology elements (classes, object properties and datatype properties) are created isolated, with no relation to the rest of the ontology.	
<ul style="list-style-type: none"> <li>• This pitfall appears in the following elements:           <ul style="list-style-type: none"> <li>&gt; <a href="https://w3id.org/affectedBy#Quality">https://w3id.org/affectedBy#Quality</a></li> <li>&gt; <a href="http://purl.org/vocommons/voaf#Vocabulary">http://purl.org/vocommons/voaf#Vocabulary</a></li> </ul> </li> </ul>	
<b>Results for P08: Missing annotations.</b>	<b>2 cases   Minor</b> 
This pitfall consists in creating an ontology element and failing to provide human readable annotations attached to it. Consequently, ontology elements lack annotation properties that label them (e.g. <code>rdfs:label</code> , <code>lemon:LexicalEntry</code> , <code>skos:prefLabel</code> or <code>skos:altLabel</code> ) or that define them (e.g. <code>rdfs:comment</code> or <code>dc:description</code> ). This pitfall is related to the guidelines provided in [5].	
<ul style="list-style-type: none"> <li>• The following elements have neither <code>rdfs:label</code> or <code>rdfs:comment</code> (nor <code>skos:definition</code>) defined:           <ul style="list-style-type: none"> <li>&gt; <a href="https://w3id.org/affectedBy#Quality">https://w3id.org/affectedBy#Quality</a></li> <li>&gt; <a href="http://purl.org/vocommons/voaf#Vocabulary">http://purl.org/vocommons/voaf#Vocabulary</a></li> </ul> </li> </ul>	
<b>Results for P10: Missing disjointness.</b>	<b>ontology*   Important</b> 
The ontology lacks disjoint axioms between classes or between properties that should be defined as disjoint. This pitfall is related with the guidelines provided in [6], [2] and [7].	
*This pitfall applies to the ontology in general instead of specific elements.	

Figure C.5: Design correctness metrics for the Q4EEPSA ontology module.




<b>Results for P04: Creating unconnected ontology elements.</b>	<b>2 cases   Minor</b> 
<p>Ontology elements (classes, object properties and datatype properties) are created isolated, with no relation to the rest of the ontology.</p> <ul style="list-style-type: none"> <li>This pitfall appears in the following elements: <ul style="list-style-type: none"> <li>&gt; <a href="https://w3id.org/eep#Procedure">https://w3id.org/eep#Procedure</a></li> <li>&gt; <a href="http://purl.org/vocommons/voaf#Vocabulary">http://purl.org/vocommons/voaf#Vocabulary</a></li> </ul> </li> </ul>	
<b>Results for P08: Missing annotations.</b>	<b>2 cases   Minor</b> 
<p>This pitfall consists in creating an ontology element and failing to provide human readable annotations attached to it. Consequently, ontology elements lack annotation properties that label them (e.g. <code>rdfs:label</code>, <code>lemon:LexicalEntry</code>, <code>skos:prefLabel</code> or <code>skos:altLabel</code>) or that define them (e.g. <code>rdfs:comment</code> or <code>dc:description</code>). This pitfall is related to the guidelines provided in [5].</p> <ul style="list-style-type: none"> <li>The following elements have neither <code>rdfs:label</code> or <code>rdfs:comment</code> (nor <code>skos:definition</code>) defined: <ul style="list-style-type: none"> <li>&gt; <a href="https://w3id.org/eep#Procedure">https://w3id.org/eep#Procedure</a></li> <li>&gt; <a href="http://purl.org/vocommons/voaf#Vocabulary">http://purl.org/vocommons/voaf#Vocabulary</a></li> </ul> </li> </ul>	
<b>Results for P10: Missing disjointness.</b>	<b>ontology*   Important</b> 
<p>The ontology lacks disjoint axioms between classes or between properties that should be defined as disjoint. This pitfall is related with the guidelines provided in [6], [2] and [7].</p> <p>*This pitfall applies to the ontology in general instead of specific elements.</p>	

Figure C.6: Design correctness metrics for the P4EEPSA ontology module.




