An Ontology-Based Semantic Building Post-Occupancy Evaluation Framework and Its Application

A thesis submitted for the degree of Doctor of Philosophy

By

Yuanhong Zhao

Department of Civil and Environmental Engineering

Brunel University London

January 2022

Abstract

Catering to sustainable development in Architecture, Engineering and Construction (AEC) industry, many building performance evaluation (BPE) schemas have been developed to support building assessment and aim to narrow down the performance gap. Post-Occupancy Evaluation (POE), viewed as a sub-process of BPE, is a systematic method to obtain feedback on building performance in use. However, building evaluation is a complex and knowledge-intensive process with scattered and fragmented knowledge, it is time-consuming and error-prone to acquire explicit knowledge.

Benefiting from the advantages of Semantic Web technology in knowledge conceptualization, ontology, as the core of the Semantic Web, has been widely taken as an effective method for knowledge management, information representation and extraction, and logical inference in the AEC industry, especially in the BPE field. However, most of the existing ontologies in the AEC industry are lightweight ontologies that mainly focus on building a structured system to represent the specific domain knowledge or information, without developing formal axioms and constraints to provide higher expressivity. Moreover, the research focus of ontology in building assessment is mainly on energy-related fields, and there is not a comprehensive POE ontology yet, especially with the focus on building occupant satisfaction, which is the starting point of this research.

This research develops an ontology-based post-occupancy evaluation framework dedicated to building performance assessment, with the ultimate aim of optimizing building operation and improving building occupants' use experience quality and well-being. In the developed framework, a heavyweight ontology is developed to structure the fragmented building performance assessment knowledge in the POE domain. In POE ontology, the building occupants' needs for building performance are generalized and classified, and the corresponded building performance assessment knowledge is formalized. In addition, a set of SWRL (Semantic Web Rule Language) rules and SQWRL (Semantic Query-Enhanced Web Rule Language) query rules are developed based on the benchmarking evaluation axioms to enable automatic rule-based reasoning and query in different identified application scenarios. This ontology model enables effective POE-related knowledge retrieving and sharing, and promotes its implementation in the POE domain.

To validate the developed framework, a case study is carried out facilitated by the Building Use Studies (BUS) Methodology to illustrate its feasibility and effectiveness in different application scenarios. This research concludes that the proposed ontology-based POE framework has the capability to conduct a multi-objective and multi-criteria POE assessment at the building operation stage and provide a multi-criteria optimised solution.

Declaration

I hereby declare that this work has not been submitted in substance for any other degree or award at this or any other university or place of learning, nor is being submitted concurrently in candidature for any degree or other award.

Acknowledgements

First and foremost, I would like to thank my supervisors, Dr QingPing Yang and Dr Rebecca De Coster, for their enduring support, patience and indispensable guidance throughout my study.

I would also like to thank everyone who has provided support and guidance during my research. Additionally, special thanks to Adrian Leaman for his kind help on the case study.

Last but not the least, I would like to thank my family for their unwavering support through this process, in particular, my parents, Shibao Zhao and Yu’e Song, and my aunt, Xueying Liu. Also, my uncle, Shida Zhao, who is sadly no longer with us, I dedicate this thesis to his memory.

Table of Contents

List of Figures

List of Tables

Publications

1. Zhao. Y, and Yang. Q, (2021) 'Development of an ontology-based semantic building post-occupancy evaluation framework', *International Journal of Metrology and Quality Engineering*, 12(19), p. 19. doi: 10.1051/ijmqe/2021019.

2. Zhao, Y. and Yang, Q. (2022) 'A Post-occupancy Evaluation of Occupant Satisfaction in Green and Conventional Higher Educational Buildings', *in 6th International Conference on Environmental Engineering and Sustainable Development (CEESD 2021).* IOP Conference Series: Earth and Environmental Science, 973(1), p. 012010. doi: 10.1088/1755-1315/973/1/012010.

3. Zhao, Y. *et al.* (2020) 'Ontology-based Knowledge Modeling of Post-occupancy Evaluation for Green Building', in *4th International Conference on Environmental and Energy Engineering (IC3E 2020)*. IOP Conference Series: Earth and Environmental Science, p. 012076. doi: 10.1088/1755-1315/495/1/012076.

4. Gao. P, Zhao. Y, Wang. F, and Lu. M, (2020) 'A New Teaching Pattern and Evaluation System of Knowledge Point Puzzle in Massive Online Computer Courses', in *2020 15th International Conference on Computer Science & Education (ICCSE)*. IEEE, pp. 103–106. doi: 10.1109/ICCSE49874.2020.9201775.

5. Fox. A, Zhang. T, Zhao. Y and Yang. Q, (2020) 'A Furniture Industry Concept for a Sustainable Generative Design Platform Employing Robot Based Additive Manufacturing', *ICRSM 2020: International Conference on Rapid and Sustainable Manufacturing*, February 13-14, London, United Kingdom.

Abbreviations

AEC	Architecture, Engineering and Construction
API	Application Programming Interface
AMOD	Agile Methodology for Ontology Development
BAS	Building Automation System
BIM	Building Information Modeling
BMS	Building Management System
BPD	Building Performance Data
BPE	Building Performance Evaluation
BPI	Building Performance Indicators
BPRU	Building Performance Research Unit
BRE	Building Research Establishment
BREEAM	Building Research Establishment Environmental Assessment Method
BSRIA	Building Services Research and Information Association
BUS	Building Use Studies
CI	Confidence Interval
CQIEOntology	Construction Quality Inspection and Evaluation Ontology
CQs	Competency Questions
CSCOntology	Construction Safety Checking Ontology
DNAS	Drivers-Needs-Actions-Systems
eeBIM	Energy Enhanced BIM
HVAC	Heating, Ventilation, and Air Conditioning
HTML	Hypertext Markup Language
IAQ	Indoor Air Quality
IDE	Intelligent Domotic environment
IDEF	Integrated Definition for Function Modeling
IEQ	Indoor Environment Quality
IFC	Industry Foundation Classes
iGBR	intelligent Green Building Rating
IoT	Internet of Things
KE	Knowledge Engineering
KME	Knowledge Management Engineering
LEED	Leadership in Energy and Environment Design
MURBs	Multi-Unit Residential Buildings
NLP	Natural Language Processing
obXML	Occupant Behaviour XML
OB	Occupant Behaviour
OntoSCS	Ontology for Sustainable Concrete Structure
OP	Occupancy Profile
OTKM	On-To-Knowledge Methodology
OWL	Web Ontology Language
OWL DL	OWL Description Logics
POE	Post-Occupancy Evaluation
PROBE	Post-occupancy Review of Buildings and their Engineering
QFD	Quality Function Deployment
RDF	Resource Description Framework
RDFS	RDF Schema

RIF	Rule Interchange Format
RIR	Relative Improvement Ratio
SAREF	Smart Appliance REFerence
SAMOD	Simplified Agile Methodology for Ontology Development
SE	Standard Error
SB	Smart Building
SBT	Sustainable Building Technology
SKME	Simple-Knowledge Engineering Methodology
SOSA	Sensor, Observation, Sample, and Actuator
SPARQL	SPARQL Protocol and RDF Query Language
SPTM	Software Project Time Management
SQWRL	Semantic Query Enhanced Web Rule Language
SRI	Smart Readiness Indicator
SSN	Semantic Sensor Network
SSO	Stimulus Sensor Observation
SW	Semantic Web
SWRL	Semantic Web Rule Language
UML	Unified Modeling Language
UNA	Unique Name Assumption
URL	Uniform Resources Locator
WSN	Wireless Sensor Network
WWW	World Wide Web
W3C	World Wide Web Consortium
XML	Extensible Markup Language

Chapter 1 Introduction

This chapter presents an overview of this research from the perspectives of research background, research problems, research motivations, the main contributions of this research and the structure of this thesis.

1.1 Background

With the growing awareness of sustainable development, the development of sustainable buildings has become a global trend. The objectives of sustainability in building development are to provide a user-centric, environment-friendly, and energy-efficient building to reduce the overall impact of the built environment on the natural environment and human health (Khoshbakht, Gou, Lu, *et al.*, 2018). As a critical economic development pillar of a country, the buildings and construction industry is also one of the biggest energy and resource consumers. According to the United Nations Environment Programme's latest global status report on buildings and construction, the global buildings and construction sector is responsible for 35% of total final energy consumption and about 38% of total energy-related CO_2 emissions (United Nations Environment Programme, 2020).

To cater for sustainable development, some building assessment schemas and various emerging complex technologies, such as IoT (Internet of Things) technology, BIM (Building Information Modeling) technology, or Semantic Web technology, etc., have been developed in different AEC industry fields to facilitate domain knowledge management, facility management, risk management, building performance evaluation and so on. However, even though many emerging smart technologies and building systems have been applied in buildings with the goal of energy-efficient and resource-saving, many buildings do not perform as planned, there is a significant performance gap between the actual and anticipated building performance in the aspects of energy consumption, occupant satisfaction, etc. Some recent studies have suggested that the actual in-use energy consumption of buildings is sometimes up to 5 to 10 times higher than the estimated design intents, which has caused the energy over-consumption issue (ICE, 2021), and the actual average total CO_2 emissions are more than 3 times higher than the average predicts, and the zero-carbon design estimates are hard to achieve in practice (Palmer, Terry and Armitage, 2016). Additionally, since people spend more than 90% of their time indoors, so the performance of the built environment has a significant impact on occupant satisfaction in the aspects of health and productivity (Wu *et al.*, 2007), but some studies show that the average satisfaction level of building users are low as well, even in green or smart technologies equipped buildings.

For many reasons, buildings do not perform as planned. One of the main reasons is the failure of effective knowledge management and information exchange between different systems. It is known that the AEC industry is a knowledge-intensive field (Svetel and Pejanovic, 2010), consisting of many domains. Building data in different domains have their specific formats, and over the decades, the rise of emerging technologies and building evaluation schemes have produced redundant knowledge, the scattered and fragmented knowledge data has made it time-consuming and error-prone to acquire explicit knowledge and information in building management field (Tao, Ota and Dong, 2017).

To solve the above-mentioned problems, some cutting-edge technologies have been used in buildings and construction fields, for example, BIM and Semantic Web technology (Konys, 2018). Benefiting from the advantage of Semantic Web technology in knowledge conceptualization, as the core of Semantic Web, ontology has been widely taken as an effective method for problem-solving, knowledge management, information representation and extraction, and logical inference in the architecture, engineering, and construction domains, especially in building performance evaluation (BPE) field (Corry *et al.*, 2015; Li *et al.*, 2021).

After an extensive review, this study suggests that there is a lack of comprehensive knowledge systematization in the occupants-participated post-occupancy evaluation domain. Additionally, the research of ontology in building performance evaluation is mainly focused on energy-related fields, and there are not enough research in occupant satisfaction related field. Moreover, after a review of the ontology application in the AEC industry, it is found that most of the existing ontologies in the building evaluation domain are lightweight ontologies that are simply based on a hierarchy of concepts and a hierarchy of relations (Fürst and Trichet, 2006). These lightweight ontologies mainly focus on building a structured system to represent the specific domain knowledge or information, without developing formal axioms and constraints to provide higher expressivity.

Since a ‘one-size fits all POE’ does not exist, therefore POEs should be tailored to specific building applications (Hay *et al.*, 2018), and ontology provides a semantic method to represent this specific domain knowledge in a structured and formal way. There is not a comprehensive POE ontology yet, especially with the focus on building occupant satisfaction, which is the starting point of this research. So, a multi-dimensional knowledge-based model of building performance assessment approach is proposed to bridge this gap.

1.2 Research problems statement

- **Problem1: Performance gap between the actual and anticipated building performance**

The global buildings and construction sector is responsible for 35% of total final energy consumption and about 38% of total energy-related CO_2 emissions (United Nations Environment Programme, 2020). However, there is a huge gap between the actual building performance and the design intents in the aspects of energy consumption, carbon emissions, occupant satisfaction, etc.

Some recent studies have found that the actual in-use energy consumption of buildings is sometimes up to 5 to 10 times higher than the estimated design intents, which has caused the energy over-consumption issue (ICE, 2021), and the actual average total CO_2 emissions are more than 3 times higher than the average predicts, and the zero-carbon design estimates are hard to achieve in practice (Palmer, Terry and Armitage, 2016). And the occupant satisfaction is at a low level.

- **Problem 2: The missing of occupant participation in building assessment activities**

Since people spend more than 90% of their time indoors, so the performance of the built environment has a significant impact on occupants' satisfaction in the aspects of health and productivity (Wu *et al.*, 2007).

Occupants are at the heart of building design and energy use. A human-centred building operation is a solution-based approach to optimise the relations between people and buildings. However, designers and developers rarely consult with them or ask their opinions after handover, the occupant satisfaction assessment in POE projects does not get enough attention and support after the building has been built in the past. There is not a comprehensive POE ontology yet, especially with the focus on occupant satisfaction.

- **Problem 3: Knowledge fragmentation and heterogeneous data in AEC industry**

The building assessment process is a knowledge-intensive field, the vast building performance knowledge is mostly documented in various references with different focuses, they are scattered and fragmented, it is time-consuming and error-prone to acquire explicit knowledge and information, especially in ensuring it is continually updated.

- **Problem 4: Limited heavyweight ontology development**

Most of the existing ontologies in the building performance assessment domain are lightweight ontologies, the ontologies that are simply based on a hierarchy of concepts and a hierarchy of relations without axioms interpretation. Lightweight ontology is an ontology or a knowledge organization system with a tree-like structure to simply represent and classify domain knowledge by rather general associations than strict formal logic. The powerful advantages of reasoning and querying capabilities of the Semantic Web are not well developed in these lightweight ontologies.

Different from the lightweight ontology, a heavyweight ontology is an ontology developed with enriched axioms used to fix the semantic interpretation of concepts and relations. Compared to a lightweight ontology, heavyweight ontology provides a higher expressivity, which allows not only obtaining a richer semantic model but also inferring new knowledge from it.

So this research proposes a heavyweight ontology based on the existing ontologies with developed enriched axioms to perform formal reasoning.

1.3 Research motivation

1.3.1 Aim

This study investigates the development of Semantic Web technologies in the buildings and construction field and identifies research gaps of knowledge systematization in the post-occupancy evaluation domain. This research aims to develop a heavyweight ontology dedicated to post-occupancy evaluation (POE) with a focus on occupant satisfaction assessment (OSA), with the ultimate aim of optimizing building operation guidelines and improving occupant use experience quality and well-being. The outcomes of this research can be taken as operation guidelines for future building management systems (BMS) maintenance.

1.3.2 Objectives

To achieve the above-mentioned aim, the research objectives have been explained as below:

- To identify domain knowledge and methodology of Semantic Web technology and its applications in the AEC industry, especially in the building performance evaluation domain;
- To develop a heavyweight OWL ontology with the benchmarking evaluation axioms encoded in SWRL (Semantic Web Rule Language) rules to capture the fragmented knowledge of building assessment in the POE domain, to achieve knowledge share, reuse among the domain experts and the knowledge-based system, and to enable automatic rule-based reasoning and query;
- Based on the ontology model, to integrate different building assessment schemes into the POE activities, like the smart building smart readiness indicator (SRI), to develop a comprehensive POE framework;
- To conduct a field study based upon the Building Use Studies (BUS) Methodology to validate the developed ontology model and to demonstrate the feasibility of Semantic Web applications in the POE domain;
- To promote the participation of building occupants in the building performance assessment process;
- To facilitate the occupant satisfaction assessment of buildings, with the ultimate aim of optimizing building operation guidelines, and improving occupants' use experience quality and well-being.

1.4 Research contribution to knowledge

- **A state-of-the-art review:** This research conducts a state-of-the-art review on the development of post-occupancy evaluation and Semantic Web ontologies, especially the ontology application in the AEC industry, and provides valuable findings.
- **Knowledge reusing and sharing**: On the one hand, this research reuses and extends existing ontologies to systematically present building occupant's needs inside buildings;

on the other hand, the developed ontology in this research is also reusable to provide semantic resources for future relevant research in the building construction industry that achieves knowledge reusing and sharing.

- **Knowledge formalization in POE domain:** Developed a comprehensive knowledge formalization system in the occupants-participated post-occupancy evaluation (POE) domain. In the POE ontology, the building occupants needs for building performance are generalized and classified, and the corresponded building performance assessment knowledge is formalized. The post-occupancy evaluation of buildings is a knowledge-intensive process and consists of different parties, so this research introduces the Semantic Web ontology techniques to represent the POE-related knowledge. The proposed heavyweight ontology model realizes the structural representation, sharing, and reuse of fragmented knowledge and heterogeneous data in the building assessment domain.
- **Semantic Web based multi-objective and multi-criteria assessment:** This research develops a set of SWRL rules and SQWRL queries rules in the proposed ontology framework, which helps to conduct an automatic multi-objective and multi-criteria POE assessment of buildings. And this ontology model also enables effective POE-related knowledge retrieving and sharing, and promotes its integration within the building life cycle assessment.
- **Enhanced POE assessment methodology:** A 'one-size fits all POE' does not exist, therefore this proposed POE framework is tailored to specific building applications with the emphasis on occupant satisfaction in the built environment. A user-centric comprehensive POE framework is developed by reusing the existing building occupants' related ontologies and schemes into this POE to realize the interoperability and reusing of different schemes.
- **Multi-objective optimisation:** This research also presents a real use case study that demonstrates how the proposed ontology can be used to infer explicit and implicit assessment knowledge to facilitate the occupant satisfaction assessment of buildings, with the ultimate aim of optimizing building operation guidelines, and improving occupants' use experience quality and well-being.

1.5 Thesis structure

As shown in Figure 1-1, there are six chapters in this thesis, the content of each chapter is introduced below.

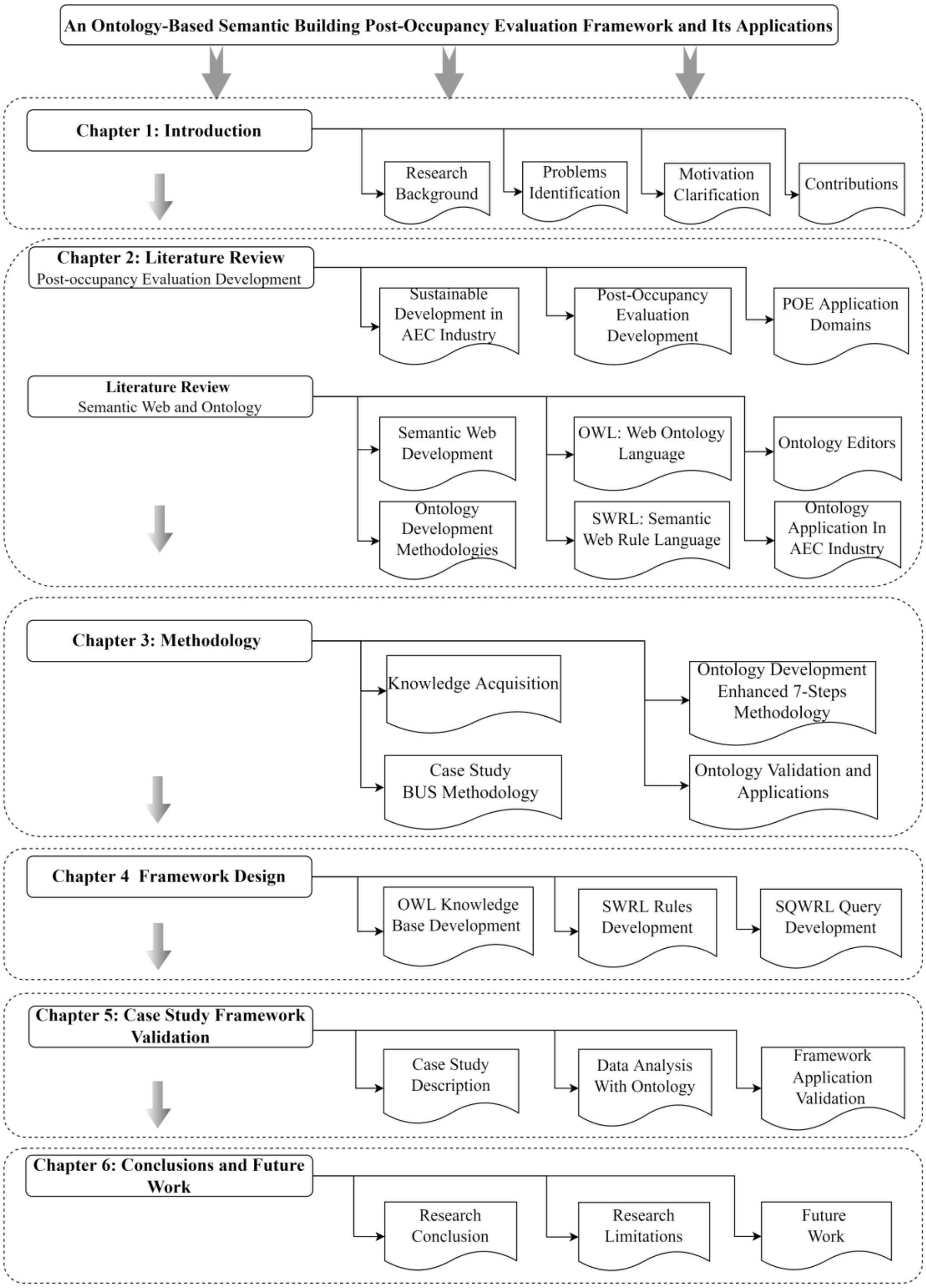


Figure 1-1 Thesis structure

Chapter 1 presents an overview of this research. The background of this research is given first in Section 1.1; Section 1.2 states the research problems from a few key aspects; the research motivation is explained with the aim and objectives in Section 1.3; Section 1.4 lists the main contributions of this research; the structure of this thesis is illustrated in Section1.5.