<b>Results for P04: Creating unconnected ontology elements.</b>	<b>2 cases   Minor</b> 
<p>Ontology elements (classes, object properties and datatype properties) are created isolated, with no relation to the rest of the ontology.</p> <ul style="list-style-type: none"> <li>This pitfall appears in the following elements: <ul style="list-style-type: none"> <li>&gt; <a href="https://w3id.org/eep#Executor">https://w3id.org/eep#Executor</a></li> <li>&gt; <a href="http://purl.org/vocommons/voaf#Vocabulary">http://purl.org/vocommons/voaf#Vocabulary</a></li> </ul> </li> </ul>	
<b>Results for P08: Missing annotations.</b>	<b>2 cases   Minor</b> 
<p>This pitfall consists in creating an ontology element and failing to provide human readable annotations attached to it. Consequently, ontology elements lack annotation properties that label them (e.g. <code>rdfs:label</code>, <code>lemon:LexicalEntry</code>, <code>skos:prefLabel</code> or <code>skos:altLabel</code>) or that define them (e.g. <code>rdfs:comment</code> or <code>dc:description</code>). This pitfall is related to the guidelines provided in [5].</p> <ul style="list-style-type: none"> <li>The following elements have neither <code>rdfs:label</code> or <code>rdfs:comment</code> (nor <code>skos:definition</code>) defined: <ul style="list-style-type: none"> <li>&gt; <a href="https://w3id.org/eep#Executor">https://w3id.org/eep#Executor</a></li> <li>&gt; <a href="http://purl.org/vocommons/voaf#Vocabulary">http://purl.org/vocommons/voaf#Vocabulary</a></li> </ul> </li> </ul>	
<b>Results for P10: Missing disjointness.</b>	<b>ontology*   Important</b> 
<p>The ontology lacks disjoint axioms between classes or between properties that should be defined as disjoint. This pitfall is related with the guidelines provided in [6], [2] and [7].</p> <p>*This pitfall applies to the ontology in general instead of specific elements.</p>	

Figure C.7: Design correctness metrics for the EXR4EEPSA ontology module.




<b>Results for P04: Creating unconnected ontology elements.</b>	<b>1 case   Minor</b> 
Ontology elements (classes, object properties and datatype properties) are created isolated, with no relation to the rest of the ontology.	
<ul style="list-style-type: none"> <li>This pitfall appears in the following elements:           <ul style="list-style-type: none"> <li>&gt; <a href="http://purl.org/vocommons/voaf#Vocabulary">http://purl.org/vocommons/voaf#Vocabulary</a></li> </ul> </li> </ul>	
<b>Results for P08: Missing annotations.</b>	<b>2 cases   Minor</b> 
This pitfall consists in creating an ontology element and failing to provide human readable annotations attached to it. Consequently, ontology elements lack annotation properties that label them (e.g. <code>rdfs:label</code> , <code>lemon:LexicalEntry</code> , <code>skos:prefLabel</code> or <code>skos:altLabel</code> ) or that define them (e.g. <code>rdfs:comment</code> or <code>dc:description</code> ). This pitfall is related to the guidelines provided in [5].	
<ul style="list-style-type: none"> <li>The following elements have neither <code>rdfs:label</code> or <code>rdfs:comment</code> (nor <code>skos:definition</code>) defined:           <ul style="list-style-type: none"> <li>&gt; <a href="https://w3id.org/eep#Execution">https://w3id.org/eep#Execution</a></li> <li>&gt; <a href="http://purl.org/vocommons/voaf#Vocabulary">http://purl.org/vocommons/voaf#Vocabulary</a></li> </ul> </li> </ul>	
<b>Results for P10: Missing disjointness.</b>	<b>ontology*   Important</b> 
The ontology lacks disjoint axioms between classes or between properties that should be defined as disjoint. This pitfall is related with the guidelines provided in [5], [2] and [7].	
*This pitfall applies to the ontology in general instead of specific elements.	

Figure C.8: Design correctness metrics for the EXN4EEPSA ontology module.




<b>Results for P04: Creating unconnected ontology elements.</b>	<b>2 cases   Minor</b> 
Ontology elements (classes, object properties and datatype properties) are created isolated, with no relation to the rest of the ontology.	
<ul style="list-style-type: none"> <li>This pitfall appears in the following elements:           <ul style="list-style-type: none"> <li>&gt; <a href="https://w3id.org/eepsa/q4eepsa#IndoorTemperature">https://w3id.org/eepsa/q4eepsa#IndoorTemperature</a></li> <li>&gt; <a href="http://purl.org/vocommons/voaf#Vocabulary">http://purl.org/vocommons/voaf#Vocabulary</a></li> </ul> </li> </ul>	
<b>Results for P08: Missing annotations.</b>	<b>3 cases   Minor</b> 
This pitfall consists in creating an ontology element and failing to provide human readable annotations attached to it. Consequently, ontology elements lack annotation properties that label them (e.g. <code>rdfs:label</code> , <code>lemon:LexicalEntry</code> , <code>skos:prefLabel</code> or <code>skos:altLabel</code> ) or that define them (e.g. <code>rdfs:comment</code> or <code>dc:description</code> ). This pitfall is related to the guidelines provided in [5].	
<ul style="list-style-type: none"> <li>The following elements have neither <code>rdfs:label</code> or <code>rdfs:comment</code> (nor <code>skos:definition</code>) defined:           <ul style="list-style-type: none"> <li>&gt; <a href="https://w3id.org/eepsa/q4eepsa#IndoorTemperature">https://w3id.org/eepsa/q4eepsa#IndoorTemperature</a></li> <li>&gt; <a href="https://w3id.org/bot#Space">https://w3id.org/bot#Space</a></li> <li>&gt; <a href="http://purl.org/vocommons/voaf#Vocabulary">http://purl.org/vocommons/voaf#Vocabulary</a></li> </ul> </li> </ul>	
<b>Results for P10: Missing disjointness.</b>	<b>ontology*   Important</b> 
The ontology lacks disjoint axioms between classes or between properties that should be defined as disjoint. This pitfall is related with the guidelines provided in [6], [2] and [7].	
*This pitfall applies to the ontology in general instead of specific elements.	