Chapter 2 firstly briefly reviews the various sustainable building schemas, such as the BREEAM, LEED, WELL, etc. Then the development of post-occupancy evaluation (POE) in the building sector has been discussed in Section 2.2, Section 2.3 and Section 2.4, and the research gaps have been identified as well. The second part of the literature review discussed the development of Semantic Web and ontology from the aspects of ontology definition, ontology languages, ontology editors, ontology development methodologies and ontology applications in different domains, etc. The Semantic Web technology is investigated in Section 2.5; ontology as the key technology of Semantic Web has been illustrated in Section 2.6 from the perspectives of the ontology definition, ontology languages, ontology query technology, ontology editors, and so on; the methodologies for building ontologies are compared in Section 2.7, after analysing the advantages and disadvantages of each method, the '7-Steps' methodology from the Noy and McGuinness (2001) is adopted in this research; in Section 2.8, the application of ontologies in different fields, especially the AEC industry domain, is reviewed in detail, the limitations of current studies are given as well; Section 2.9 is the summary of the development of Semantic Web and ontology.

Chapter 3 introduces the research methodology of this study. The overview of the used methodologies has been illustrated in Section 3.1; the main OWL ontology knowledge base development methods have been explained in Section 3.2; the POE-related SWRL rules and SQWRL query development methods have been explained in Section 3.3 and Section 3.4; Section 3.5 is the summary of this chapter.

Chapter 4 presents the framework design and development. By following the methodology adopted from Chapter 3, the procedures of developing the POE ontology are illustrated step by step within software Protégé 5.5.0. The overview of the proposed ontology-based post-occupancy evaluation framework has been illustrated in Section 4.1; the OWL ontology knowledge base structure processes have been explained step by step in Section 4.2; in Section 4.3, the POE related rules are edited into SWRL rules to support the ontology reasoning in this research; the SQWRL queries are edited in Section 4.4; the ontology reasoning and querying examples are given in Section 4.5; the chapter summary is given in Section 4.6.

Chapter 5 conducts a validation of the proposed framework using a POE case study with a focus on occupant satisfaction assessment in buildings. Section 5.1 describes the case study in detail, including the licensed Building Use Studies (BUS) Methodology and the assessed sample buildings; the data collection and analysis and integration with ontology are presented in Section 5.2; Section 5.3 presents the ontology application scenarios and framework validation; Section 5.4 is the summary of this chapter.

Chapter 6 is the conclusion part, which briefly summarized this research. Research problems are reviewed in Section 6.1; major research findings are declared in Section 6.2; research methodology is reviewed in Section 6.3; the contributions to the knowledge are recalled in Section 6.4; the limitations of this research and the potential future research trend are given in Section 6.5.

Chapter 2 Literature review

This chapter presents a state-of-the-art literature review of post-occupancy evaluation (POE), Semantic Web and Ontology. Section 2.1 starts with a brief introduction to the development of sustainability concepts in the AEC industry sector, and a few outstanding building evaluation systems have been introduced as well; the building performance gaps have been identified in Section 2.2; Section 2.3 presents the development of post-occupancy evaluation (POE), and the occupant satisfaction assessment in different building-related domains have been reviewed in the section; Section 2.4 summaries the literature review on the development of POE.

The introduction of Semantic Web development is given in Section 2.5, then the ontology technologies development and relevant research are discussed in Section 2.6, including the Web Ontology Language (OWL), Semantic Web Rule Language (SWRL), ontology query languages formats, ontology editors, and so on; the methodology for developing ontology is reviewed in Section 2.7 the application of ontology technologies in the AEC industry is discussed in Section 2.8, especially the application in the knowledge management domain; Section 2.9 summaries of the literature review on the development of Semantic Web and ontology.

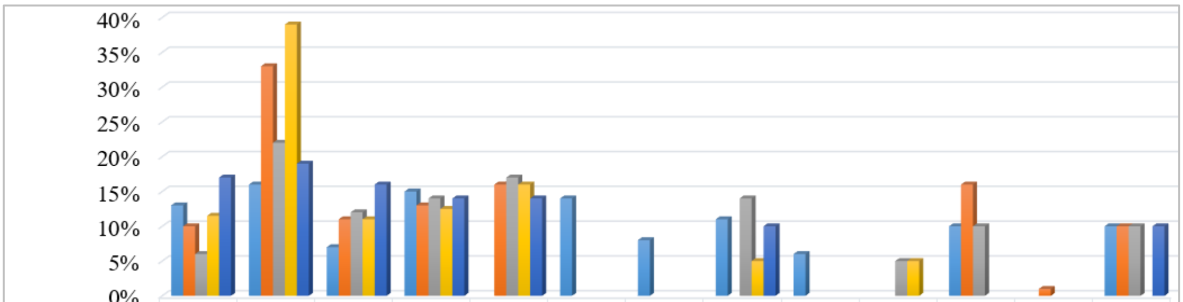
2.1 Sustainable development in the AEC industry

The increasingly severe environmental problems, energy over-consumption issues, and the shortage of natural resources have become the main restriction factors for the sustainable development of the environment, society and global economy. Under the consideration of long-term development, the concept of sustainable development was put forward in World Commission on Environment and Development by Brundtland Commission in 1987 (Liu, Low and He, 2012), which refers to “meet the needs of the present generation without compromising the ability of future generations to meet their own needs”. This concept has been widely accepted in different areas of development, for example, business development, agriculture, industry sector, architecture, engineering and construction industry and so on.

As a critical economic development pillar of a country, the buildings and construction industry also is one of the biggest energy and resource consumers. According to the United Nations Environment Programme’ latest global status report for buildings and construction, the global buildings and construction sector is responsible for 35% of total final energy consumption and about 38% of total energy-related CO_2 emissions (United Nations Environment Programme, 2020). The growing awareness of sustainable development has promoted the development and implementation of green buildings in the modern construction industry worldwide (Darko and Chan, 2016). The objectives of sustainability in building development are to provide a user-centric, environment-friendly, and energy-efficient building to reduce the overall impact of the built environment on the natural environment and human health (Ding *et al.*, 2018; Khoshbakht, Gou, Lu, *et al.*, 2018).

Cater to the development needs of sustainable buildings, many methods have been developed to guide the development of buildings and improve buildings performance as well as ensure occupants' well-being and productivity (Stevenson, 2009), for example, some building assessment schemes, such as LEED (Leadership in Energy and Environment Design), BREEAM (Building Research Establishment Environmental Assessment Method) (Awadh, 2017; Bernardi *et al.*, 2017; Ding *et al.*, 2018), Green Star, WELL Building Standard (IWBL, 2016), and various smart technologies like BMS systems, BIM technology, Semantic Web technology, SRI (Ma and Verbeke, 2021), etc.

The current building assessment schemes, for example, the BREEAM (Building Research Establishment Environmental Assessment Method) developed by BRE (Building Research Establishment) in 1990 (Awadh, 2017; Behm and Pearce, 2017; Doan *et al.*, 2017), and the most widely applied LEED (Leadership in Energy and Environmental Design) launched by USGBC (United States of Green Building Council) in 1994, are both developed to help to set standards for encouraging continuous sustainable performance improvement and innovation during the whole building life cycle from planning, design, construction, operation and maintenance stage to refurbishment (BRE Gloabl, 2018). As shown in Figure 2-1, these sustainable building assessment standards are developed with different focus categories, in which, the energy, site, water, material, and indoor environment are the most general categories, and with the highest importance on energy efficiency (Shan and Hwang, 2018).



	Site	Energy	Water	Material	IEQ	Health and Well-being	Pollution	Management	Waste	Emission	Transport	Integrated Process	Innovation
BREEAM	13%	16%	7%	15%		14%	8%	11%	6%		10%		10%
LEED	10%	33%	11%	13%	16%						16%	1%	10%
Green Star	6%	22%	12%	14%	17%			14%		5%	10%		10%
Green Globes	11.50%	39%	11%	12.50%	16%			5%		5%			
ESGB	17%	19%	16%	14%	14%			10%					10%

BREEAM LEED Green Star Green Globes ESGB

Figure 2-1 The categories of different building assessment schemes

Since these systems do not specify the stage to be assessed, most buildings focus on the assessment in the design stage, but not the operation stage (Ding et al., 2018). This is a common issue with other building evaluation systems, as a result, there is an unbalanced building evaluation development between the design stage and operation stage (Zhou, 2014). Moreover, the building designers and developers are no longer responsible for the operation and maintenance of buildings after the handover, and they rarely consult with building occupants to get their feedback after the handover. The buildings are serviced for occupants, and the occupants' activities or behaviours are the main influence part of building performance (Delzendeh *et al.*, 2017; Vigna *et al.*, 2020; Mahdavi *et al.*, 2021). Meeting the occupant's requirement is the basic as well as the crucial target for the success of any buildings, let alone the smart or sustainable buildings (Fekry, Zafarany and Shamseldin, 2014), as the occupants' feedback is useful for highlighting design problems, especially those relating to controls and BMSs.

Since people spend more than 90% of their time indoors, the built environment of buildings has a significant impact on occupants' health and productivity (Ning and Chen, 2016). Compared with the research focused on sustainable buildings in the aspects of energy-saving and resource efficiency, however, the occupant satisfaction related research has not received the same emphasis. The development of buildings should not only include environmentally responsible and resource-efficient building concepts but also integrate human well-being and performance.

For many reasons, many buildings do not perform as planned. For example, a lack of monitoring and feedback following occupancy, user behaviours are not corrected and lessons are not learned for future projects, the specification uncertainty in building modelling and poor practice in operation, and so on (Preiser and Vischer, 2005; Stevenson, 2009, 2018; Bordass and Leaman, 2015; Hassin and Azlani, 2018). A large number of studies have been conducted to identify the building performance gaps and the probably caused reasons (Scofield, 2009; Zero Carbon Hub, 2013; De Wilde, 2014; Palmer, Terry and Armitage, 2016; van Dronkelaar *et al.*, 2016; Taylor *et al.*, 2018; ICE, 2021). In 2016, another U.K. government-funded project conducted a building assessment survey of 56 buildings, the results showed that even the BREEAM certificated office buildings were not performing as they should do, the buildings consumed up to 3 to 10 times the energy they should, and occupants had lower satisfaction compared with the non-certificated buildings (Palmer, Terry and Armitage, 2016; Stevenson, 2018). In addition, many existing buildings could not respond to occupants' requirements towards satisfied IEQ of buildings (David Jiboye, 2012; Gou, Prasad and Siu-Yu Lau, 2013; Thatcher and Milner, 2016; Khoshbakht, Gou, Lu, *et al.*, 2018; Gilani, Quinn and McArthur, 2020).

To better understand the buildings' performance as well as the well-being and productivity of occupants, a comprehensive building performance evaluation (BPE) diagnostic method, Post-Occupancy Evaluation (POE) have been developed in the 1980s to narrow down the performance gap existing in energy efficiency, occupants satisfaction and so on (Preiser and Vischer, 2005).

2.2 Post-occupancy evaluation (POE)

The gap between actual operating performance and design intent performance has highlighted the need for the development of post-assessment methods. One of the barriers preventing the implementation of building performance evaluation is the missing of a more precise assessment method of buildings occupants' experience of environmental comfort rather than using the conventional methods (Preiser and Vischer, 2005).

Building Performance Evaluation (BPE) has its origins in England, from the early work in 1965 by Manning of the Pilkington Research Unit (Durosaiye, Hadjri and Liyanage, 2019), and the work by Markus et al. (1972) of the Building Performance Research Unit (BPRU) at the University of Strathclyde in Scotland (Preiser and Schramm, 2002).

As shown in Figure 2-2, Preiser and Vischer (2005) describe the Building Performance Evaluation (BPE) as a systematic process to evaluate the building performance and ensure the feedback is available through the whole building life cycle from the stages of strategic planning, design, construction, occupation, and operation. BSRIA (Building Services Research and Information Association), an ISO 9001 qualified construction and building service organization in the UK, describes the BPE as a method to evaluate energy performance, occupant comfort and compares the performance feedback with initial design goals at any building life's stages of planning, programming, design, construction, occupancy, recycling.

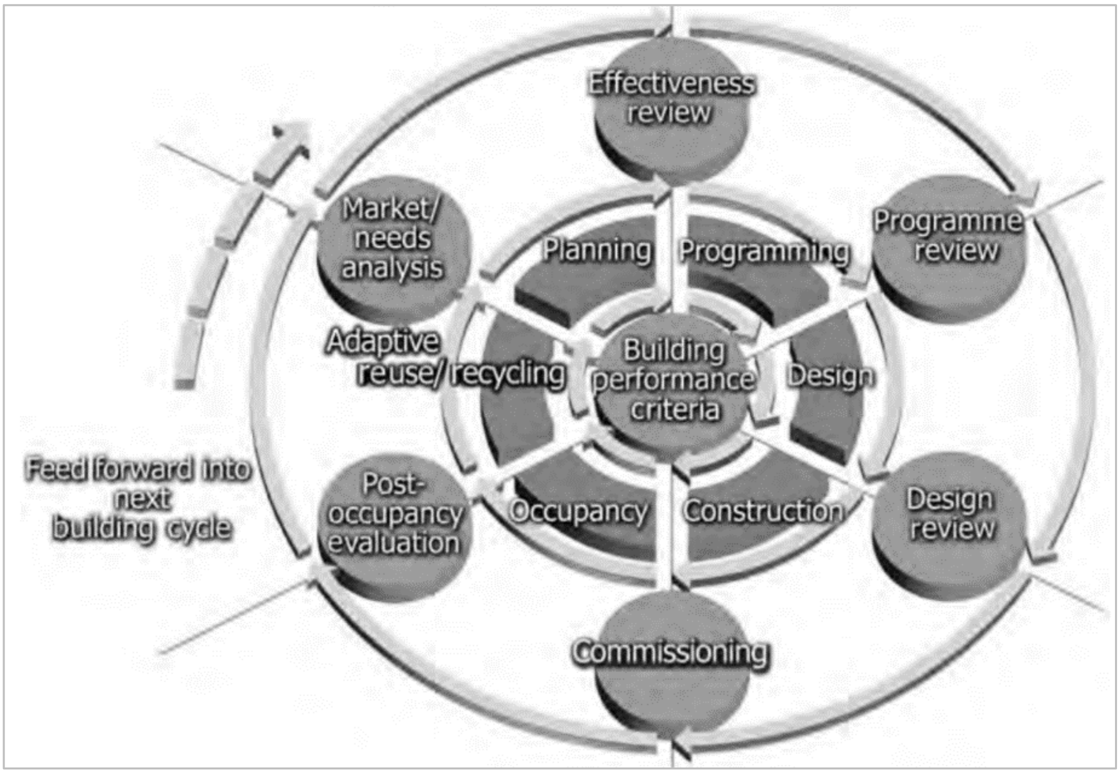


Figure 2-2 Building performance evaluation (BPE) model
(Source: (Preiser and Vischer, 2005))

In 1997, a post-occupancy evaluation model was developed into an integrative framework for BPE (Preiser and Schramm, 1997). Post-occupancy evaluation (POE) is viewed as a sub-process of the BPE method to evaluate the actual building performance against the theoretical design intents after the building has been occupied for some time, to understand how the building is performing, and to capture lessons learned (Preiser and Vischer, 2005; Mastor and Ibrahim, 2010). The POE offers an opportunity to investigate the buildings' actual performance based upon the occupants' satisfaction levels in the aspects of building overall design, indoor environmental quality, thermal comfort, etc. (Ning and Chen, 2016).

POE originated in the United States and has been used since the 1960s. Zimring and Reizenstein (1980) defined POE as 'an investigation of the designed environment with regard to its human users'.

The definitions of POE by Preiser in different periods are as follows:

- "Post-occupancy evaluation is the process of systematically comparing actual building performance, i.e., performance measures, with explicitly stated performance criteria." (Preiser, 1995)
- "POE can be defined as the act of evaluating buildings in a systematic and rigorous manner after they have been built and occupied for some time." (Preiser and Vischer, 2005)

In contrast with other building evaluation systems, which place more consideration on energy and materials performance or aesthetics of buildings, POE promotes the participation of building occupants, the end-users, focuses on their requirements of buildings in the aspects of health, safety, convenience, amenity, psychological comfort, living quality and satisfaction, it emphasizes on building occupants' needs (Zimmerman and Martin, 2001; Preiser, 2002; Preiser and Vischer, 2005; Pereira, Rodrigues and Rocha, 2016).

In the British Standards of "Briefing for design and construction –Part 1: Code of practice for facilities management (Buildings infrastructure)" (BSI, 2015), POE is defined as "Process of evaluating an asset/facility after it has been completed and is in use to understand its actual performance against that required and to capture lessons learned." In another UK standard of "Technical Manual SD5078: BREEAM UK New Construction 2018", POE has been developed as one of the assessment criteria with one credit under the indicator of 'Maintenance 05 Aftercare', to incent the application of POE exercise one year after the building been occupied in the aspects of environmental conditions, facilities and amenities, control, operation and maintenance, energy and water consumption condition and so on (BRE Global Ltd, 2018).

As described by the BRE Group, POE offers a way to provide performance feedback throughout the whole lifecycle of buildings, from the initial concept design, construction, operation and maintenance to occupation. And the feedback for the POE can help to provide development guidance for future similar projects. It can not only provide feedback on buildings' performance but also produce a positive impact on running costs, occupants' well-being and business efficiency.

There are various approaches to POE studies, which have been classified according to the intensity of the investigation (Vischer, 2001). The degree and extent of POE studies primarily depend on the necessity and purpose of the POE to meet the short, medium or long term benefits and the availability of funds (Mastor and Ibrahim, 2010).

There are 3 levels of effort in the POE system, and every level is developed into the same 3 phases and different 3 steps (Preiser, 1995; Wang, 2011; Fronczek-Munter, 2013), the structure of POE is shown in the following Figure 2-3.

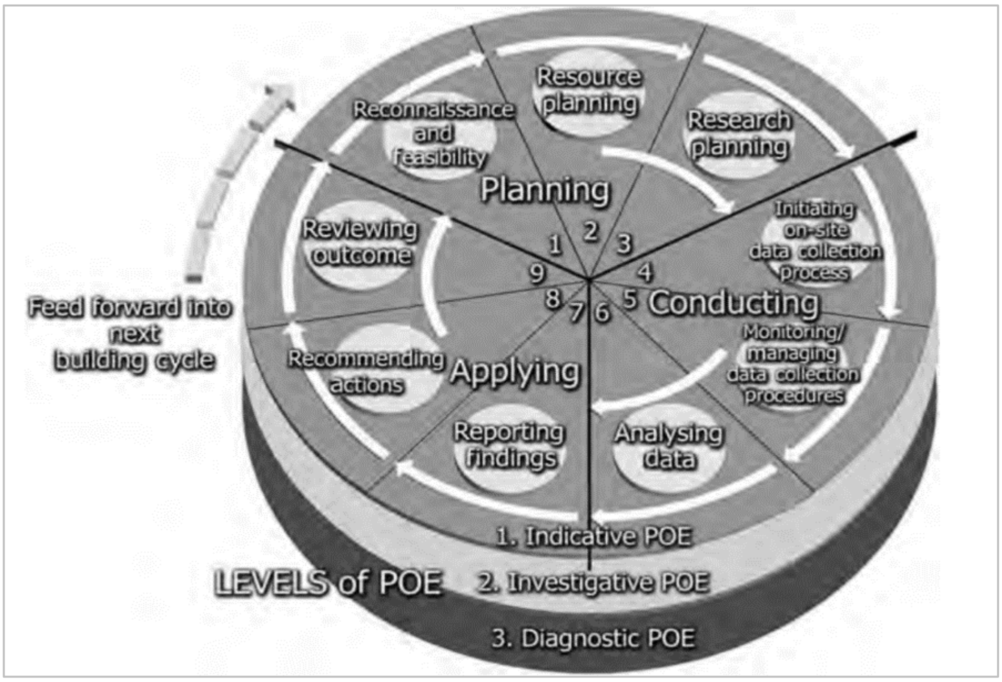


Figure 2-3 POE framework (Source:(Preiser and Vischer, 2005))

Level 1: Indicative -- Short-term Evaluation

Indicative POE is a quick, general inspection of building performance to provide timely theoretical and technical improvement guidance for organizations through assessing the building performance and identifying the positive impacts and key problems of buildings within a few hours. The methods used in this evaluation include structured interviews with experienced personnel, the occupants involved in a group meeting, on-site inspections, etc (Preiser, 1995).

Level 2: Investigative--Medium-term Evaluation

Investigative POE is a more in-depth evaluation of building performance, which is used to lead a more detailed evaluation on building performance to provide mode detailed improvement suggestions for designers, projects owners, end-users and other relevant organizations. It stylizes interview and surveys questionnaires, photographic/video recordings, and physical measurements. In addition, it takes a longer time than the first level, from a week to a few months, which depends on the investigation depth and the amount of personnel involved (Preiser, 1995).

Level 3: Diagnostic-- Long-term Evaluation

This level is a longitudinal and cross-sectional evaluation of building performance, including a series of complicated and comprehensive data collection and analysis in the aspects of facilities safety, privacy, physical measurements, as well as surveys and interviews of stakeholders. Diagnostic POE not only provides suggestions on improving building performance, but also provides theoretical support for the improvement of existing standards, criteria, and guidance literature. It is a long-term evaluation activity, it can take months or years (Preiser, 1995; Turpin-Brooks and Viccars, 2006).

Every level of the POE system has 3 phases, i.e. Planning, Conducting, Applying, and every phase has 3 different steps (Preiser, 1995). Every step is expressed in Figure 2-4, each step has a different assessment aim, tasks, actions (work plan), resources and results, the details of these procedures will be discussed in the methodology part.

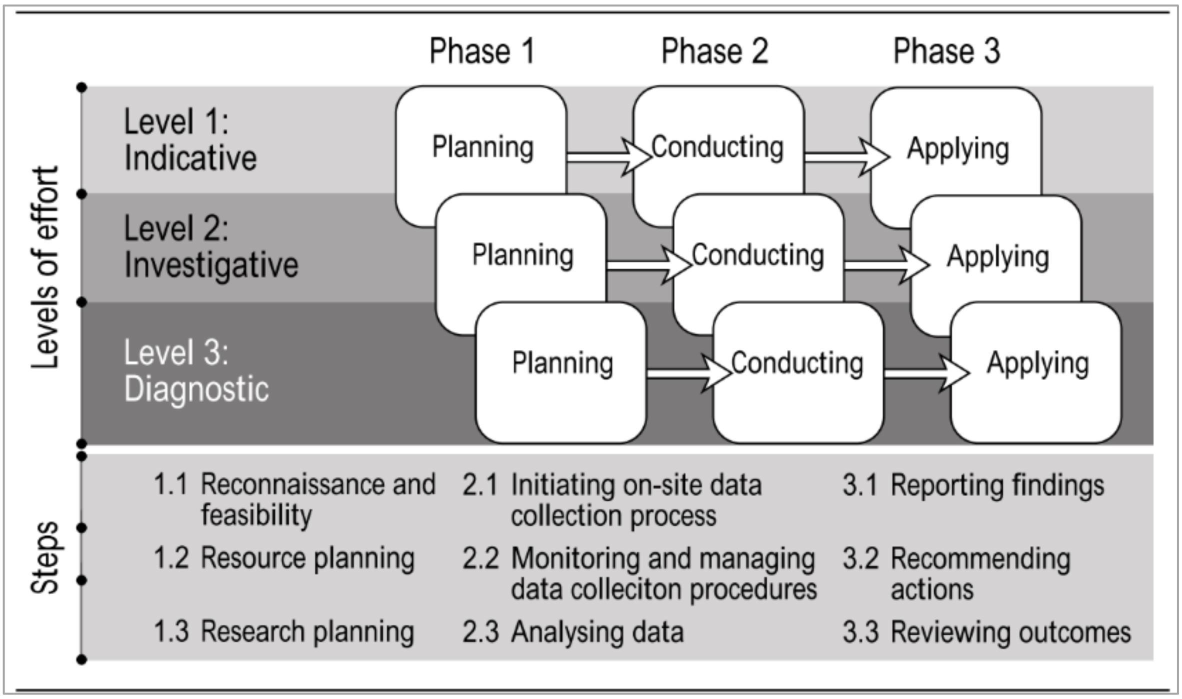


Figure 2-4 Post-Occupancy Evaluation Process Model (Source:(Preiser, 1995))

2.3 Applications of POE in different fields

The value of the POE implementation is being increasingly recognized and hundreds of POEs have been applied in various fields, when it comes to building fields, it can be used in facility management tools (Preiser, 1995), as well as in different building types, such as healthcare (Carthey, 2006; Fronczek-Munter, 2013), (Wang, 2011; Alborz and Berardi, 2015; Hermawati *et al.*, 2015), green buildings (Watson, 2007), offices (Kishnani et al., 2014), commercial buildings and residential housing (DARKWA, 2006; Teasdale-St-Hilaire, 2013; Khair et al., 2015), university dormitories (Disterheft et al., 2015; Ning and Chen, 2016) and so on.

Preiser (1995) developed POE as a facility management tool for the facility managers to use for total quality management. POE technologies can identify the performance problems in occupied facilities and provide recommendations to improve and optimise the facility performance in the continuous quest for quality improvement. This model is also used to optimise the university facility management. Preiser’s POE facility management tool is shown below.

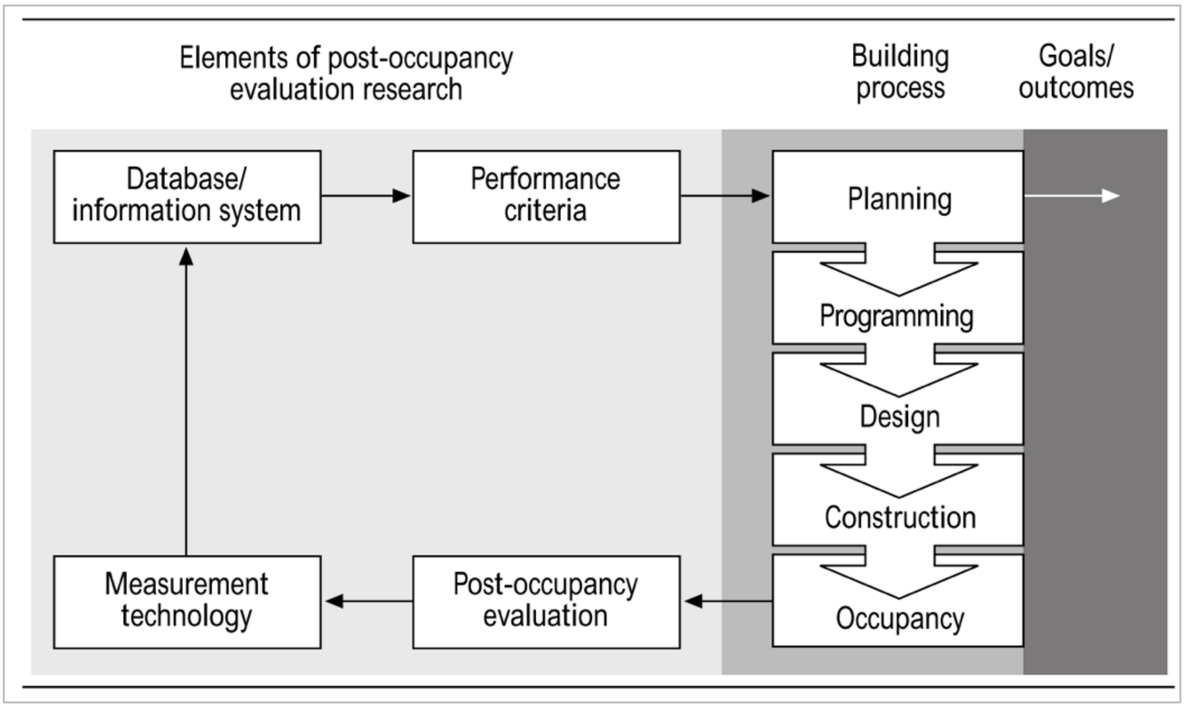


Figure 2-5 POE as a Facility Management Tool (Source: (Preiser, 1995))

In 1995, a U.K. government-led Post-occupancy Review of Buildings and their Engineering (PROBE) project, started the first systematic performance assessment among green and conventional office buildings in the U.K., the project result found that, as a group, the green buildings performed better than conventional buildings, however, the occupants had different satisfactions attitudes on their building (Bordass *et al.*, 2001b, 2001a; Bordass, Leaman and Ruyssevelt, 2001; Leaman and Bordass, 2001; Robert Cohen *et al.*, 2001).

Carthey (2006) proposed a standardized POE methodology for Australian health projects, named "Australasian Post Occupancy Evaluation Methodology", as shown in Figure 2-6, which has been agreed that this methodology would be adopted to evaluate all health capital projects across Australia. In response to this research, POE is defined as "the systematic evaluation of health service buildings or facilities" in this research.

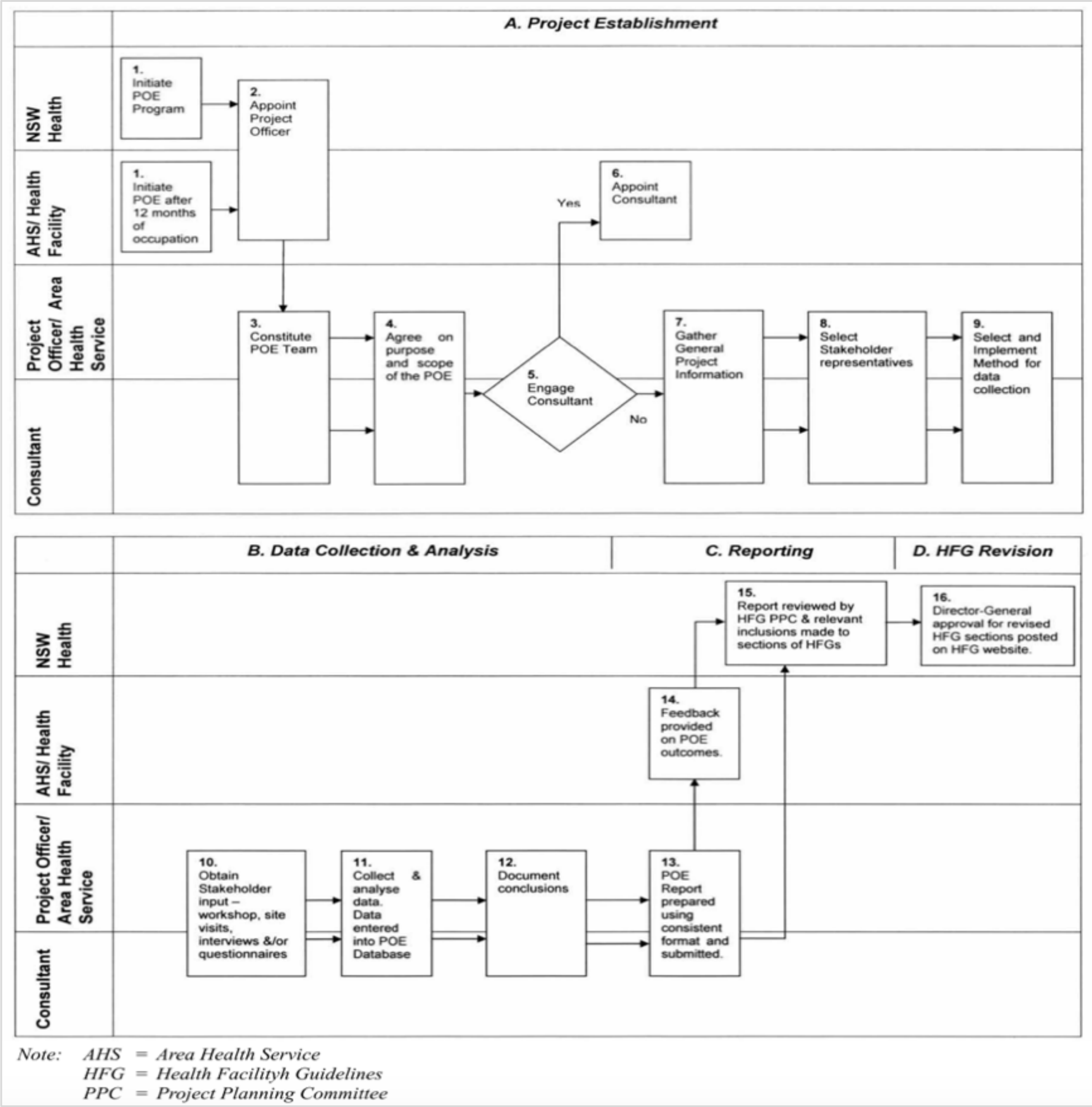


Figure 2-6 Process for Conduct of NSW Health POE(Source: (Carthey, 2006))

Kishnani et al. (2014) conducted a series of post-occupancy evaluations among 11 office buildings in Singapore to evaluate the effect of the national green building evaluation tool, i.e. Green Mark. By analyzing the performance feedback by POE method between 8 Green Mark-labelled offices and 3 conventional office buildings in different aspects of energy use, indoor environment quality and occupant well-being, the result of this research has pointed out that there was a large observed variance in occupant density and non-compliance in temperature specifications operating. At last, this research discussed how Green Mark might become a policy measure in the future, and also proposed some improvement suggestions to deal with the problems mentioned above.

Yu *et al* (2015; 2016) have developed eight first-level indicators and 149 indexes to establish the POE-SGREP index system, i.e. Post Occupancy Evaluation System of Green Residential Environmental Performance, which is applied to produce a case study of a green residential building located in the cold region in China. The results have shown that the indoor environment and outdoor environment satisfy the design requirements and the whole project environment makes customers satisfied.

A POE application research carried out by Dundar (2016) in three different school buildings, analysed the spatial compositional issues of designing school buildings. The results contributed to improving the quality of school facilities in the future design work in consideration of facilities' location, size, proportion etc. Hermawati et al. (2015) introduced an IDEF (Integrated Definition for Function Modeling) framework model to develop an indicative POE to evaluate the quality of the designed built-environment of school facilities and the research result has provided the applicability of this framework for evaluating school facilities.

Abdou and Dghaimat (2016) had a POE for a private school in the UAE, and a proposed POE criterion based on technical evaluation and occupant's perception measurement was presented. The outcomes of this research helped to increase the awareness of POE application in the facility management process and provide the design experiences for improving the current and future design of education buildings in the UAE. The proposed POE Methodology for School Facility is shown in Figure 2-7.

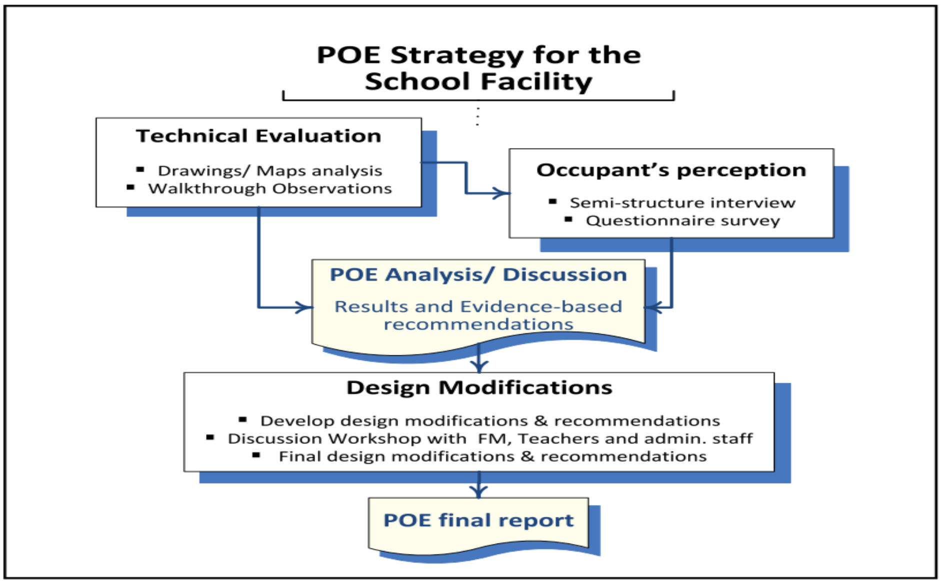


Figure 2-7 The POE Methodology for School Facility by Abdou and Dghaimat (2016)

As shown in Figure 2-8, Teasdale-St-Hilaire (2013) had an overview of the existing POE protocol within different building types and found that very little information is available in the public domain.

POE Protocol	Target Building Type
EcoSmart (EcoSmart Foundation 2007)	MURBs
Keen Engineering POE protocol (Keen Engineering 2005)	Office buildings
Watson (Watson 1996, 2003); Watson and Fitzgerald (1998)	Various
Higher Education Funding Council for England (HEFCE 2006)	Higher education buildings
Baird (Baird 2010)	Sustainable commercial and institutional buildings
Post-Occupancy Review of Buildings and their Engineering, or PROBE (Cohen et al. 1999; Leaman, Stevenson, and Boyce)	Offices, institutional industrial and government buildings
Soft Landings Framework (BSRIA 2009)	Non-specific
Building Research Establishment (BRE)'s Design Quality Method (Cook 2007)	Various, including schools, hospitals, and housing
Sanders (2010)	Housing
Birt and Newsham (2009)	Commercial and office buildings

Figure 2-8 POE Protocol and target building type (Source: (Teasdale-St-Hilaire, 2013))

Teasdale-St-Hilaire (2013) mentioned that the post-occupancy evaluation has focused on non-residential buildings, and there was a lack of multi-unit residential buildings evaluation, to fill the gap, he presented a POE methodology for multi-unit residential buildings (MURBs) based on seven performance indicators i.e. energy and water use efficiency, indoor air quality, lighting and the visual environment, acoustics, thermal comfort, and building envelope performance. The framework of POE methodology for MURBs is shown in Figure 2-9, the survey was divided into 4 steps to develop a conduct an evaluation feedback loop. The research result can provide feedback for the stakeholders to improve the building performance, what's more, the outcome can help to develop building codes, regulations, and guidelines for the building industry. In addition, this research also considered the building envelope performance as a performance indicator in this evaluation framework.