Figure C.9: Design correctness metrics for the EK4EEPSA ontology module.

## C.2 Structural Metrics

In this section, the structural metrics obtained with the Protégé's Ontology Metrics Tab are shown.



<b>Metrics</b>	
<b>Axiom</b>	<b>62</b>
Logical axiom count	<b>12</b>
Declaration axioms count	<b>19</b>
Class count	<b>3</b>
Object property count	<b>3</b>
Data property count	0
Individual count	<b>1</b>
DL expressivity	ALERIF+
<b>Class axioms</b>	
<b>SubClassOf</b>	<b>1</b>
EquivalentClasses	0
DisjointClasses	0
GCI count	0
Hidden GCI Count	0
<b>Object property axioms</b>	
<b>SubObjectPropertyOf</b>	<b>1</b>
EquivalentObjectProperties	0
InverseObjectProperties	0
DisjointObjectProperties	0
<b>FunctionalObjectProperty</b>	<b>1</b>
InverseFunctionalObjectProperty	0
<b>TransitiveObjectProperty</b>	<b>1</b>
SymmetricObjectProperty	0
AsymmetricObjectProperty	0
ReflexiveObjectProperty	0
IrreflexiveObjectProperty	0
<b>ObjectPropertyDomain</b>	<b>3</b>
<b>ObjectPropertyRange</b>	<b>3</b>
<b>SubPropertyChainOf</b>	<b>1</b>
<b>Data property axioms</b>	
SubDataPropertyOf	0
EquivalentDataProperties	0
DisjointDataProperties	0
FunctionalDataProperty	0
DataPropertyDomain	0
DataPropertyRange	0
<b>Individual axioms</b>	
<b>ClassAssertion</b>	<b>1</b>
ObjectPropertyAssertion	0
DataPropertyAssertion	0
NegativeObjectPropertyAssertion	0
NegativeDataPropertyAssertion	0
SameIndividual	0
DifferentIndividuals	0
<b>Annotation axioms</b>	
<b>AnnotationAssertion</b>	<b>31</b>
AnnotationPropertyDomain	0
AnnotationPropertyRangeOf	0

Figure C.10: Structural metrics for the AffectedBy ODP.

Metrics	
Axiom	<b>80</b>
Logical axiom count	<b>25</b>
Declaration axioms count	<b>15</b>
Class count	<b>6</b>
Object property count	<b>8</b>
Data property count	0
Individual count	<b>1</b>
DL expressivity	ALERIF
Class axioms	
SubClassOf	<b>3</b>
EquivalentClasses	0
DisjointClasses	0
GCI count	0
Hidden GCI Count	0
Object property axioms	
SubObjectPropertyOf	0
EquivalentObjectProperties	0
InverseObjectProperties	0
DisjointObjectProperties	0
<b>FunctionalObjectProperty</b>	<b>3</b>
InverseFunctionalObjectProperty	0
TransitiveObjectProperty	0
SymmetricObjectProperty	0
AsymmetricObjectProperty	0
ReflexiveObjectProperty	0
IrreflexiveObjectProperty	0
<b>ObjectPropertyDomain</b>	<b>7</b>
<b>ObjectPropertyRange</b>	<b>7</b>
<b>SubPropertyChainOf</b>	<b>4</b>
Data property axioms	
SubDataPropertyOf	0
EquivalentDataProperties	0
DisjointDataProperties	0
FunctionalDataProperty	0
DataPropertyDomain	0
DataPropertyRange	0
Individual axioms	
<b>ClassAssertion</b>	<b>1</b>
ObjectPropertyAssertion	0
DataPropertyAssertion	0
NegativeObjectPropertyAssertion	0
NegativeDataPropertyAssertion	0
SameIndividual	0
DifferentIndividuals	0
Annotation axioms	
<b>AnnotationAssertion</b>	<b>40</b>
AnnotationPropertyDomain	0
AnnotationPropertyRangeOf	0

Figure C.11: Structural metrics for the EEP ODP.