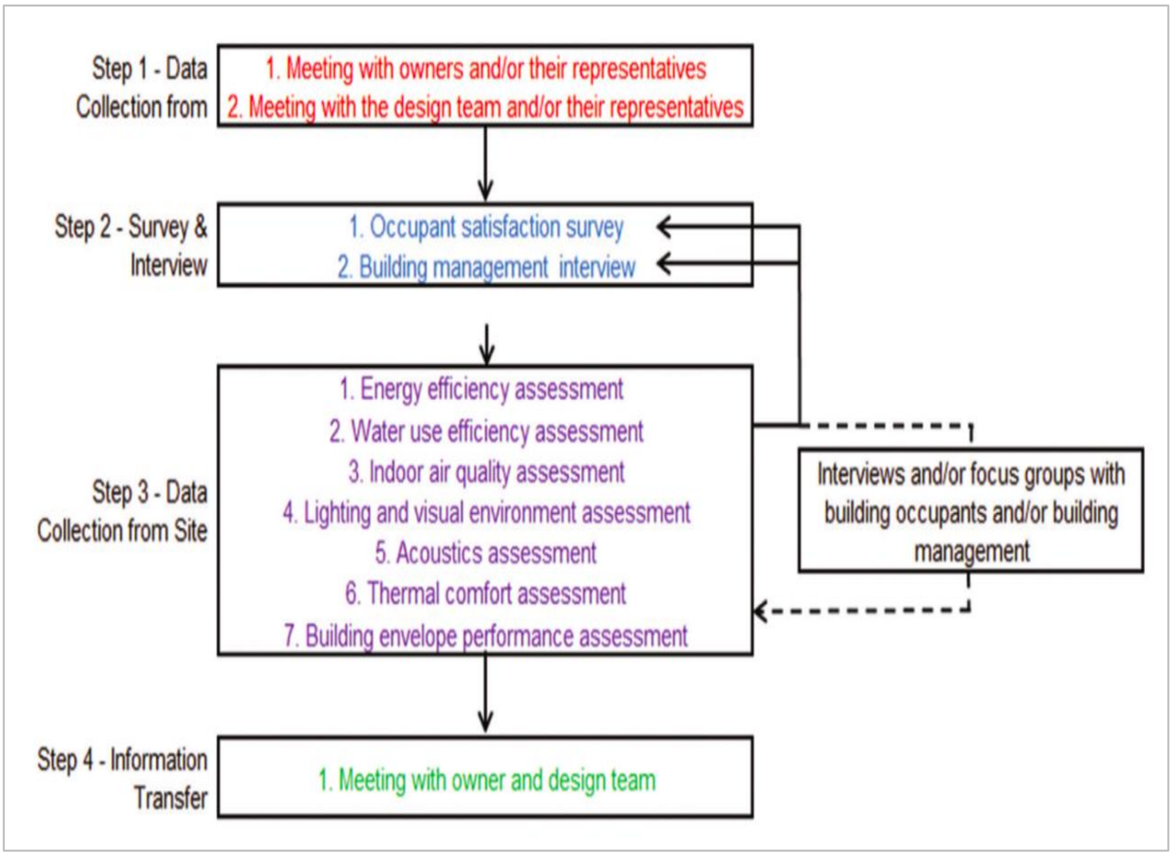


Figure 2-9 Framework of POE Methodology for MURBs
(Source: (Teasdale-St-Hilaire, 2013))

Khair et al.(2015) conducted a study by using POE tools to determine the physical environment elements of public low-cost housings based on occupants' preferences in Malaysia. This study used the convenience sampling technique to collect data and the result indicated that the physical environment elements were critical for these low-cost housings, especially the dwelling unit features, housing conditions, location and safety, etc.

Khoshbakht, Gou, Lu, et al. (2018) had a POE study with a focus on occupants satisfaction between sustainable-certificated and conventional office buildings in different counties, the research found out that, in occident countries such as U.S.and U.K., there were no significant differences in occupants' satisfaction in almost all IEQ aspects between these two types of buildings. However, in oriental countries, like China and South Korea, occupants in sustainable/green buildings had significantly higher satisfaction than those in non-green buildings.

As of now, the POE activities have covered many building assessment categories, for example, thermal comfort, IEQ, lighting, ventilation, acoustics, building overall design, controlling, etc, which offers an opportunity to investigate the buildings' actual performance based upon the occupants' feedback in these aspects.

2.3.1 Benefits of POE implementation

As a building performance evaluation tool, the POE system has many benefits, some of which are listed below:

- Helpful for energy, water and materials saving, pollution, climate and environmental issues improvement, occupant satisfaction improvement, etc. (Preiser, 1995; Zimmerman and Martin, 2001; J Woo, 2017; Yu, Ma and Chen, 2017);
- Help to build the communication link between different stakeholders, and building systems, for example, the communication between the occupants and designer or developers (Alborzfard and Berardi, 2013);
- It probes outcomes and makes recommendations to provide the reference data and knowledge for future similar building projects and not repeat failures (Brown *et al.*, 2010; Mastor and Ibrahim, 2010; Khoshbakht, Gou, Lu, *et al.*, 2018)
- Helpful to decision making during the design stage of building projects (Zimmerman and Martin, 2001; Hiromoto, 2015);
- As a way to develop benchmark data on building performance criteria to support the development of building codes, regulations, standards and guidelines (Kujawski, 2013; Teasdale-St-Hilaire, 2013; Brioso et al., 2015), and so on.

2.3.2 Barriers to POE development

Besides the benefits of POE, various barriers preventing the widespread implementation of post-occupancy evaluation are identified.

- A 'one-size fits all POE' does not exist, for different building types or building systems, the POE assessment targets are different, therefore, the POE needs to be tailored according to the characteristics of specific building applications (Leaman, Stevenson and Bordass, 2010; Alborzfard and Berardi, 2013; Guinther, Carll-White and Real, 2014; Hay *et al.*, 2018; Li, Froese and Brager, 2018);

- The buildings and construction industry is full of fragmented information and knowledge, the fragmentation has hindered the POEs development (Abanda and Tah, 2008; Nawawi and Khalil, 2008; Meir *et al.*, 2009; Haron and Khairudin, 2012; Bordass and Leaman, 2015; UKGBC, 2016; Jiang, Wang and Wu, 2018; Edirisinghe and Woo, 2021);
- The variety of POE research foci and methodologies have generated fragmented body knowledge systems, which has resulted in the AEC industry becoming a knowledge-intensive industry (Alborzfard and Berardi, 2013; Zhang, Boukamp and Teizer, 2015; Thatcher and Milner, 2016; Bernardi *et al.*, 2017);
- Facing the challenges of choosing the appropriate assessment indicators and techniques among the vast and scattered knowledge has caused difficulties to promote POE implementation (Alborzfard and Berardi, 2013; Zhang, Boukamp and Teizer, 2015; Thatcher and Milner, 2016; Bernardi *et al.*, 2017; Zhao and Yang, 2021);

2.4 Review summary of post-occupancy evaluation development

As analysed in previous sections, for many reasons, buildings do not perform as planned. To achieve better building performance, post-occupancy evaluation (POE), as a user-centric building assessment method, has been used to evaluate the actual building performance against the theoretical design intents after the building has been occupied for some time. However, many factors have hindered the POE development, one of the main factors is the failure of effective knowledge management and information exchange in the POE domain when facing fragmented knowledge and redundant information of buildings. It is known that the AEC industry is a knowledge-intensive field, over the decades, the rise of emerging technologies and building evaluation schemes have produced a large amount of knowledge and data, but the scattered and fragmented knowledge has made it time-consuming and error-prone to acquire explicit knowledge and information in building management field, and that has slowed the POE development as well.

The Semantic Web provides an environment where the data can be processed automatically by machines, shared and exchanged between different applications, agents, and a community of users (Berners-Lee, Hendler and Lassila, 2001; Bikakis *et al.*, 2013). As the key technique of the Semantic Web, ontology annotates semantics and provides a common, comprehensible foundation for data information on the Semantic Web. Moreover, ontology can provide a common vocabulary, a grammar for processing data, and can supply a semantic description of data that can be used for semantic inference (Taye, 2010).

To resolve the above-analysed POE development obstacles, the Semantic Web technology, ontology has been introduced into this research to develop a comprehensive structured POE framework, to capture the fragmented building assessment knowledge in the POE domain, and also to achieve knowledge share, reuse among the domain experts and the knowledge-based building performance assessment systems and to enable automatic rule-based reasoning and query.

The Semantic Web, ontology, and their application in the AEC industry are introduced in detail in the following sections.

2.5 Semantic Web

The Semantic Web coined by Tim Berners-Lee has been regarded as "an extension of the World Wide Web (WWW) but not a separate Web, in which information is given well-defined meaning, better enabling computers and people to work in cooperation" (Berners-Lee, Hendler and Lassila, 2001). It is developed based on the standards set by the international standardisation consortium for the World Wide Web, the World Wide Web Consortium (W3C), the W3C describes the Semantic Web as: ''The Semantic Web is a vision for the future of the Web in which information is given explicit meaning, making it easier for machines to automatically process and integrate information available on the Web ''. The aim to develop the Semantic Web is to make the Web data machine-understandable, Tim Berners-Lee has pointed out that the development of the Semantic Web is not only to make the Web suitable for information sharing and collaboration with each other but also to make the Web understandable and processable by machines (Berners-Lee, Hendler and Lassila, 2001).

Data in the Semantic Web is given well-defined meaning in machine-readable formats, which enables machines to understand, and automatically process, integrate and interpret data information. Compared to the current Web, the Semantic Web makes information retrieval and extraction more accurate, and enhances interoperability between humans and computers (Zhao and Yang, 2021). As an approach to incorporating knowledge modelling into the Web, the Semantic Web introduces semantic technologies into the formal modelling and knowledge representation domain (Hitzler, Krötzsch and Rudolph, 2010). To achieve the above-mentioned functionalities of the Semantic Web, the technical standards RDF (Resource Description Framework) and OWL (Web Ontology Language), SPARQL (SPARQL Protocol and RDF Query Language) have been developed by the W3C to facilitate the exchange of semantically rich information, and formally represent the metadata (Antoniou and Harmelen, 2008). The complexity and variety of applications referring to the Semantic Web are increasing every day, Semantic Web technologies promote common data formats and exchange protocols on the Web and have been widely used in a variety of application domains, for example, in data integration, cataloguing services, knowledge management, knowledge representation and sharing, resources discovery and classification, and so on.

The Semantic Web is developed by a layered approach, with each development process step building a layer on top of another layer, this layer structure of the Semantic Web is called the Semantic Web Stack (Antoniou and Harmelen, 2008). As shown in Figure 2-10, the Semantic Web Stack, created by Tim Berners-Lee, also called Semantic Web Layer Cake, which illustrates the architecture of the Semantic Web.

Each layer in this stack exploits and uses the capabilities of the layer below (Antoniou and Harmelen, 2008). It shows the hierarchy of the languages and technologies. It illustrates how these technologies are implemented per step to achieve the Semantic Web, and also shows how Semantic Web is an extension of the current Web, but not a replacement of the current Web.

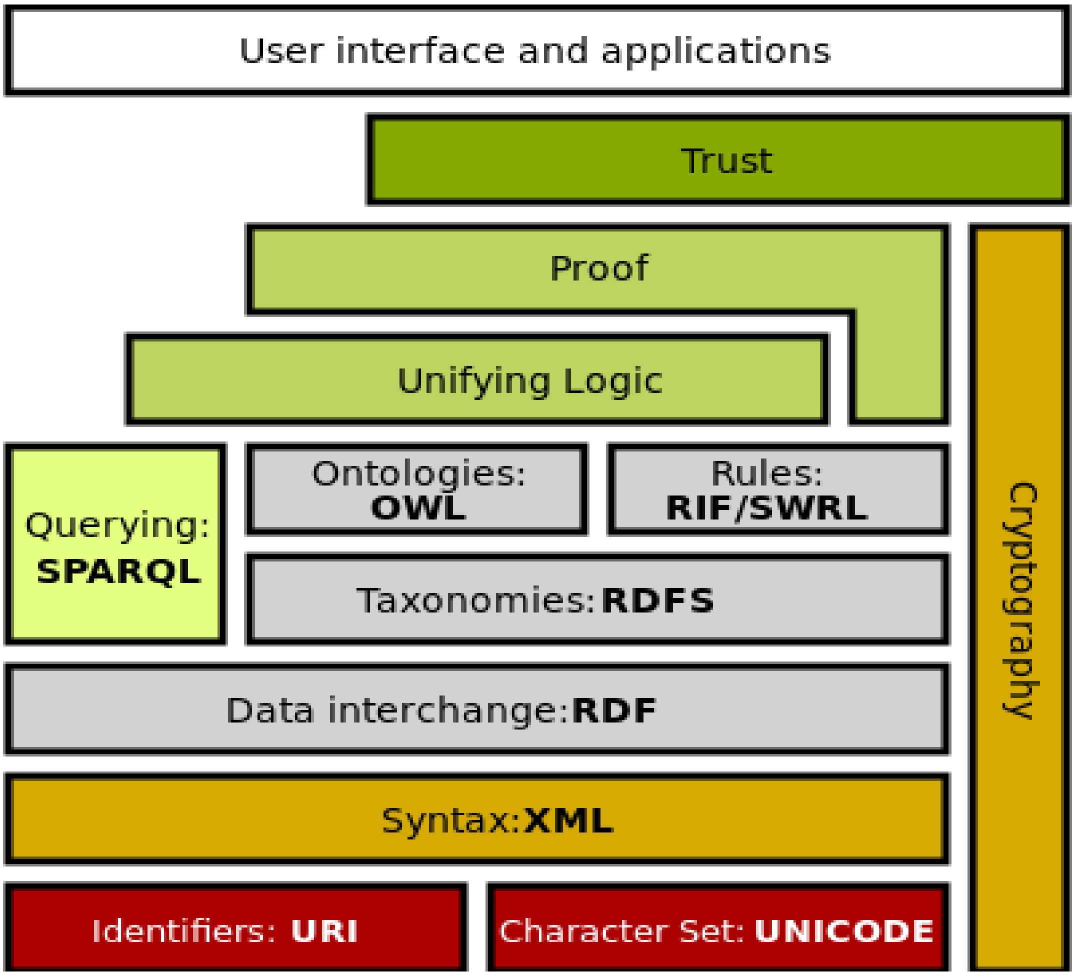


Figure 2-10 Semantic Web Layer Cake (Source: (Antoniou and Harmelen, 2008))

The functions and relationships between each layer are explained below:

- Unicode + URI: This is the foundation layer of the Semantic Web, including Unicode and URIs. Unicode is an information technology standard, a character-encoding system that serves to uniformly encode, represent, and handle content on the Web. A URI (Uniform Resource Identifier) is a sequence of characters, used to give a unique identity to resources on the Web (Bosak and Bray, 1999).
- XML: Extensible Markup Language (XML), is a language designed to describe structured documents in a format that is both human-readable and machine-readable, it allows to define the structured Web documents with a user-defined vocabulary. It is particularly suitable for sending documents across the Web (Bosak and Bray, 1999).
- RDF: Resource Description Framework (RDF) is a data model, a formal language for describing structured information in a machine-accessible way. An RDF data model can be described in a variety of syntaxes, such as RDF/XML, N3, Turtle, and so on.

The RDF data model does not rely on XML, but RDF has an XML-based syntax, which is why it locates on top of the XML layer. As a graphical formalism, RDF expresses the resources in a form of ‘‘Subject-Predicate-Object’’ triples, which sometimes can be called a statement. RDF consequently is often viewed as the basic representation format for developing the Semantic Web (Hitzler, Krötzsch and Rudolph, 2010).

- RDFS: RDF Schema is an extension of the RDF, it provides a data-modelling vocabulary for describing properties and classes of RDF data, and provides a semantic hierarchy of properties and classes. It can be viewed as a primitive language for writing ontologies. RDF Schema is based on RDF. RDFS also support reasoning, which is not strong due to the lack of rich expressiveness.
- Ontology OWL: Web Ontology Language (OWL) is the ontology layer of the Semantic Web, OWL extends RDF Schema by adding more advanced constructs to describe more complex semantics of RDF statements. It allows the additional constraints on classes and properties, such as cardinality constraints, restrictions of values, or characteristics of classes such as Boolean combinations of classes. It is based on description logic and therefore brings reasoning capabilities to the Semantic Web (Antoniou and Harmelen, 2008).
- SPARQL: short for “SPARQL Protocol and RDF Query Language”, is an RDF query language, it can be used to query any information encoded in RDF format. Querying language is essential for Semantic Web applications to achieve the retrieve information function.
- RIF: Rule Interchange Format (RIF) is a Semantic Web rule interchange format. It uses XML language to express Web rules, and it can be used to express the relations that cannot be directly expressed using the description logic used in OWL. It includes three dialects: Core dialect, RIF Basic Logical dialects (RIF-BLD), and RIF Production rule dialects (RIF-PRD).
- Logic layer: The logic layer enhances the ontology language further and enables the writing of application-specific axioms and inference rules (Antoniou and Harmelen, 2008).
- Proof layer: The Proof layer executes the rules and evaluates the inferred results to proof validation.

- Trust layer: Based on the trusted agents' recommendations, the trust layer establishes a certain trust relationship through Proof exchange and digital signature.

2.6 Ontology

Ontology, as the backbone of the Semantic Web, is not simply a conceptual framework but a concrete, syntactic structure that models the semantics of a given domain in a machine-understandable language (Jacob, 2005). This section presents the development of ontology from the aspects of ontology definition, ontology modelling methods, ontology rules and query languages, ontology editors, and the application of ontology in the knowledge management engineering domain.

2.6.1 Ontology definitions

Drawn from the philosophy domain, the term ontology refers to the science that describes the nature of being and their relations. However, in computer and information science, ontology has been widely used in the domain of knowledge management, representation and sharing by conceptualizing the knowledge concepts and their relations. There are several widely cited definitions when the ontology is adopted in computer science and artificial intelligence.

There is no universally accepted formal definition of ontology, but the one widely cited is the one proposed by Gruber (1993, 1995), in which he defined ontology as *''an explicit specification of a conceptualization''*.

Grüninger and Fox (1995) defined ontology as a formal method to describe the entities and the relationships, attributes and constraints between them.

Uschold and Grüninger (1996) summarized in their research on ontology development that a lack of a shared understanding had led to poor communication between people and their organisations, and difficulties in identifying requirements when building IT systems. Based on the ontology definition from Gruber (1993, 1995), they redefined ontology as a term that refers to the shared understanding of a given domain, and it can be used as a unifying problem-solving framework.

Swartout *et al.*(1997) described ontology as a hierarchical set of terms used to describe a domain that can be used as a framework foundation for a knowledge base.

Studer, Benjamins and Fensel (1998) had an overview of the developed ontology method in the field of knowledge engineering (KE), they pointed out that ontology provides a vocabulary of terms and relations with which to model the domain, and plays a key role in analysing, modelling and processing the domain knowledge. They elaborated Gruber's (1993, 1995) definition of ontology as 'Conceptualization' refers to an abstract model formed by identifying the concepts related to a certain phenomenon; 'Explicit' means that the types and constraints of terms used are explicitly defined; 'Formal' refers to the fact that the ontology should be machine-readable, which excludes natural language; 'Shared' reflects the notion that an ontology is not private to some individual, but it can be shared and accepted with others.

Stevens, Goble and Bechhofer (2000) had an overview of the ontologies applications in the bioinformatics and molecular biology domains, illustrated the ontology building process, techniques and methods in use at that time, and introduced the use of ontologies within bioinformatics through examples taken from this given domain. They also gave their understanding of Gruber's ontology definition that ontology is a concrete form of a conceptualisation of given domain knowledge. The conceptualisation is the knowledge about entities and the relationships between them. The specification is an expression of a concrete form of this conceptualisation.

The ontology has been characterised by Noy and McGuinness (2001) as a common domain vocabulary, which defines the domain knowledge or information concepts and clarifies their relations to facilitate communication among domain experts and achieve the communication between the domain experts and knowledge-based systems. And they also developed the ''7-Steps'' ontology building guidance, which is widely adopted to develop the ontology in different domains. The ''7-Steps'' method includes determining the domain and scope, considering reusing existing ontologies, enumerating import terms, defining the classes, defining the slots, defining the facets of the slots, and creating instances. It is also known as the SKME (Simple-Knowledge Engineering Methodology) (Zhou, Goh and Shen, 2016).

Stanford University's Knowledge Systems Laboratory (KSL) has explained ontology as a formal and declarative knowledge representation system, the terms related to the relative subject domain and the logical relationship statements between the terms are declared in this system. So, based on the above ontology understanding, Darlington and Culley (2008) consider ontology as a useful vocabulary to represent and share knowledge about a specific subject area and a series of relations among them and make it explicit.

According to Tserng *et al*.(2009), ontology is considered as an explicit formal specification of the concepts in a specific domain and the relations among them, and there are two vital components of ontology, domain concepts and relations respectively.

Hitzler, Krötzsch and Rudolph (2010) described ontology as a description of knowledge about a domain of interest, and its core is a formally defined machine-processable specification.

As briefly reviewed above, the researchers gave their different understanding of ontology within different domains, there is not a universally accepted definition of ontology. In general, ontology provides a hierarchy of concepts (entities) in a given domain and clarifies their relationships between them, it provides a platform to achieve interoperability between people and machines and promotes the knowledge reusing and sharing in a common understating syntax.

2.6.2 Components of an ontology

The common components of an ontology include classes, relations, functions, attributes, restrictions, rules, instances and axioms (Gruber, 1993). Based on the enumerated main ontology components from Maedche and Staab (2001), both Öhgren (2004), Hu *et al.*(2013), Hou (2015), and Reyes-Peña and Tovar-Vidal (2019) adopted a mathematical definition in their research to represents an ontology in a '5-tuple' form:

Öhgren (2004): $O := \{C, R, H^{C}, rel, A^{O}\}$

or

Hu *et al.*(2013): $O = < C, P, I, R, A >$

or

Reyes-Peña and Tovar-Vidal (2019): $\text{Ontology} =< C, R, F, I, A >$

where,

- O: the ontology of the given domain.
- C or H^C: a concepts hierarchy or Classes, represents a set of entities within the given domains.
- R: Relations, describe the relations between concepts or a concept's properties.
- P: Properties, a set of attributes related to the concepts in an ontology, there are mainly two properties: the object property and data property.
- I: Instances or Individuals, the objects that represent the base components of an ontology (Slimani, 2015).
- A or A^O: a set of ontology axioms, are the restrictions or rules, that are used to restrict the attributes and relations for classes or instances.
- F or *rel*: means functions, the particular types of relations that can be used in place of an individual term in a statement (Öhgren, 2004).

There are many similarities in the structure of ontologies, regardless of what language they are expressed in. It can be seen from the various definitions of ontology that the essential components of an ontology are the individuals (instances), classes (concepts), attributes, and relations or properties (Gruber, 1993).

2.6.3 Web Ontology Language (OWL)

Web Ontology Language (OWL) is a W3C (2004) recommended formal language for ontology modelling on the Semantic Web. It is one of the core technologies of Semantic Web Stack. As a computational logic-based language, instead of just representing information on the Web, it allows the applications to process the information and represent the information in a format that both human beings and machines can understand, it has enhanced the interoperability between people and computers. Different from most programming languages, the OWL does not use the Unique Name Assumption (UNA), which means that two different names could refer to the same individual (DeBellis, 2021).

OWL is a vocabulary extension of RDF, because of its strong knowledge expressivity capability and efficient reasoning (ontology-based reasoning and rule-based reasoning) capability, OWL makes it possible to describe concepts in an unambiguous manner based on set theory and logic (DeBellis, 2021). The OWL language can be used to formalize a particular domain by defining classes and properties of them, to define the individuals and assert properties about them, and to reason about the classes and individuals (Smith, Welty and McGuinness, 2004).

The OWL language system includes three increasingly expressive sublanguages, called *species* of OWL: OWL Lite, OWL DL, and OWL Full, the higher levels of the language contain the lower levels. As summarized in the book << Foundations of Semantic Web Technologies >> by Hitzler, Krötzsch and Rudolph (2010), the main characteristics of each sublanguage are shown in Figure 2-11.

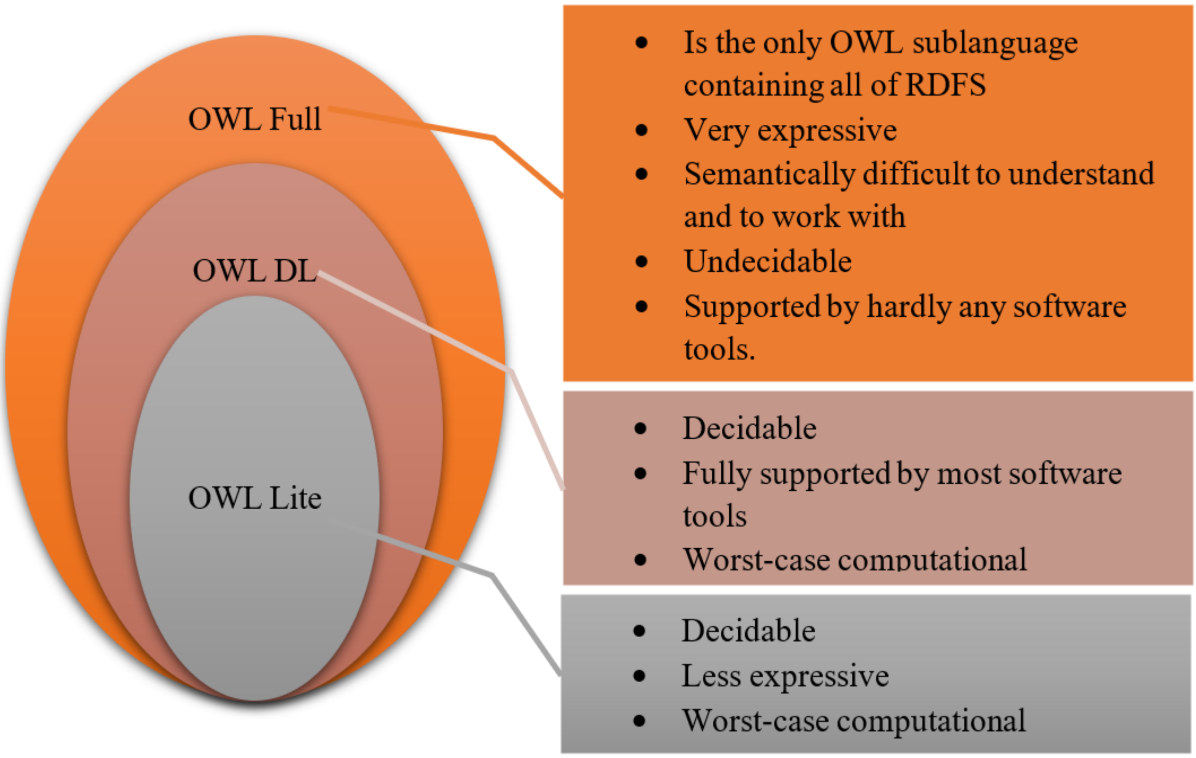


Figure 2-11 OWL three sublanguages

OWL Lite: It is suitable for users who primarily need a classification hierarchy and simple constraints. It has a less expressive capability and has lower formal complexity than OWL DL, and it only supports cardinality with constraints values of 0 or 1 (McGuinness and Harmelen, 2004).

OWL DL (Description Logics): Contains OWL Lite, it is suitable for users who want maximum expressiveness while retaining computational completeness and decidability of reasoning systems (McGuinness and Harmelen, 2004). It permits efficient reasoning support, fully supported by most software tools, but loses full compatibility with RDF (Grigoris and Frank, 2004).

OWL Full: Contains OWL DL and OWL Lite, it can be viewed as an extension of RDF, it has maximum expressiveness and is fully upward-compatible with RDF, both syntactically and semantically (Grigoris and Frank, 2004). However, it is undecidable and hardly any of the inference engines support it (Hitzler, Krötzsch and Rudolph, 2010).

Because of the ability to represent rich and complex knowledge and reasoning ability, OWL is recommended by W3C as a proper ontology description language to be used in ontology development. The OWL languages have been widely used for modelling ontologies, which provides a high-level description capability for web resources, however, the expressiveness of OWL is not strong enough to support the reasoning in ontologies. To overcome this deficiency, the SWRL (Semantic Web Rule Language) has been developed to extend OWL-DL, it allows the users to write Horn-like rules that can be expressed in terms of OWL concepts (Horrocks et al., 2004). Compared to the OWL, SWRL provides more powerful deductive reasoning capabilities that can reason about OWL individuals (O'Connor, Tu, *et al.*, 2007; Moreira, 2012). The following section further explains the SWRL.

2.6.4 Semantic Web Rule Language (SWRL)

Semantic Web Rule Language, short as SWRL, is a standard ontological rule language, which is based on the integration of the ''OWL DL and OWL Lite sublanguages of the OWL Web Ontology Language with the Unary/Binary Datalog RuleML sublanguages of the Rule Markup Language'' (O'Connor *et al.*, 2005). It provides users with the ability to write Horn-like rules that can be expressed in terms of OWL concepts (classes, properties, individuals) to reason about OWL individuals (O'Connor, Shankar, *et al.*, 2007; Hu *et al.*, 2013; Munir and Sheraz Anjum, 2018; Wu *et al.*, 2021). SWRL provides deductive reasoning capabilities by adding rules to extend OWL-DL, after defining the SWRL rules into the ontology knowledge base, the new facts can be inferred from the existing OWL knowledge base (O'Connor, Shankar, *et al.*, 2007).

An SWRL rule is in the form of an implication between an antecedent (body) and a consequent (head) that are connected by the implication symbol arrow '→', and it is written in a human-readable syntax as: antecedent –> consequent, which can be read as: if the condition specified in the antecedent applies, then the condition specified in the consequent must also hold. Both the antecedent (body) and the consequent (head) consist of a conjunction of zero or more atoms, combined by the symbol caret '∧', and the question mark (?) distinguishes variable names from individuals. SWRL rules reason about OWL individuals, primarily in terms of OWL classes and properties (O'Connor *et al.*, 2005). There are seven types of atoms provided by SWRL: class atoms, individual property atoms, data valued property atoms, different individual atoms, same individual atoms, built-in atoms, and data range atoms (Horrocks *et al.,* 2004; Yan *et al.*, 2015).

In this syntax, an SWRL rule has the form:

$$\textbf{\textit{antecedent –> consequent}} \quad \textbf{\textit{(1)}}$$

where the antecedent and consequent consist of the conjunction of atoms, as expressed in Equation (2):

$$Atoma_1 \wedge Atoma_2 \wedge \cdots Atoma_n \rightarrow Atomc_1 \wedge Atomc_2 \wedge \cdots Atomc_m \quad (2)$$

As shown in Equation (3), every atom is a predicate in OWL:

$$p(arg_1, arg_2 \ldots \ldots arg_n) \quad (3)$$

where p is a predicate symbol defined in OWL and arg_1, arg_2 … arg_n are the expression of specific terms or parameters. The form of the atom can be C(x), P(x, y), sameAs(x, y) or differentFrom(x, y), where C is an OWL classes description, *P* represents the OWL properties, and x, y are either variables, OWL individuals or OWL data values, whilst sameAs(x, y) or differentFrom(x, y) is the built-in functions of SWRL (Horrocks *et al.*, 2004; Yan *et al.*, 2015).

A simple use of this syntax to assert that the combination of the *hasParent* and *hasSister* properties implies the *hasAunt* property. Informally, this rule could be written as: *hasParent(?X, ?Y) ^ hasSister(?Y, ?Z) ⇒ hasAunt(?X, ?Z)*

In a human-readable scene, it says: if individual *X hasParent Y*, and individual *Y hasSister Z*, then the asserted relation between the individual *X* and *Z* is: *X hasAunt Z.*

2.6.5 Ontology querying

2.6.5.1 SPARQL

A number of semantic query languages have been developed to retrieve and extract information in various formats and sources. SPARQL, pronounced *sparkle* /ˈspɑːkəl/, short for ‘‘SPARQL Protocol and RDF Query Language’’, is a W3C recommended official semantic query language for Linked Open Data and RDF databases (Harris and Seaborne, 2013). As the most advanced query language, SPARQL enables users to retrieve and manipulate information that can be mapped to RDF format (DuCharme, 2013). As OWL can be expressed in RDF format, SPARQL is designed for querying RDF, so SPARQL is used as an OWL query language in many applications (O’Connor and Das, 2009).

SPARQL has several query forms for different query purposes, they are SELECT query, which is used for standard queries, the result returns bound variables; CONSTRUCT query returns results in RDF graph; ASK simply generates a True/False query result, without specific content; DESCRIBE query is used to extract parts of the useful information that users deem (DeBellis, 2021).

In a SPARQL query string, a group graph pattern is delimited with braces: **{ },** variables are indicated by a question mark ‘‘?’’ prefix. Each SPARQL query consists of two parts. The first part at the beginning consists of several namespace prefixes (DeBellis, 2021). There are two clauses in a SPARQL query, the SELECT clause and WHERE clause, SELECT clause is used to confirm the variables to show in the query results, and the WHERE clause provides the graph pattern that matches the data graph. And can also add some constraint functions, like FILTER, to restrict the query results (Harris and Seaborne, 2013). A basic SPARQL SELECT query pattern is shown below:

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX owl: <http://www.w3.org/2002/07/owl#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?subject ?object

WHERE

{ ?subject rdfs:subClassOf ?object }

As explained in the research from O'Connor and Das (2009), even though the SPARQL is a powerful OWL query language, it does not have a native understanding of OWL. It can be used to effectively represent information in an RDF serialization, but it cannot understand the language structure of the serialized information it represents. Moreover, some OWL constructs do not have canonical RDF serialization. Therefore, another OWL query language SQWRL (Semantic Query-Enhanced Web Rule Language) has been developed to support comprehensive querying of OWL.

2.6.5.2 SQWRL

SQWRL (Semantic Query-Enhanced Web Rule Language; pronounced *squirrel* /ˈskwɪrəl/) is an SWRL-based query language that provides SQL-like operators for extracting information from OWL ontologies (O'Connor, 2016). Different from the OWL and SWRL, SQWRL adopts the Unique Name Assumption (UNA) for matched individuals when querying. It replaces the rule consequent with a retrieval specification. The SWRL's built-in libraries have been applied to the SQWRL to define a set of operators that can be used to construct retrieval specifications (O'Connor and Das, 2009). The existing SWRL editors can be implemented to write and edit SQWRL queries, the core SQWRL operator is *sqwrl:select.* It contains one or more arguments, which are typically variables used in the schema specification of the query, and builds a table with the arguments as columns of the table (O'Connor and Das, 2009). By using these operators, queries can easily retrieve the results of two or more closure operations, and it allows writing much more complex queries at one time.

The left side of the SQWRL query operator has the same form as the SWRL antecedent with its associated semantics, and there are no restrictions on the left side of the query – any valid SWRL antecedent is a valid SQWRL pattern specification, but in the consequent, there is an *sqwrl:select* statement that lists every parameter that users queried the value of every time the rule fires (DeBellis, 2021).

The SQWRL language provides two sets of query operators (O'Connor, 2016): the core operators and the collection operators. The core operators that use an SWRL rule antecedent as a pattern specification and replace the rule consequent by SQWRL selection operators; to support more advanced querying capabilities, SQWRL provides a series of collection operators with the capabilities of grouping, aggregation, as well as the limited forms of negation as failure and disjunction.

SQWRL query can be operated in combination with SWRL rules to retrieve the knowledge which is inferred by SWRL rules. As the SQWRL queries have no access to the information it accumulates from within a rule, so the results from the SQWRL cannot be written back to the OWL ontology and do not perform any ontology modifications (O'Connor, 2016).

The following examples illustrate the pattern specification of SQWRL query:

List all individuals in an ontology:

owl:Thing(?i) -> sqwrl:select(?i) ***(4)***

As said, the queries can be operated in combination with SWRL rules to retrieve the knowledge which is inferred by SWRL rules. For example, first of all, the SWRL rule is written in the OWL ontology knowledge base to define persons over 17 years old as adults. Then the corresponding SQWRL query can be written to retrieve all the adults in the OWL ontology.

SWRL Rule: Person(?p) ^ hasAge(?p, ?age) ^ swrlb:greaterThan(?age, 17) -> Adult(?p)

SQWRL query: Adult(?p) -> sqwrl:select(?p)

After executing the query, the retrieved individuals would be listed in the ontology editor but the query results cannot be written back to the OWL ontology.

2.6.6 Ontology editors

Ontology design requires the application of software tools, available in commercial or open-source. This section lists some ontology editors (Ukpe and Mustapha, 2016).

Protégé: developed by the Stanford Center for Biomedical Informatics Research (BMIR) at Stanford University (Musen, 2015a). Protégé is a free, open-source ontology editor and framework for constructing domain models and knowledge-based applications with ontologies (O'Connor and Das, 2006; Sachs, 2006). Protégé's plug-in architecture can be adapted to build both simple and complex ontology-based applications (Ukpe and Mustapha, 2016). Protégé has become the most widely used software for building and maintaining ontologies (Musen, 2015a). Protégé has the Protégé desktop system and a cloud-based WebProtégé.

OntoStudio (formerly OntoEdit): a most widespread commercial modelling environment for creating and maintaining ontologies. Based on the higher-order logic method, the highly expressive ontology language ObjectLogic (F-Logic 2) and all subsets of the Semantic Web-Standards (OWL2 RL, RDF(S) are implemented (Alatrish, 2013).

Vitro: was originally developed at Cornell University, it is an integrated web-based ontology and instance editor and Semantic Web application with customizable public browsing, it can create or load ontologies in OWL format, edit instances and relationships, and so on (Ivanovic, 2021).

SWOOP: developed by the University of Maryland, is an open-source, hypermedia inspired Web-based featherweight OWL ontology editor. It is built primarily as a Web ontology browser and editor (Alatrish, 2013), designed with OWL validation, presentation syntax views, and supports multiple ontology environments (Kalyanpur *et al.*, 2006; Kapoor and Sharma, 2010; Ukpe and Mustapha, 2016).

There are some other archived or inactive ontology editors, for example, the Knoodl, WebOnto the Apollo from the Open University (Matoušek, Král and Falc, 2004), Ontolingua developed by Stanford University Knowledge Systems Lab, NeOn Toolkit which is an open-source, multi-platform ontology editor that supports the development of ontologies in F-Logic and OWL/RDF, and so on (Ukpe and Mustapha, 2016).

In this research, the Protégé editor (O'Connor and Das, 2006) has been used to build the ontology knowledge model, its SWRL tab plugin supports the editing and execution of SWRL rules. The SWRL tab is an SWRL API-based development tool that provides a set of standalone graphical interfaces for managing SWRL rules and SQWRL queries.

2.7 Methodologies for building ontology

The ontology development process is complex, and it is critical to choose which languages, tools, and methodologies to use for developing ontologies in different domains and for different purposes. After reviewing the ontology development languages and supported development tools in the previous sections, this section reviews the ontology development methodologies.