Metrics	
Axiom	<b>40</b>
Logical axiom count	<b>9</b>
Declaration axioms count	<b>11</b>
Class count	<b>4</b>
Object property count	<b>3</b>
Data property count	<b>2</b>
Individual count	<b>1</b>
DL expressivity	AL(D)
<b>Class axioms</b>	
SubClassOf	0
EquivalentClasses	0
DisjointClasses	0
GCI count	0
Hidden GCI Count	0
<b>Object property axioms</b>	
SubObjectPropertyOf	0
EquivalentObjectProperties	0
InverseObjectProperties	0
DisjointObjectProperties	0
FunctionalObjectProperty	0
InverseFunctionalObjectProperty	0
TransitiveObjectProperty	0
SymmetricObjectProperty	0
AsymmetricObjectProperty	0
ReflexiveObjectProperty	0
IrreflexiveObjectProperty	0
ObjectPropertyDomain	<b>3</b>
ObjectPropertyRange	<b>2</b>
SubPropertyChainOf	0
<b>Data property axioms</b>	
SubDataPropertyOf	0
EquivalentDataProperties	0
DisjointDataProperties	0
FunctionalDataProperty	0
DataPropertyDomain	<b>2</b>
DataPropertyRange	<b>1</b>
<b>Individual axioms</b>	
ClassAssertion	<b>1</b>
ObjectPropertyAssertion	0
DataPropertyAssertion	0
NegativeObjectPropertyAssertion	0
NegativeDataPropertyAssertion	0
SameIndividual	0
DifferentIndividuals	0
<b>Annotation axioms</b>	
AnnotationAssertion	<b>20</b>
AnnotationPropertyDomain	0
AnnotationPropertyRangeOf	0

Figure C.12: Structural metrics for the RC ODP.

Metrics	
<b>Axiom</b>	<b>128</b>
Logical axiom count	27
Declaration axioms count	37
Class count	17
Object property count	0
Data property count	5
Individual count	1
DL expressivity	AL(D)
<b>Class axioms</b>	
SubClassOf	13
EquivalentClasses	3
DisjointClasses	0
GCI count	0
Hidden GCI Count	3
<b>Object property axioms</b>	
SubObjectPropertyOf	0
EquivalentObjectProperties	0
InverseObjectProperties	0
DisjointObjectProperties	0
FunctionalObjectProperty	0
InverseFunctionalObjectProperty	0
TransitiveObjectProperty	0
SymmetricObjectProperty	0
AsymmetricObjectProperty	0
ReflexiveObjectProperty	0
IrreflexiveObjectProperty	0
ObjectPropertyDomain	0
ObjectPropertyRange	0
SubPropertyChainOf	0
<b>Data property axioms</b>	
SubDataPropertyOf	0
EquivalentDataProperties	0
DisjointDataProperties	0
FunctionalDataProperty	0
DataPropertyDomain	5
DataPropertyRange	5
<b>Individual axioms</b>	
ClassAssertion	1
ObjectPropertyAssertion	0
DataPropertyAssertion	0
NegativeObjectPropertyAssertion	0
NegativeDataPropertyAssertion	0
SameIndividual	0
DifferentIndividuals	0
<b>Annotation axioms</b>	
AnnotationAssertion	64
AnnotationPropertyDomain	0
AnnotationPropertyRangeOf	0

Figure C.13: Structural metrics for the FoI4EPPSA ontology module.

<b>Metrics</b>	
<b>Axiom</b>	<b>197</b>
Logical axiom count	30
Declaration axioms count	43
Class count	30
Object property count	0
Data property count	0
Individual count	1
DL expressivity	AL
<b>Class axioms</b>	
<b>SubClassOf</b>	<b>29</b>
EquivalentClasses	0
DisjointClasses	0
GCI count	0
Hidden GCI Count	0
<b>Object property axioms</b>	
SubObjectPropertyOf	0
EquivalentObjectProperties	0
InverseObjectProperties	0
DisjointObjectProperties	0
FunctionalObjectProperty	0
InverseFunctionalObjectProperty	0
TransitiveObjectProperty	0
SymmetricObjectProperty	0
AsymmetricObjectProperty	0
ReflexiveObjectProperty	0
IrreflexiveObjectProperty	0
ObjectPropertyDomain	0
ObjectPropertyRange	0
SubPropertyChainOf	0
<b>Data property axioms</b>	
SubDataPropertyOf	0
EquivalentDataProperties	0
DisjointDataProperties	0
FunctionalDataProperty	0
DataPropertyDomain	0
DataPropertyRange	0
<b>Individual axioms</b>	
<b>ClassAssertion</b>	<b>1</b>
ObjectPropertyAssertion	0
DataPropertyAssertion	0
NegativeObjectPropertyAssertion	0
NegativeDataPropertyAssertion	0
SameIndividual	0
DifferentIndividuals	0
<b>Annotation axioms</b>	
<b>AnnotationAssertion</b>	<b>124</b>
AnnotationPropertyDomain	0
AnnotationPropertyRangeOf	0

Figure C.14: Structural metrics for the Q4EEPSA ontology module.