There is no one-size-fits-all construction methodology for ontology development within different domains, and no one method that is superior to the other (Noy and McGuinness, 2001; Zhang *et al.*, 2019). The decision of which approach to be applied depends on the characteristics of the ontology to be developed. The current methodologies for ontology development are analysed as follows.

- Uschold and King's methodology (Uschold and King, 1995):

Uschold and King's methodology refers to first methodology which is developed based on the experience of developing the Enterprise Ontology in 1995 (Fernández-López, 1999; Fernández-López and Gómez-Pérez, 2002; Andryani, Negara and Darma, 2015). According to Uschold and King (1995), the stages to develop an ontology in this method are shown as below:

- *Purpose Identifying*: To clarify the reason to build the ontology and what its intended use is, and help to identify the range of its potential users.
- *Building the Ontology*: there are three sub-steps explained below

    - Ontology capture: identify the core concepts and the relations between them in a given domain.
    - Ontology coding: the explicit representation of the above-captured knowledge in formal ontology languages.
    - Integrating existing ontologies: it is worth considering reusing ontologies that already exist.

- *Evaluation:* make technical judgments about the ontologies, the associated software environments, and documents against the reference frame.
- *Documentation:* according to the type and purpose of the ontology to establish guidelines for documenting ontologies.

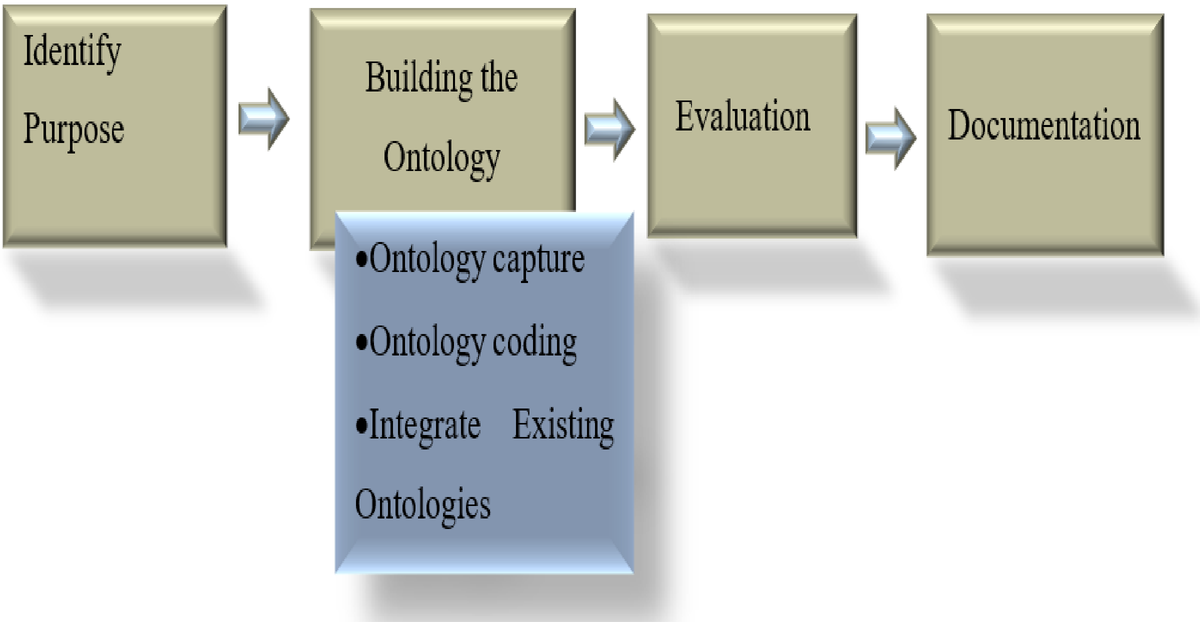


Figure 2-12 Uschold and King's methodology

At the ontology building stage, Uschold and King (1995) proposed three methods to define the main terms of ontology: Top-down approach, Bottom-up approach, and Middle-out approach. The characteristics of each approach are shown below:

- Top-down approach: the most general terms are defined first, and then specialized these terms into more specific concepts. This approach results in better control of the level of detail. However, this method risks less stability in the model, which leads to rework and a greater overall effect (Uschold and Gruninger, 1996; Corcho, Fernández-López and Gómez-Pérez, 2003).
- Bottom-up approach: the most specific concepts are defined first, and then generalized into more general concepts (Noy and McGuinness, 2001; Corcho, Fernández-López and Gómez-Pérez, 2003). This approach results in a very high level of detail with the risk that many of the concepts may not be important in the final ontology (Uschold and King, 1995).

- Middle-out approach: a mix of Top-down and Bottom-up methods, in which the most important concepts are defined first and then generalized and specialized appropriately. This method leads to less rework and less overall effort (Noy and McGuinness, 2001).

Although Uschold and King's methodology is widely recognized as the first ontology building method, it does not describe any techniques for performing each of the above four stages (Abdelghany, Darwish and Hefni, 2019), and there is a lack of conceptual ontology modelling activities from the domain knowledge capture to the ontology implementation(Fernández-López and Gómez-Pérez, 2002).

- Grüninger and Fox's Methodology (TOVE)

As one of the earliest ontology development methodologies, the TOVE (Toronto Virtual Enterprise) method was proposed by Grüninger and Fox in 1995, which is used in the Enterprise Integration Laboratory for the design and evaluation of integrated ontologies (Grüninger and Fox, 1995). The framework of this approach is illustrated in Figure 2-13. The main steps of this method are as follows:

- Motivating scenarios: according to Grüninger and Fox, *''the development of ontologies is motivated by scenarios that arise in the applications''*. Identify intuitively possible applications and solutions. When it comes to the new or extended ontologies, one or more motivating scenarios should be described, as well as the cooperating intended solutions to the problems presented in the scenarios. Providing scenarios helps to understand the motivations for the developed ontologies from the perspective of the applications (Fernández-López and Gómez-Pérez, 2002).
- Informal competency questions: based on the scenarios proposed in the first stage, a set of questions, called competency questions, are raised to determine the scope of the ontology. The competency questions do not need to be exhaustive, they serve as constraints on what the ontology can be, and they do not generate ontological commitments but help to evaluate whether the new or extended ontology meets the requirements (Corcho, Fernández-López and Gómez-Pérez, 2003).
- First-Order Logic: Terminology: specification of the terminology of the ontology, like the objects, properties of objects, relations, and attributes, must be specified using first-order logic (or equivalently, in KIF).

- Formal competency questions: the proposed competency questions of the new or extended ontologies are formalised in terms of the formally defined terminology.
- First-Order Logic: Axioms: the axioms that specify the definitions of concepts and constraints on their interpretation are defined in first-order logic. The axioms in the ontology are necessary and sufficient to express the competency questions (Jones, Bench-Capon and Visser, 2007).
- Completeness theorems: an evaluation stage that evaluates the competency of the ontology by defining the conditions under which the solutions to the questions are complete (Jones, Bench-Capon and Visser, 2007).

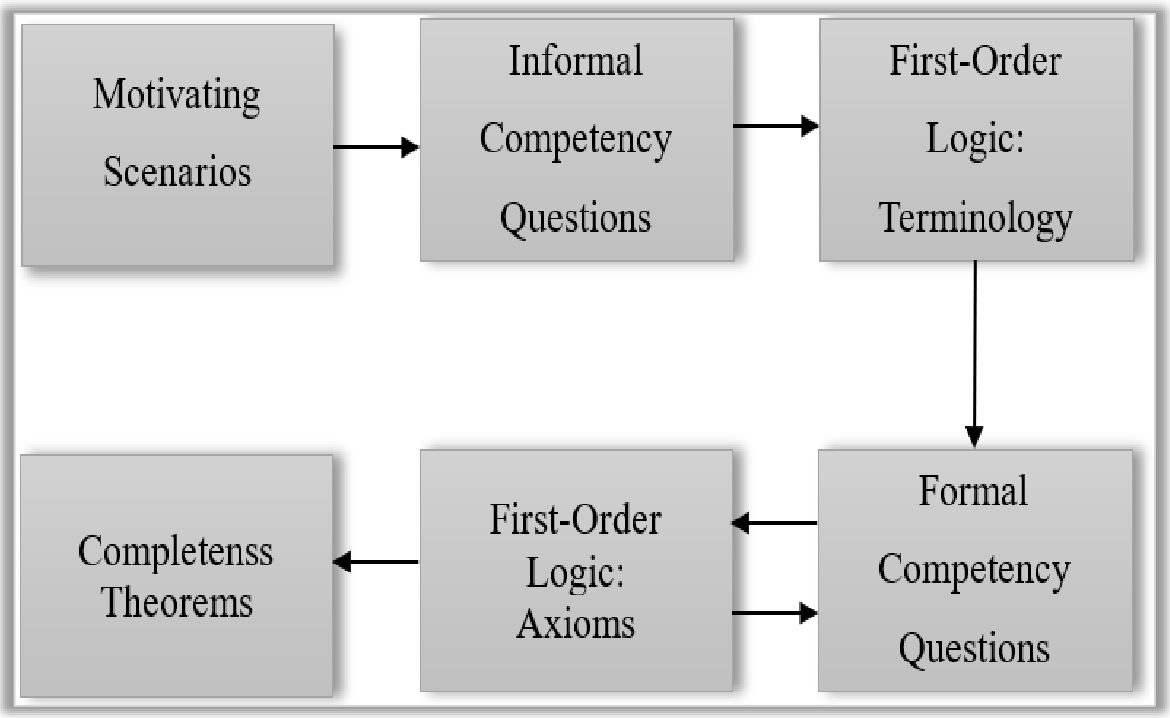


Figure 2-13 Grüninger and Fox's Methodology

Grüninger and Fox’s TOVE methodology is a logic-based formal method that can be used to transform the informal natural-language-expressed scenarios into logic-expressed computable models. However, this methodology lacks the life cycle model selection process for modelling ontology, and the description of techniques or activities are not described in detail (Fernández-López and Gómez-Pérez, 2002). Although this method has been used to develop some ontologies, the domain is limited to business (Abdelghany, Darwish and Hefni, 2019).

- The METHONTOLOGY methodology

The METHONTOLOGY is a well-structured method to build ontologies from scratch, which enables to build ontologies at the knowledge level (Fernández-López, Gómez-Pérez and Juristo, 1997; Fernández-López *et al.*, 1999). As shown in Figure 2-14, the METHONTOLOGY proposed an ontology building life cycle which is consisted of a set of activities in three categories: Management activities, Development activities, and Support activities (Gómez-Pérez, Fernández-López and Corcho, 2004). The separate development processes in these three activities categories are illustrated as follows:

- Ontology management activities include planning, project control, and quality assurance.
- Ontology development-oriented activities include specification, conceptualization, formalization, and implementation.
- Ontology support activities include knowledge acquisition, evaluation, ontology integration, documentation, and configuration management. This category of activities executes in parallel with the ontology development-oriented activities.

The ontological engineering workbench of ODE (Ontology Design Environment) (Fernández-López *et al.*, 1999) and WebODE (Corcho *et al.*, 2005) are used to provide technical support to METHONTOLOGY. Both of them support ontology building, covering the entire life cycle and automatically implementing ontologies (Fernández-López *et al.*, 1999).

Even the METHONTOLOGY proposed a set of activities to construct ontologies, but it does not provide any techniques for performing these activities in a formal manner (Abdelghany, Darwish and Hefni, 2019). A sizeable part of the development process is well instructed, but it does not provide structured guidelines on how to create a glossary or how to select appropriate concepts based on the given domain and its tasks (Szturcová and Rapant, 2013). Szturcová and Rapant (2013) proposed an extension of the METHONTOLOGY, this method was enhanced by defining a set of basic primitive concepts of the given domain within the early stage of ontology development. This method was tested by a case study of creating a prototype of a road-traffic system, and this research carried out that it is general and applicable within other methodologies that always start with a glossary of terms.

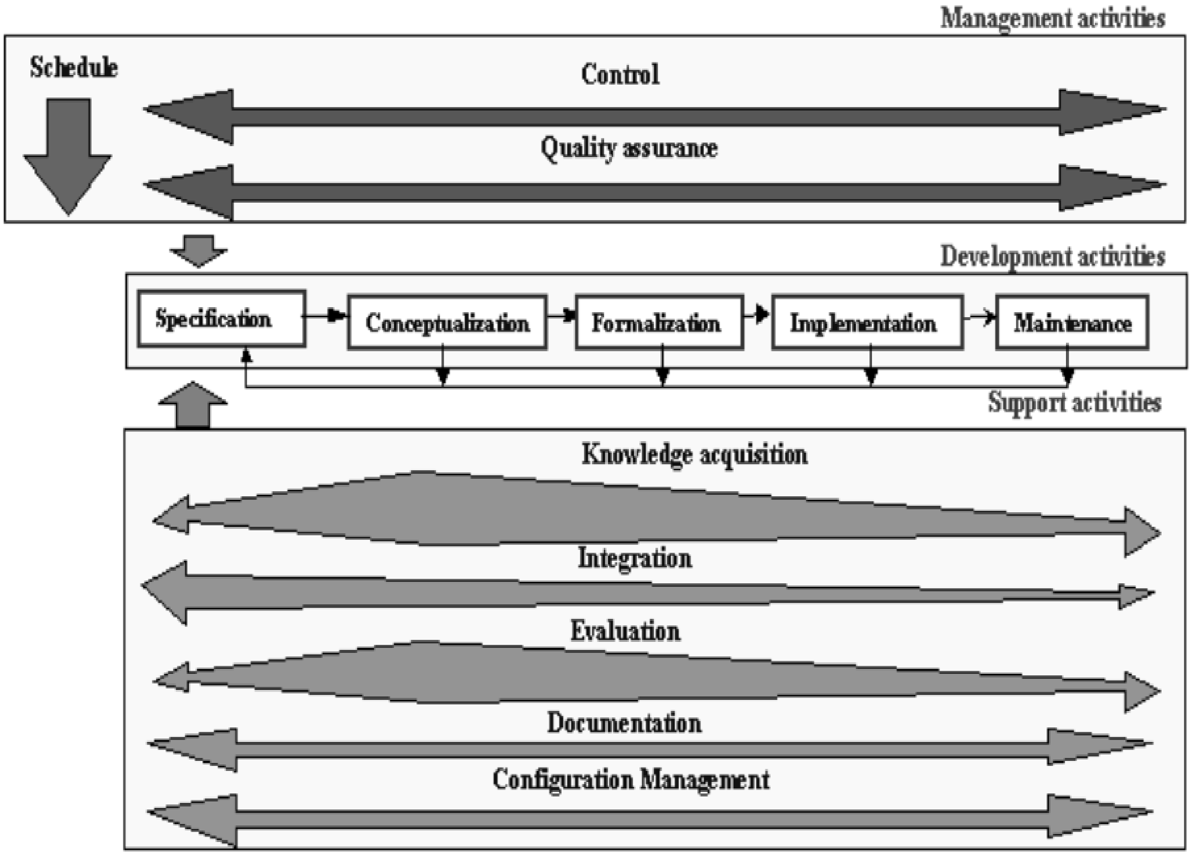


Figure 2-14 Ontology development life cycle (Source: (Fernández-López et al., 1999))

- On-To-Knowledge Methodology

The On-To-Knowledge methodology is a method used to develop ontology-based knowledge management (KM) systems (Staab, Studer and Gmbh, 2001). According to the research from Staab *et al.* (2001; 2004), the On-To-Knowledge Methodology has the development process with s shown in Figure 2-15, where:

- Feasibility study: a decision support to determine the economical and technical project feasibility and to identify problems and target solutions.
- Ontology kickoff: where the actual ontology development starts, the scope and domain of the ontology are defined, as well as the competency questions, the outcome of this stage is the ontology requirements specification document.
- Refinement: based on the requirements specification document from the last step, the application-oriented target mature ontology is developed in this stage. Domain experts define the relevant concepts and describe the relations between them.

- Evaluation: the target ontology is tested within the target application systems to check whether the ontology can meet the requirements specification or be able to answer competency questions.
- Maintenance: to maintain, update, insert or delete processes of ontology based on the strict rules.

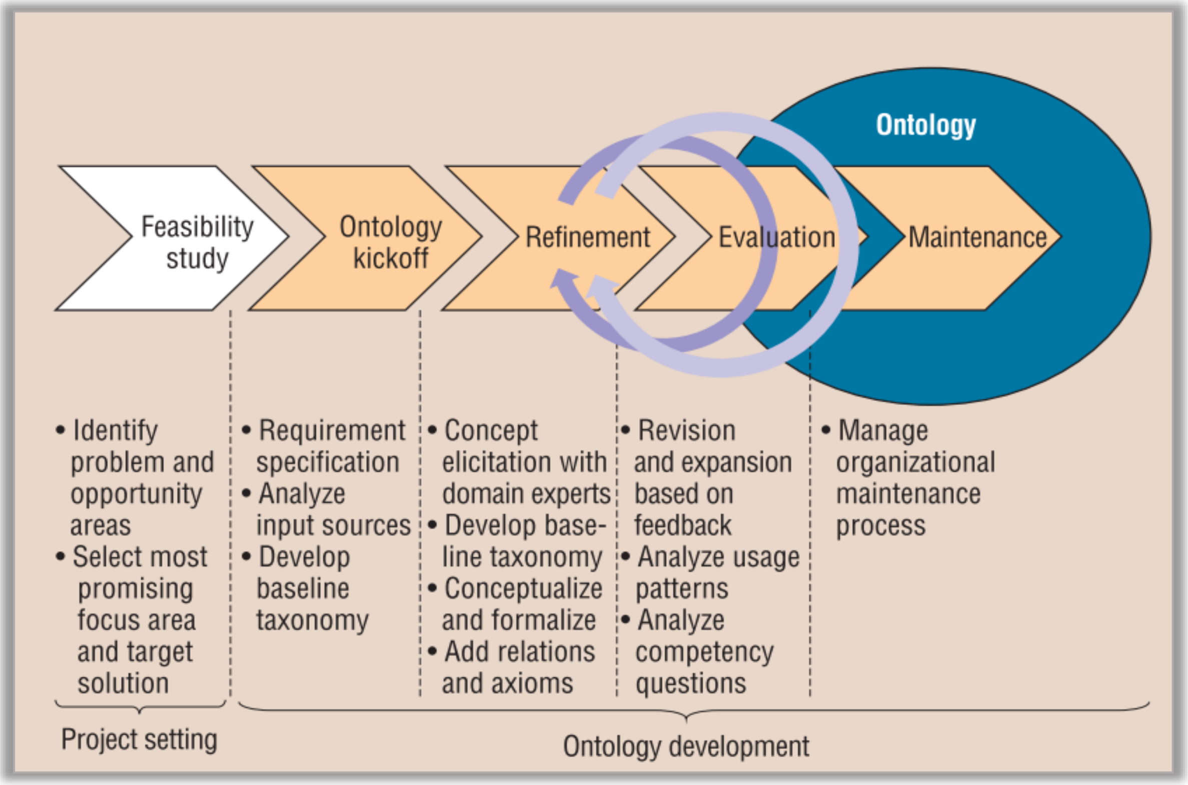


Figure 2-15 On-To-Knowledge Methodology (Source: (Sure, Staab and Studer, 2004))

- Simple-Knowledge Engineering Methodology (SKEM)

The Simple-Knowledge Engineering Methodology (SKEM), also known as the '7-Steps' methodology or Ontology Development 101 methodology, was proposed by Noy and McGuinness (2001), which provides a practical guide for domain ontology development. This method provides detailed and well-structured ontology building process guidelines for beginners and gives a practical ontology development example illustrated in the wine and food domain within the ontology-editing environment of Protégé-2000. Among the various ontology development methodologies, the SKEM is the most prevailing one. According to Noy and McGuinness (2001), ''an ontology is a formal explicit description of concepts in a domain of discourse (classes), properties between concepts describing various features and attributes of concepts (slots), and restrictions on slots (facets)''.

As shown in Figure 2-16, this method proposed 7 steps to construct an ontology:

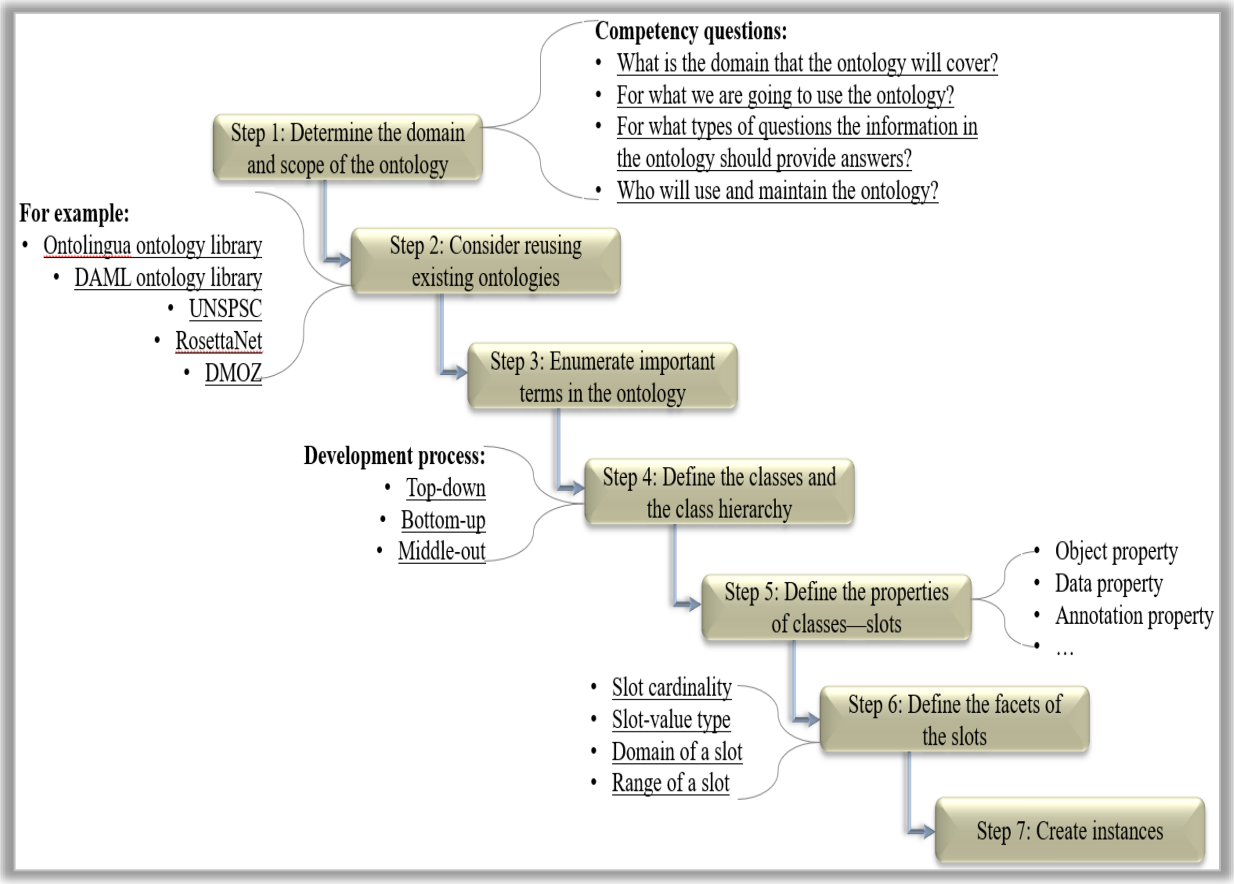


Figure 2-16 The SKEM ontology development process

- Step 1 Determine the domain and scope

In this step, the development of an ontology is starting with defining its domain and scope, that is, answering the competency questions. By following the Grüninger and Fox's Methodology (TOVE) (1995) to list a set of competency questions that can help to determine the ontology domain and scope, like :

- What is the domain that the ontology will cover?
- For what we are going to use the ontology?
- For what types of questions the information in the ontology should provide answers?

- Who will use and maintain the ontology?

The answers to competency questions are different from each different domain application, by answering these questions at the beginning of the ontology development can help users to limit the scope of the model throughout the entire ontology design process.

- Step 2 Consider reusing existing ontologies

It is almost always worth considering whether relevant ontologies that already exist can be reused or extended. With the rapid technological development in ontology in different domains, there are already many reusable ontologies available on the Web and in the literature in electronic form that can be imported into the ontology software environment, like the Ontolingua ontology library, or the DAML ontology library, even the commercial ontologies of UNSPSC, or DMOZ, etc (Noy and McGuinness, 2001). The ontology reusing can save ontology engineers time and effect.

- Step 3 Enumerate important terms in the ontology

When designing an ontology, it is always useful for ontology engineers to list all the relevant terms in the given domain. In the beginning, it is important to get the full list of concepts first without worrying about the overlap between the listed concepts, relationships between concepts, or properties of concepts. This step is the preparation work for the next two steps.

- Step 4 Define the classes and the class hierarchy

After listing all the relevant domain concepts in the previous step, this step adopts the methods from Uschold and King (1995) to classify the concepts into class hierarchies.

- Top-down development process: the most general concepts are defined first, and then, the specialize them into more specific concepts (Uschold and Gruninger, 1996; Corcho, Fernández-López and Gómez-Pérez, 2003).
- Bottom-up development process: the most specific concepts are defined first, and then generalized into more general concepts (Noy and McGuinness, 2001; Corcho, Fernández-López and Gómez-Pérez, 2003).

- Combination development process: also known as the Middle-out approach, is a mix of Top-down and Bottom-up methods, in which the most important concepts are defined first and then generalized and specialized appropriately. This method leads to less rework and less overall effort (Noy and McGuinness, 2001).

The Middle-out approach is often the easiest for many ontology developers to define the class hierarchy. But whichever method the developers choose, the class hierarchy building usually starts by defining classes by following certain rules, for example, ‘‘if class A is a subclass of class B, then every instance of A is also an instance of B’’.

- Step 5 Define the properties of classes-slots

Without defining the properties of classes, the classes alone in the ontology can not provide sufficient information to answer the competency questions proposed in step 1. All terms (concepts, relations, properties) are already enumerated in Step 3, after defining the class hierarchy in Step 4, most of the remaining terms are likely to be properties of these classes. properties of defined classes, and these properties become slots attached to classes, and all subclasses of a class inherit the slot of that class (Noy and McGuinness, 2001).

There are mainly three types of properties, the object property which defines the internal relations of the concepts; the data property which defines the relations between concepts and the data-type values; and the annotation property which presents the annotations on classes, properties, and individuals, etc.

- Step 6 Define the facets of the slots

The facets of properties refer to the characteristics of the values the properties or slots can have, like the value type, allowed values, or the quantity of the values (cardinality). In which, the slot cardinality defines how many values a property can have; slot-value type facets describe the slots values types, like String, Number, Boolean, Enumerated, etc. The domain and range of a slot also need to be determined

- Step 7 Create instances

The last step is to create the individual instances of the classes in ontology. The instances of classes can be defined by the following method, choosing a class first, then creating the individuals of the chosen class, and at the end, filling the slot values in.

This ‘7-Steps’ methodology provides a well-structured practical guide on how to build an ontology, it is easy to use for beginners and is the most prevailing one, so this method has been adopted in this research to build the ontology.

Even though this methodology is well explained, it does not include an evaluation step (Bravo, Reyes and Reyes Ortiz, 2019). The methodology proposed by Noy and McGuinness is clear and detailed but does not offer explicit instructions on conceptualization (Bautista-Zambrana, 2015).

- SAMOD methodology (Simplified Agile Methodology for Ontology Development)

Peroni (2017) developed a novel agile methodology for developing an ontology, named SAMOD (Simplified Agile Methodology for Ontology Development). This method develops ontologies through three simple and small interactive workflow steps that focus on creating well-developed and documented models by using significant exemplars of data.

The development of ontologies by adopting this method starts with defining the motivating scenario and informal competency questions (Grüninger and Fox, 1995), and then a glossary of terms is listed to define the domain. Based on the following three iterative steps– where each step ends with the release of a snapshot of the current state of the process called milestone: collect requirements and develop a modelet, merge the modelet with the current final mode, refactor the current final model, the detailed instruction of each step is briefly summarized in Figure 2-17.

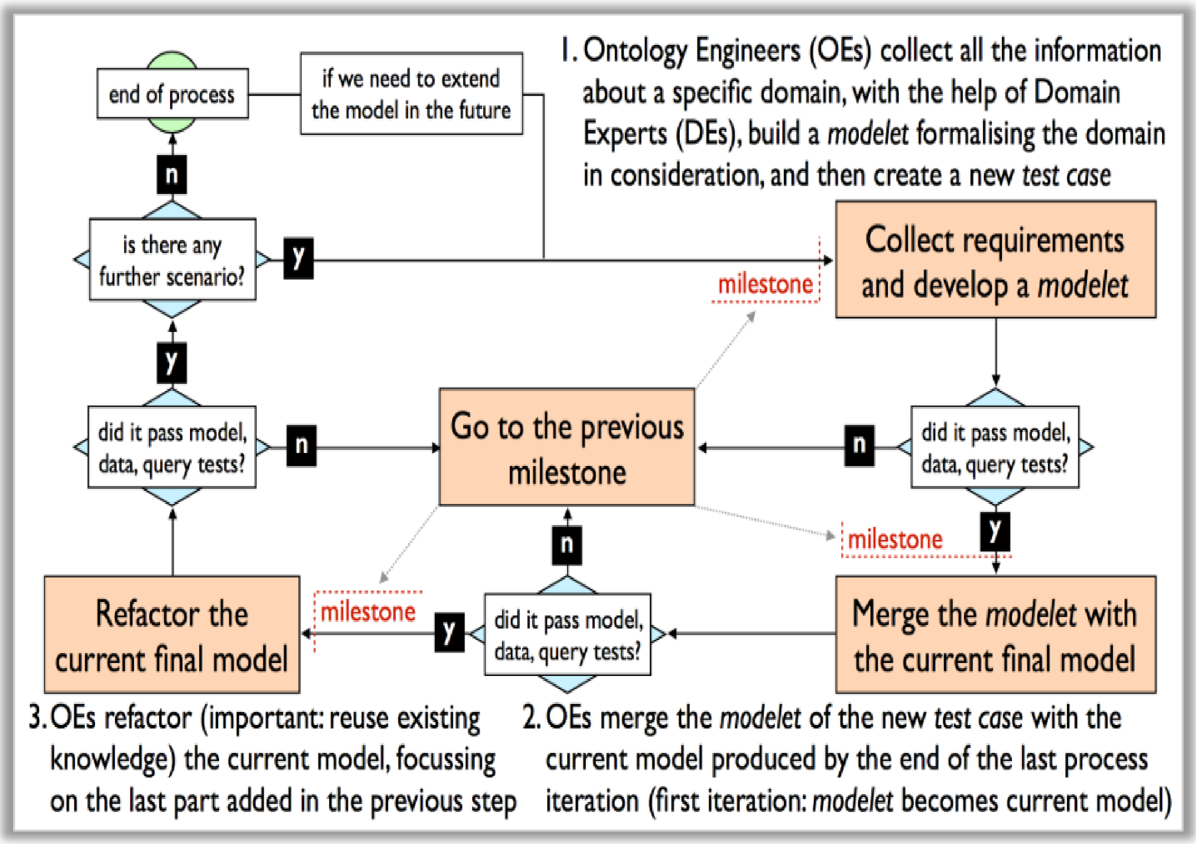


Figure 2-17 SAMOD methodology development process (Source: (Peroni, 2017))

- Agile Methodology for Ontology Development (AMOD)

Recently, to bridge the gap between ontology engineering and software engineering, the method of AMOD (Agile Methodology for Ontology Development) is proposed by Abdelghany, Darwish, & Hefni (2019), which is a methodology to develop ontologies by integrating fundamental agile principles and practices of software development (Smirnov *et al.*, 2021). As shown in Figure 2-18, the AMOD ontology development method has three phases: pre-game, development, and post-game (TAKHOM, 2019).

The advantage of this method is that it can be customized based on multiple factors, like ontology complexity, interest domain, ontology size, etc. According to the IEEE standard, Abdelghany, Darwish, & Hefni (2019) conducted a compliance analysis of different ontology development methods (for example, On-To-Knowledge, METHONTOLOGY, Grüninger and Fox’s methodology, Uschold and King methodology, etc.), the results showed that the AMOD methodology is more compliant with the IEEE standard than the other methods, it has also achieved a better satisfaction rate up to 56% for IEEE standard process.

AMOD has been adopted to develop the Software Project Time Management (SPTM), the successful application of AMOD in the development of SPTM ontology has indicated its applicability and a high degree of acceptance by ontology engineers, and this study also showed that the agile methods have simplified the ontology development activities (Abdelghany, Darwish and Hefni, 2019). Fundamentally, the AMOD methodology is a conceptual integration that does not address the heterogeneity of the presentations (Smirnov *et al.*, 2021).

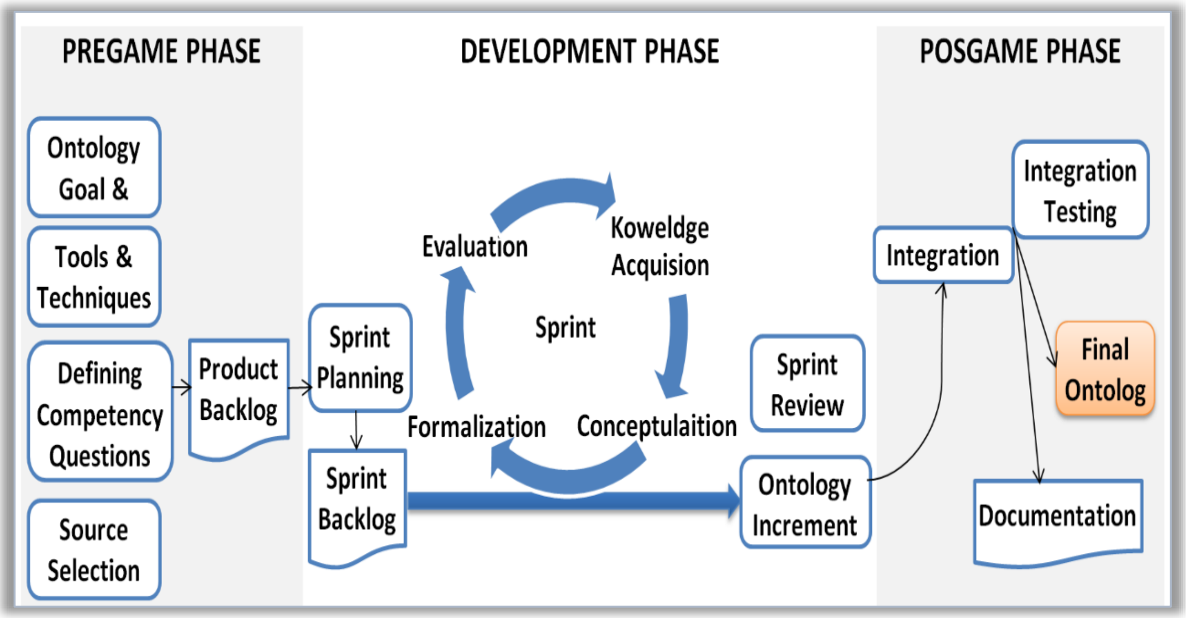


Figure 2-18 Framework of AMOD (Source: (Abdelghany, Darwish and Hefni, 2019))

As the concept of ontology was put forward in the early 20th century, many methodologies have been developed to support ontology development activities. Except for the above-analysed methodologies, there are also some other methodologies, for example, the SENSUS methodology, which is a method for building the skeleton of the specific ontologies starting from a huge ontology that covers more than 50,000 concepts (Swartout *et al.*, 1997); the KACTUS methodology, which is an application-driven project concerned with knowledge reuse in technical domains (Schreiber, Wielinga and Jansweijer, 1995); the CommonKads methodology by Schreiber *et al.*(2000); the ONIONS Methodology which developed by the motivation to integrate sources of information which heterogenic in knowledge acquisition (Andryani, Negara and Darma, 2015), and so on.

There are many well-developed ontology modelling methodologies, but as said, there is no single unified methodology for developing ontologies. The application of methodologies to developing ontologies depends on the different developing software and different scenarios that arise in the applications, but the fundamental development principles and mechanics are the same and the development processes have lots of similarities. Based on the vast review of the current ontology development methodologies, the SKEM (7-Steps) method from ‘‘Ontology Development 101’’ (Noy and McGuinness, 2001) is applied in this research to develop the proposed ontology model, and the reasons for choosing this method are given in Chapter 3. The processes of the developed ontology in this research are illustrated step-by-step in Chapter 3.

2.8 Ontology applications in the AEC industry

As the core technology of the Semantic Web, ontology has been widely used in knowledge engineering, artificial intelligence, and computer science. The functionalities of ontology have shown advantages in knowledge representation, sharing, reusing, retrieval, inference, etc. Currently, the practical applications of ontology cover many fields, such as the bio-informatics and molecular biology domains (Stevens, Goble and Bechhofer, 2000), education domain (Sim and Brouse, 2014; Yee-King *et al.*, 2019), health management (Polenghi *et al*., 2021), economics domain (Yoo and No, 2014), customer needs exploration and management (Chen, Chen and Leong Kah, 2011; Chen *et al.*, 2013; Oriţă, Drăghici and Beney, 2013), manufacturing industry (Gecen, 2017), e-commerce field (Gómez-Pérez, Fernández-López and Corcho, 2004), product design (Zhang, Hu and Xu, 2010; Abadi, Ben-Azza and Sekkat, 2018), decision support systems development (Smirnov *et al.*, 2021), mechanical engineering (Ma and Tian, 2014), aerospace industry (Sanya and Shehab, 2014, 2015), the Architecture, Engineering and Construction (AEC) industry (Pauwels and Terkaj, 2016; Pauwels, Zhang and Lee, 2017; Zhao *et al.*, 2020; Zhao and Yang, 2021; Zheng, Törmä and Seppänen, 2021), impact of Covid-19 on the banking sector (Patel *et al.*, 2021), and so on.

As briefly reviewed above, many domains have adopted ontology technology for domain knowledge management. In general, as a Semantic Web technology that can achieve interoperability between humans and machines, ontology has been taken as an effective method for problem-solving, knowledge sharing, retrieval, and inference, especially in the Architecture, Engineering and Construction (AEC) industry.

The ontology applications in the AEC industry can be subdivided into different fields to be discussed, including project management, compliance checking (Zhong *et al.*, 2012, 2018; Li *et al.*, 2021), job hazards analysis and risk management (Tserng *et al.*, 2009; Wang and Boukamp, 2011; Zhang, Boukamp and Teizer, 2015; Zhong and Li, 2015; Ding *et al.*, 2016; Pruvost and Scherer, 2017), building performance analysing (Jin *et al.*, 2019; Lork *et al.*, 2019; Wolosiuk and Mahdavi, 2020; Zhao and Yang, 2021), safety management (Gangolells and Casals, 2012; Zhang, Boukamp and Teizer, 2014, 2015; Lu *et al.*, 2015), structural design (Brandt *et al.*, 2008; Hou, 2015; Hou, Li and Rezgui, 2015), sustainability development (Konys, 2018), etc.

The following sub-sections present more details on ontology applications in different AEC industry domains.