Metrics	
<b>Axiom</b>	<b>40</b>
Logical axiom count	5
Declaration axioms count	19
<b>Class count</b>	<b>6</b>
Object property count	0
Data property count	0
<b>Individual count</b>	<b>1</b>
DL expressivity	AL
<b>Class axioms</b>	
<b>SubClassOf</b>	<b>4</b>
EquivalentClasses	0
DisjointClasses	0
GCI count	0
Hidden GCI Count	0
<b>Object property axioms</b>	
SubObjectPropertyOf	0
EquivalentObjectProperties	0
InverseObjectProperties	0
DisjointObjectProperties	0
FunctionalObjectProperty	0
InverseFunctionalObjectProperty	0
TransitiveObjectProperty	0
SymmetricObjectProperty	0
AsymmetricObjectProperty	0
ReflexiveObjectProperty	0
IrreflexiveObjectProperty	0
ObjectPropertyDomain	0
ObjectPropertyRange	0
SubPropertyChainOf	0
<b>Data property axioms</b>	
SubDataPropertyOf	0
EquivalentDataProperties	0
DisjointDataProperties	0
FunctionalDataProperty	0
DataPropertyDomain	0
DataPropertyRange	0
<b>Individual axioms</b>	
<b>ClassAssertion</b>	<b>1</b>
ObjectPropertyAssertion	0
DataPropertyAssertion	0
NegativeObjectPropertyAssertion	0
NegativeDataPropertyAssertion	0
SameIndividual	0
DifferentIndividuals	0
<b>Annotation axioms</b>	
<b>AnnotationAssertion</b>	<b>16</b>
AnnotationPropertyDomain	0
AnnotationPropertyRangeOf	0

Figure C.15: Structural metrics for the P4EEPSA ontology module.

Metrics	
<b>Axiom</b>	<b>207</b>
Logical axiom count	33
Declaration axioms count	47
<b>Class count</b>	<b>33</b>
Object property count	0
Data property count	0
<b>Individual count</b>	<b>1</b>
DL expressivity	AL
<b>Class axioms</b>	
<b>SubClassOf</b>	<b>32</b>
EquivalentClasses	0
DisjointClasses	0
GCI count	0
Hidden GCI Count	0
<b>Object property axioms</b>	
SubObjectPropertyOf	0
EquivalentObjectProperties	0
InverseObjectProperties	0
DisjointObjectProperties	0
FunctionalObjectProperty	0
InverseFunctionalObjectProperty	0
TransitiveObjectProperty	0
SymmetricObjectProperty	0
AsymmetricObjectProperty	0
ReflexiveObjectProperty	0
IrreflexiveObjectProperty	0
ObjectPropertyDomain	0
ObjectPropertyRange	0
SubPropertyChainOf	0
<b>Data property axioms</b>	
SubDataPropertyOf	0
EquivalentDataProperties	0
DisjointDataProperties	0
FunctionalDataProperty	0
DataPropertyDomain	0
DataPropertyRange	0
<b>Individual axioms</b>	
<b>ClassAssertion</b>	<b>1</b>
ObjectPropertyAssertion	0
DataPropertyAssertion	0
NegativeObjectPropertyAssertion	0
NegativeDataPropertyAssertion	0
SameIndividual	0
DifferentIndividuals	0
<b>Annotation axioms</b>	
<b>AnnotationAssertion</b>	<b>127</b>
AnnotationPropertyDomain	0
AnnotationPropertyRangeOf	0

Figure C.16: Structural metrics for the EXR4EEPSA ontology module.

Metrics	
<b>Axiom</b>	<b>114</b>
Logical axiom count	19
Declaration axioms count	31
Class count	16
Object property count	2
Data property count	0
Individual count	1
DL expressivity	ALI
<b>Class axioms</b>	
<b>SubClassOf</b>	<b>13</b>
EquivalentClasses	0
DisjointClasses	0
GCI count	0
Hidden GCI Count	0
<b>Object property axioms</b>	
SubObjectPropertyOf	0
EquivalentObjectProperties	0
<b>InverseObjectProperties</b>	<b>1</b>
DisjointObjectProperties	0
FunctionalObjectProperty	0
InverseFunctionalObjectProperty	0
TransitiveObjectProperty	0
SymmetricObjectProperty	0
AsymmetricObjectProperty	0
ReflexiveObjectProperty	0
IrreflexiveObjectProperty	0
<b>ObjectPropertyDomain</b>	<b>2</b>
<b>ObjectPropertyRange</b>	<b>2</b>
SubPropertyChainOf	0
<b>Data property axioms</b>	
SubDataPropertyOf	0
EquivalentDataProperties	0
DisjointDataProperties	0
FunctionalDataProperty	0
DataPropertyDomain	0
DataPropertyRange	0
<b>Individual axioms</b>	
<b>ClassAssertion</b>	<b>1</b>
ObjectPropertyAssertion	0
DataPropertyAssertion	0
NegativeObjectPropertyAssertion	0
NegativeDataPropertyAssertion	0
SameIndividual	0
DifferentIndividuals	0
<b>Annotation axioms</b>	
<b>AnnotationAssertion</b>	<b>64</b>
AnnotationPropertyDomain	0
AnnotationPropertyRangeOf	0

Figure C.17: Structural metrics for the EXN4EEPSA ontology module.



Metrics	
Axiom	<b>81</b>
Logical axiom count	<b>20</b>
Declaration axioms count	<b>29</b>
Class count	<b>25</b>
Object property count	<b>4</b>
Data property count	0
Individual count	<b>1</b>
DL expressivity	ALC
Class axioms	
SubClassOf	<b>16</b>
EquivalentClasses	<b>3</b>
DisjointClasses	0
GCI count	0
Hidden GCI Count	<b>3</b>
Object property axioms	
SubObjectPropertyOf	0
EquivalentObjectProperties	0
InverseObjectProperties	0
DisjointObjectProperties	0
FunctionalObjectProperty	0
InverseFunctionalObjectProperty	0
TransitiveObjectProperty	0
SymmetricObjectProperty	0
AsymmetricObjectProperty	0
ReflexiveObjectProperty	0
IrreflexiveObjectProperty	0
ObjectPropertyDomain	0
ObjectPropertyRange	0
SubPropertyChainOf	0
Data property axioms	
SubDataPropertyOf	0
EquivalentDataProperties	0
DisjointDataProperties	0
FunctionalDataProperty	0
DataPropertyDomain	0
DataPropertyRange	0
Individual axioms	
ClassAssertion	<b>1</b>
ObjectPropertyAssertion	0
DataPropertyAssertion	0
NegativeObjectPropertyAssertion	0
NegativeDataPropertyAssertion	0
SameIndividual	0
DifferentIndividuals	0
Annotation axioms	
AnnotationAssertion	<b>32</b>
AnnotationPropertyDomain	0
AnnotationPropertyRangeOf	0

Figure C.18: Structural metrics for the EK4EEPSA ontology module.

### C.3 Ontology Module Quality Metrics

In this section, the ontology module quality metrics obtained with TOMM (Tool for Ontology Module Metrics) are shown.

**AffectedBy ODP**

No. of classes in ontology: 3  
 No. of OP in ontology: 3  
 No. of DP in ontology: 0  
 No. of Ind in ontology: 1  
 Size of ontology: 7  
 Atomic size of module: 4.285714285714286  
 No. of axioms in ontology: 62  
 Appropriateness of ontology: 0.14423216139535733  
 Intra module distance: 0.0  
 Cohesion of ontology: 0.0  
 Attribute richness of ontology: 0.6666666666666666  
 Inheritance richness of ontology: NaN  
 Encapsulation of ontology: 0.8472222222222222  
 Coupling of ontology: 0.0  
 Is the ontology independent?: false  
 Redundancy of ontology set: 0.1055350553505535

Relative size of module: 0.04516129032258064  
 Relative intra module distance of module: Infinity  
 Correctness of module: True, the module is logically correct, no new axioms have been added to the ontology.  
 Completeness of ontology: True, the module is logically complete. The meaning of every entity is preserved as in the source ontology.  
 Time taken for processing: 0.21 seconds, 0.0035 minutes, 5.833333333333333E-5 hours.

**EEP ODP**

No. of classes in ontology: 6  
 No. of OP in ontology: 10  
 No. of DP in ontology: 0  
 No. of Ind in ontology: 2  
 Size of ontology: 18  
 Atomic size of module: 5.222222222222222  
 No. of axioms in ontology: 137  
 Appropriateness of ontology: 0.5751127945603786  
 Intra module distance: 0.0  
 Cohesion of ontology: 0.0  
 Attribute richness of ontology: 1.3333333333333333  
 Inheritance richness of ontology: NaN  
 Encapsulation of ontology: 0.9429951690821256  
 Coupling of ontology: 0.0  
 Is the ontology independent?: false  
 Redundancy of ontology set: 0.1055350553505535

Relative size of module: 0.11612903225806452  
 Relative intra module distance of module: Infinity  
 Correctness of module: True, the module is logically correct, no new axioms have

been added to the ontology.

Completeness of ontology: False, the module is not logically complete. The meaning of the entity: <<https://w3id.org/affectedBy#affectedBy>> is not preserved in the module as it is in the source ontology.

Time taken for processing: 0.231 seconds, 0.00385 minutes, 6.416666666666666E-5 hours.

### **RC ODP**

No. of classes in ontology: 4

No. of OP in ontology: 3

No. of DP in ontology: 2

No. of Ind in ontology: 1

Size of ontology: 10

Atomic size of module: 2.7

No. of axioms in ontology: 40

Appropriateness of ontology: 0.06184665997806821

Intra module distance: 0.0

Cohesion of ontology: 0.0

Attribute richness of ontology: 0.0

Inheritance richness of ontology: NaN

Encapsulation of ontology: 0.95

Coupling of ontology: 0.0

Is the ontology independent?: false

Redundancy of ontology set: 0.1055350553505535

Relative size of module: 0.06451612903225806

Relative intra module distance of module: Infinity

Correctness of module: True, the module is logically correct, no new axioms have been added to the ontology.

Completeness of ontology: False, the module is not logically complete. The meaning of the entity: <<https://w3id.org/eep#Execution>> is not preserved in the module as it is in the source ontology.

Time taken for processing: 0.247 seconds, 0.004116666666666667 minutes, 6.861111111111111E-5 hours.

### **FoI4EEPSA ontology module**

No. of classes in ontology: 21

No. of OP in ontology: 14

No. of DP in ontology: 7

No. of Ind in ontology: 3

Size of ontology: 45

Atomic size of module: 4.488888888888889

No. of axioms in ontology: 519

Appropriateness of ontology: -1.0

Intra module distance: 500.0

Cohesion of ontology: 0.04086333720480066  
 Attribute richness of ontology: 0.09523809523809523  
 Inheritance richness of ontology: 2.125  
 Encapsulation of ontology: 0.98330122029544  
 Coupling of ontology: 0.0  
 Is the ontology independent?: false  
 Redundancy of ontology set: 0.1055350553505535

Relative size of module: 0.2903225806451613  
 Relative intra module distance of module: 1.0  
 Correctness of module: False, the module is not logically correct. The following axiom exists in the module but not in the original ontology:  
 DataPropertyAssertion(<http://xmlns.com/foaf/0.1/name> \_:genid154 "Pieter Pauwels")  
 Completeness of ontology: True, the module is logically complete. The meaning of every entity is preserved as in the source ontology.  
 Time taken for processing: 0.14 seconds, 0.0023333333333333335 minutes, 3.888888888888889E-5 hours.

#### **Q4EEPSA ontology module**

No. of classes in ontology: 30  
 No. of OP in ontology: 0  
 No. of DP in ontology: 0  
 No. of Ind in ontology: 1  
 Size of ontology: 31  
 Atomic size of module: 2.935483870967742  
 No. of axioms in ontology: 197  
 Appropriateness of ontology: 0.8931442160683094  
 Intra module distance: 1194.0  
 Cohesion of ontology: 0.18572796934865904  
 Attribute richness of ontology: 0.0  
 Inheritance richness of ontology: 4.833333333333333  
 Encapsulation of ontology: 0.9599548787366047  
 Coupling of ontology: 0.0  
 Is the ontology independent?: false  
 Redundancy of ontology set: 0.1055350553505535

Relative size of module: 0.2  
 Relative intra module distance of module: 1.015075376884422  
 Correctness of module: True, the module is logically correct, no new axioms have been added to the ontology.  
 Completeness of ontology: False, the module is not logically complete. The meaning of the entity: <https://w3id.org/affectedBy#Quality> is not preserved in the module as it is in the source ontology.  
 Time taken for processing: 0.188 seconds, 0.0031333333333333335 minutes, 5.222222222222223E-5 hours.

**P4EEPSA ontology module**

No. of classes in ontology: 6  
 No. of OP in ontology: 0  
 No. of DP in ontology: 0  
 No. of Ind in ontology: 1  
 Size of ontology: 7  
 Atomic size of module: 2.4285714285714284  
 No. of axioms in ontology: 40  
 Appropriateness of ontology: 0.06184665997806821  
 Intra module distance: 16.0  
 Cohesion of ontology: 0.23333333333333334  
 Attribute richness of ontology: 0.0  
 Inheritance richness of ontology: 4.0  
 Encapsulation of ontology: 0.8083333333333333  
 Coupling of ontology: 0.0  
 Is the ontology independent?: false  
 Redundancy of ontology set: 0.1055350553505535

Relative size of module: 0.04516129032258064  
 Relative intra module distance of module: 2.125  
 Correctness of module: True, the module is logically correct, no new axioms have been added to the ontology.  
 Completeness of ontology: True, the module is logically complete. The meaning of every entity is preserved as in the source ontology.  
 Time taken for processing: 0.156 seconds, 0.0026 minutes, 4.333333333333334E-5 hours.

**EXR4EEPSA ontology module**

No. of classes in ontology: 33  
 No. of OP in ontology: 0  
 No. of DP in ontology: 0  
 No. of Ind in ontology: 1  
 Size of ontology: 34  
 Atomic size of module: 2.9411764705882355  
 No. of axioms in ontology: 207  
 Appropriateness of ontology: 0.9287633280968262  
 Intra module distance: 1649.0  
 Cohesion of ontology: 0.1727114898989899  
 Attribute richness of ontology: 0.0  
 Inheritance richness of ontology: 5.333333333333333  
 Encapsulation of ontology: 0.9624261943102523  
 Coupling of ontology: 0.0  
 Is the ontology independent?: false  
 Redundancy of ontology set: 0.1055350553505535

Relative size of module: 0.21935483870967742  
 Relative intra module distance of module: 1.010915706488781  
 Correctness of module: True, the module is logically correct, no new axioms have

been added to the ontology.

Completeness of ontology: True, the module is logically complete. The meaning of every entity is preserved as in the source ontology.

Time taken for processing: 0.111 seconds, 0.00185 minutes, 3.0833333333333335E-5 hours.

#### **EXN4EEPSA ontology module**

No. of classes in ontology: 16

No. of OP in ontology: 2

No. of DP in ontology: 0

No. of Ind in ontology: 1

Size of ontology: 19

Atomic size of module: 3.0

No. of axioms in ontology: 114

Appropriateness of ontology: 0.431104854657681

Intra module distance: 271.0

Cohesion of ontology: 0.1277233115468409

Attribute richness of ontology: 0.0

Inheritance richness of ontology: 2.6

Encapsulation of ontology: 0.9317738791423003

Coupling of ontology: 0.0

Is the ontology independent?: false

Redundancy of ontology set: 0.10236220472440945

Relative size of module: 0.11728395061728394

Relative intra module distance of module: 1.066420664206642

Correctness of module: True, the module is logically correct, no new axioms have been added to the ontology.

Completeness of ontology: False, the module is not logically complete. The meaning of the entity: <<https://w3id.org/eep#Execution>> is not preserved in the module as it is in the source ontology.

Time taken for processing: 0.074 seconds, 0.0012333333333333332 minutes, 2.0555555555555555E-5 hours.

#### **EK4EEPSA ontology module**

No. of classes in ontology: 25

No. of OP in ontology: 4

No. of DP in ontology: 0

No. of Ind in ontology: 1

Size of ontology: 30

Atomic size of module: 2.6666666666666665

No. of axioms in ontology: 81

Appropriateness of ontology: 0.23741268501935214

Intra module distance: 32.0

Cohesion of ontology: 0.017241379310344827

Attribute richness of ontology: 1.56

Inheritance richness of ontology: 4.0  
Encapsulation of ontology: 0.9094650205761317  
Coupling of ontology: 0.0  
Is the ontology independent?: false  
Redundancy of ontology set: 0.1055350553505535

Relative size of module: 0.1935483870967742  
Relative intra module distance of module: 16.125  
Correctness of module: False, the module is not logically correct. The following axiom exists in the module but not in the original ontology:  
AnnotationAssertion(<http://www.w3.org/2003/06/sw-vocab-status/ns#term.-status> <https://w3id.org/ee psa/ek4ee psa#BelowGroundLevelSpaceIndoorTemperature> “stable”)  
Completeness of ontology: False, the module is not logically complete. The meaning of the entity: <https://w3id.org/bot#hasStorey> is not preserved in the module as it is in the source ontology.  
Time taken for processing: 0.049 seconds, 8.166666666666667E-4 minutes, 1.3611111111111111E-5 hours.

**EEPSA ontology**

No. of classes in ontology: 112  
No. of OP in ontology: 29  
No. of DP in ontology: 9  
No. of Ind in ontology: 12  
Size of ontology: 162  
Atomic size of ontology: 4.055555555555555  
No. of axioms in ontology: 1256  
Appropriateness of ontology: -1.0  
Attribute richness of ontology: 0.4375  
Inheritance richness of ontology: 3.6785714285714284  
Time taken for processing: 0.011 seconds, 1.8333333333333334E-4 minutes, 3.0555555555555556E-6 hours.