2.8.1 Ontology application in regulation knowledge management and compliance checking

Regulation compliance checking is an essential process in the construction industry, which allows the operations to follow relevant standards to ensure construction quality and safety. However, the traditional manual construction quality compliance checking process is time-costing, cumbrous, and error-prone. As OWL ontology has the advantages of knowledge modelling and logical reasoning, it has been widely used to systematically model the compliance checking knowledge in various construction regulations, in that, the specific regulations knowledge can be transferred into the OWL ontology model, the SWRL language can be used to formulate compliance checking rules, and the query languages of SPARQL or SQWRL can be used to retrieval and infer formulated compliance checking knowledge.

Bouzidi *et al.* (2011) converted the document-based regulatory texts of the photovoltaic field into the semantic-based SPARQL rules to automatically model and query the compliance checking. Hu *et al.*(2013) explored the application feasibility of the semantic ontology method in construction regulation constraints and quality conformance checking field, the construction quality conformance checking constraints are modelled into axioms/OWL and SWRL rules in a computer-interpretable ontology model.

Zhong *et al.* (2012) proposed the CQIEOntology (Construction Quality Inspection and Evaluation Ontology) for automated construction quality inspection and evaluation, in which, the constraints of the regulation are modelled into OWL axioms and SWRL rules. Moreover, it allows construction quality inspection to be carried out concurrently with the construction process, rather than as an afterthought. This ontology model has been demonstrated and verified through a case study based on regulation examples taken from a national construction quality compliance checking standard. But this ontology does not cover the OWL knowledge querying by SPARQL or SQWRL. Zhong cooperated with some other researchers to do some similar compliance checking research. In 2015, Zhong and Li (2015) proposed a semantic method for construction risk and safety management, which integrated the construction risk and safety management into the construction process to facilitate the risk management as a paralleling function to the construction process rather than an afterthought. However, this research mainly focuses on illustrating the feasibility of their proposed approach through building a simple construction process and risk management model which only includes a simple construction process and limited types of risk. Only the construction technical risks are included in their developed model but not other types of risk, such as the material delivery risk. In the same year, Zhong *et al.* (2015) conducted another study in the construction plan domain, developing an ontology-driven semantic approach to defining technical plan and verifying it automatically in construction. In 2018, Zhong *et al.* (2018) combined the BIM (Building Information Modelling) technique and ontology to develop a framework for building environmental monitoring and compliance checking. There are four specific ontologies in this framework, they are building information ontology, SSN (Semantic Sensor Network) ontology, building regulation ontology, and building environmental monitoring ontology. This framework promotes the effective interoperation of monitoring systems and information sharing between stakeholders. One year later, in 2019, Zhong *et al.* (2019) conducted a scientometric analysis and critical review of existing construction-related ontology studies collected from the Scopus database. This overview research is very valuable as its scientometric analysis results provide the emerging trends of future research for this domain researchers, for example, the research fields of deep-learning-based ontology information extraction for automated compliance checking field or automatic generation of domain ontology from documents, etc.

Beach *et al.* (2015) presented a rule-based semantic method for automated regulatory compliance in the construction domain. Zhang and El-Gohary (2015, 2016a; 2016b) conducted a few studies in the construction automatic regulatory compliance checking domain, they introduced a semantic and rule-based Natural Language Processing (NLP) approach and machine learning techniques to automatically extract information from construction regulatory textual documents in a human-like manner, which supports the automatic compliance checking in the construction process.

Construction safety related knowledge and specific information are scattered and fragmented. To organize, store and re-use construction safety knowledge, Zhang, Boukamp and Teizer (2014, 2015) explored the interaction between ontology and BIM in the automatic construction safety management domain, the proposed ontology prototype models support automated safety planning for job hazard analysis using BIM. The proposed prototype from their research supports a safety manager to plan for safety at the front-end of a project more efficiently. Xiao *et al.* (2018) developed the CSCOntology for construction safety checking knowledge, which identifies hazards before they occur and is a core process of safety management on construction sites. Wu *et al.* (2021) proposed a conceptual framework that integrated with the computer vision and ontology technology in the construction safety management domain, computer vision is used to extract information from images captured from the on-site environment, and ontology and SWRL are applied to code the relevant risk and safety management regulatory rules. By comparing the visual information extracted from the on-site environment images via the computer vision with the predefined safety regulatory SWRL rules, the potential hazards can be identified and the corresponding mitigation measures can be inferred, which is helpful for constructors to avoid the risk beforehand. By running this ontology, the inferred results show a high similarity to the accuracy of the manual safety checking pattern. However, the practical implementation of computer vision in the construction domain is still weak, as the construction site is a complicated environment, it is difficult for the computer vision algorithm to accurately detect information from on-site real-time images, moreover, only some simple SWRL rules has been coded in this ontology model, which means more rules for hazards should be encoded in the future.

The research from Li *et al.* (2021) has pointed out that most regulations compliance checking is still done by manual review, and even though ontology has been successfully used to achieve codes knowledge representation and automatic compliance checking, however, the construction of code ontologies is still completed manually by the experienced domain researchers, that may cause poor class hierarchy structure. So they proposed a semiautomatic code ontology construction method based on ifcOWL, which does not rely too much on domain experts' knowledge. First, extend ifcOWL to represent code information, and then the extended ifcOWL is converted into code ontology. In the end, a railway code ontology has been developed as experimental content to prove the feasibility of their method. Li *et al.* (2021) research has achieved semiautomatic generation of code ontology via hierarchical classification based on concepts in ifcOWL. But this code ontology developed in their research does not cover all the relationships, and some important compliance checking knowledge and relationships still need to be added to this code ontology to achieve a more complete compliance checking.

2.8.2 Sustainable building development domain

Another emerging application of ontology technology is the AEC industry sustainable development domain, such as the sustainable structural design considering low embodied energy and carbon (Hou, Li and Rezgui, 2015), the sustainable building technology (Tah and Abanda, 2011), sustainable construction strategies, sustainable or green building evaluation field (Jiang, Wang and Wu, 2018), and so on.

Abanda and Tah (2008) explored the feasibility of Semantic Web applications in the emerging sustainable building technology domain in the UK construction industry, the developed ontology named Sustainable Building Technology (SBT) Ontology. The top-level concepts of SBT ontology are classified into three classes: building construction technology, organization and standards. This SBT ontology work served as the groundwork for their further research in 2011, they developed a photovoltaic system ontology to represent photovoltaic system domain knowledge (Tah and Abanda, 2011).

Hou, Li and Rezgui (2015) developed the OntoSCS (Ontology for Sustainable Concrete Structure) prototype system for facilitating decision-making in the design process by offering optimised structural design solutions and selections of material suppliers, this system considered embodied energy and CO_2 as the selection indicators for sustainable structural design. This OntoSCS system applies an effective way of managing both structural design and building sustainability knowledge by combining ontology and SWRL rules. With the capability of simultaneously computing structural load capacity as well as embodied energy and carbon, it could eventually assist structural engineers to understand the environmental impact of their designs and make better decisions in the early design stage of buildings.

Howsawi and Zhang (2017) proposed an ontology to provide the potential sustainable construction strategies for energy saving in building construction and operation stages, to promote sustainable construction development in Saudi Arabia. Konys (2018) developed an ontology-based knowledge management method for the sustainability assessment domain, which enables the reuse and interoperability of the sustainability assessment domain knowledge.

In the field of green building evaluation, the conventional green building evaluation is conducted after the design is complete or after the building has been occupied for some time. Nowadays, most building evaluation tasks are conducted in a manual way, which is time-consuming and error-prone. Jiang, Wang and Wu (2018) developed an approach to facilitate intelligent green building evaluation by combing BIM and ontology to facilitate the process of green building evaluation. In their research, a green building evaluation ontology (GBEOntology) is developed to formalize and represent fragmented knowledge include in evaluation standards, while BIM is used for extracting the related building information and integrating the information into the ontology model. In their research, a green building evaluation ontology (GBEOntology) is developed to formalize and represent fragmented knowledge included in evaluation standards, while BIM is used for extracting the related building information and integrating the information into the ontology model. The development of their proposed method includes text knowledge acquisition from the related green building evaluation standards, evaluated building BIM information extraction, and ontology building and rule-based reasoning. This proposed evaluation approach enables the structural representation and reuse of green building evaluation standard knowledge that can save time and manual efforts. However, this work does not cover all the information involved in the green building standards, and it only validates the feasibility of the proposed approach for the design stage evaluation, but not the operation stage. Moreover, some needed building evaluation data cannot be obtained directly from BIM, which leads to some extra manual work. So, this proposed green building evaluation approach is not fully automated and still needs manual input.

Different from the traditional method, Zhang *et al.* (2019) developed an intelligent green building rating (iGBR) framework supported by the semantic and social method to enable the real-time green building evaluation rating in the building design process under the BIM-based design environment, which helps designers to identify more potential ways to optimise the final rating and achieve better sustainability in the design stage. As shown in Figure 2-19, the iGBR framework consists of four parts, Semantic Knowledge Representation, BIM-based Building Design, Rule-based Reasoning, and Social Involvement. Here, the ontology is used as a semantic technology to encapsulate the green building rating knowledge, and the related rating score calculation principles are encoded in a set of SWRL rules to support reasoning.

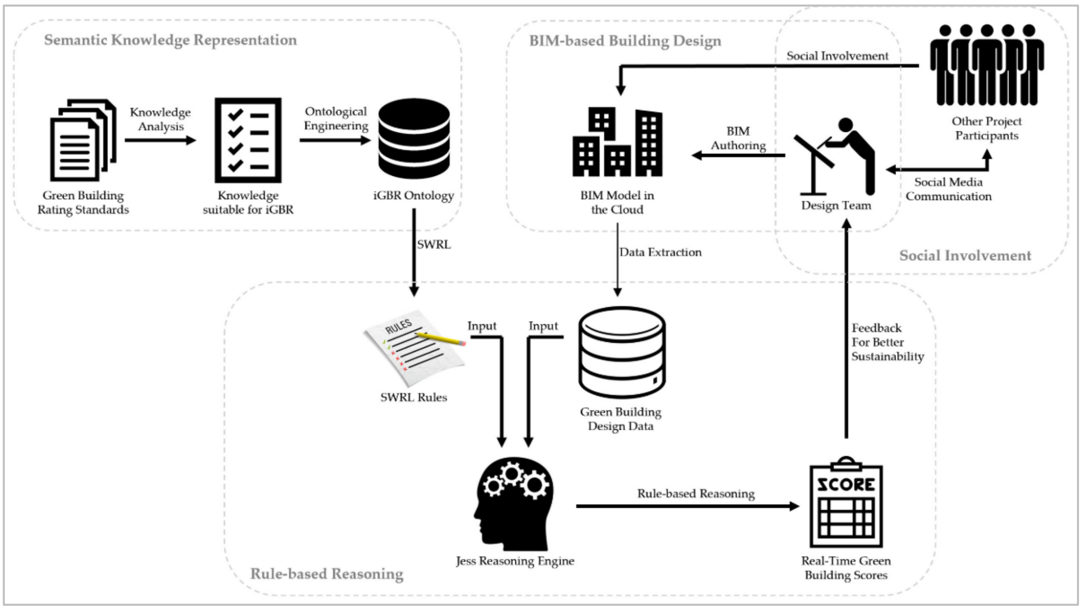


Figure 2-19 The iGBR framework (Source: (Zhang *et al.*, 2019))

However, this iGBR framework is not able to encode all articles of a green building rating system and can not give the complete score of a green building evaluation process, but only partial articles that can be represented in an ontology and expressed in SWRL rules. There are some other studies on the combination of ontology and BIM domain, for example, the cost estimation during the tendering stages (Abanda, Kamsu-Foguem and Tah, 2017), the expert system for temporal and spatial construction planning (Getuli, 2017), the eeBIM (Energy Enhanced BIM) ontology-based framework for building energy performance utilization domain, and it also can be used to identify the energy performance problems at the early design stage (Kadolsky, Baumgärtel and Scherer, 2014; 2015) etc. However, this eeBIM ontology framework focuses more on building energy performance simulations at the early design stage, not fitting the total building performance evaluation from the whole lifecycle of the building.

2.8.3 Building management systems and building performance assessment domain

The different building management systems (BMS) generate a large amount of data when running in buildings, and these data are stored in different formats systems, which has caused problems of scattered and fragmented data. The data heterogeneity leads to the failure of effective data exchange and interoperability between different building management systems. As the core technology of Semantic Web technology, ontology has been widely used in the field of building management systems (Simeone and Fioravanti, 2012; Wicaksono, Aleksandrov and Rogalski, 2012; Corry *et al.*, 2014) and building performance evaluation domain (Corry *et al.*, 2015; Hong, D'Oca, Taylor-Lange, *et al.*, 2015; Hong, D'Oca, Turner, *et al.*, 2015; Jin *et al.*, 2019; Lork *et al.*, 2019; Zhao and Yang, 2021) because of its powerful data exchange ability, rich semantic expression and integration capacity, intelligent query function, and other advantages.

The increasing complexity of modern building automation systems (BAS) makes challenges to efficient building management. Many research initiatives developed applications to support building performance analysis and optimization leveraging Semantic Web technologies. These building performance analysis studies typically focus intensively on the design stage and the operation stage of buildings. Wicaksono, Aleksandrov and Rogalski (2012) adopted ontology technology to structure a framework to analyse the relations between the energy consumption of technical equipment of buildings, occurring activities in buildings, and relevant environmental factors to improve energy efficiency in building automation systems. Dibley *et al. (2012)* developed an intelligent sensor-based ontology (OntoFM) to support real-time building monitoring and improve the intelligence of applications in the facility management domain. In 2017, Mahdavi and Taheri (2017) introduced an ontology to represent and incorporate the multiple layers of monitored building data. The monitored data are classified into six categories of inhabitants, indoor and external environmental conditions, control systems and devices, equipment, and energy flows. The UML graphic method is adopted in this research to help structure the various monitored data instances and define relationships between instances. The developed ontology supports the data acquisition, storage, and processing in multiple systems, and it can be used to achieve operational optimization of existing facilities and provide the optimization for future designs. According to the research from Schneider, Pauwels and Steiger (2017), the heterogeneous information about the control logic of building automation systems (BAS) is normally stored using different data formats, to provide a shared common understanding of control logic information, they proposed a novel CTRLont ontology, which formally specifies the control logic in BAS. The explicit concepts and relationships of control logic were represented in OWL ontology, and a rule-based verification approach was applied in the end to demonstrate the feasibility of this model. The developed CTRLont ontology can be integrated with other existing building management ontologies to help information exchange and interoperability among different systems, and improve the performance of technical equipment throughout the lifecycle of BAS.

Hong, D'Oca, Turner, *et al.* (2015) pointed out that the behaviour and activities of occupants are pivotal factors affecting the efficient utilization of energy in buildings. In their first study (Hong, D'Oca, Turner, *et al.*, 2015), an ontology-driven DNAS ('Drivers-Needs-Actions-Systems') framework is developed to represent the occupant behaviours that would directly or indirectly impact energy consumption in buildings. This standardized framework provides a shared common understanding model to the relevant domain researchers to observe, model, and simulate energy-related occupant behaviour in buildings. The second part of their study implemented the DNAS framework using an XML (Extensible Markup Language) schema, namely the obXML ('occupant behavior XML'), for supporting the development of occupant information modelling and integration with building simulation tools (Hong, D'Oca, Taylor-Lange, *et al.*, 2015).

Throughout buildings' life-cycle, they produce vast quantities of data. However, many buildings do not perform as originally intended because of the vast heterogeneous data. To make the performance gap explicit between simulated and observed building performance, Corry *et al.*(2015) developed a performance framework (PF) ontology to describe how heterogeneous building data sources can be transformed into semantically rich information. Figure 2-20 illustrates the structure of concepts included in the PF ontology and how this PF ontology could be integrated with other existing ontologies, such as the ifcOWL, the SimModel (Simulation Domain Model), and the SSN ontology. In this study, they conducted a demonstration of the building thermal comfort to illustrate how building performance can be assessed more effectively using integrated data sets.

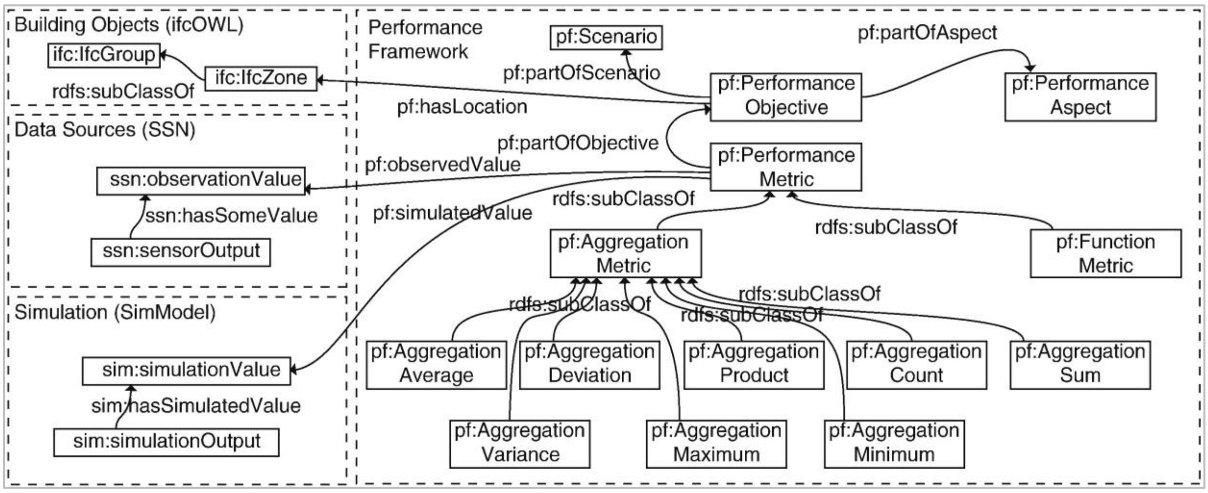


Figure 2-20 Performance framework ontology integrated with existing ontologies (Source: (Corry et al., 2015))

Lork *et al.* (2019) developed an ontology-based building energy management framework which consists of three modules, namely, the building management system (BMS) module, benchmarking (BMK) module, and evaluation & control (ENC) module. The measured real-time building parameters are modelled in the BMS model, the BMK module uses the standard-based method to formalize the assessment benchmark knowledge from the relevant evaluation standards, and the ENC module evaluates the collected real-time building consumption data from the BMS module and the predicted energy consumption assessment thresholds from the BMK module. Compared with traditional energy management schemes, the proposed integrated framework has achieved significantly better results with an average of 8.7% energy-saving improvement in the case study buildings. This ontology framework is flexible to be extended and integrated with different building systems or other existing ontologies.

Mahdavi, Taheri and Wolosiuk (2019) developed the Building Performance Indicators (BPI) ontology by capturing the essential indicators from the thermal, air quality, visual, and acoustical building performance domains. The attributes and properties of the captured building performance variables are specified in this ontology. The feasibility of the BPI ontology has been demonstrated by the visualization application. This proposed BPI concepts ontology can be used as a base conceptual model for future studies in building performance assessments or other related domains.

One of the advantages of ontology is knowledge and data exchange in specific domains. Based on this advantage, Wolosiuk and Mahdavi (2020) presented the development process of the building performance data (BPD) ontology, which captures and categorizes the vast heterogeneous building-related performance or simulated data and its properties for subsequent integration and collaboration in a variety of building performance evaluation engineering applications.

Pritoni *et al* (2021) conducted a systematic review of the existing metadata schemas and ontologies for building energy applications and concluded that it is difficult to achieve collaboration between systems and existing ontologies due to the lack of semantic interoperability between different applications. This research provides modellers with a common understanding of a comprehensive repository of metadata schemas that can help reuse existing schemas. As shown in Figure 2-21, the authors had a deep analysis of 5 particular existing ontologies, the SAREF (orange), SSN/SOSA (purple), Brick (blue), BOT (red), RealEstateCore (green), to illustrate significant differences between different schemas when building modelling for energy use cases.

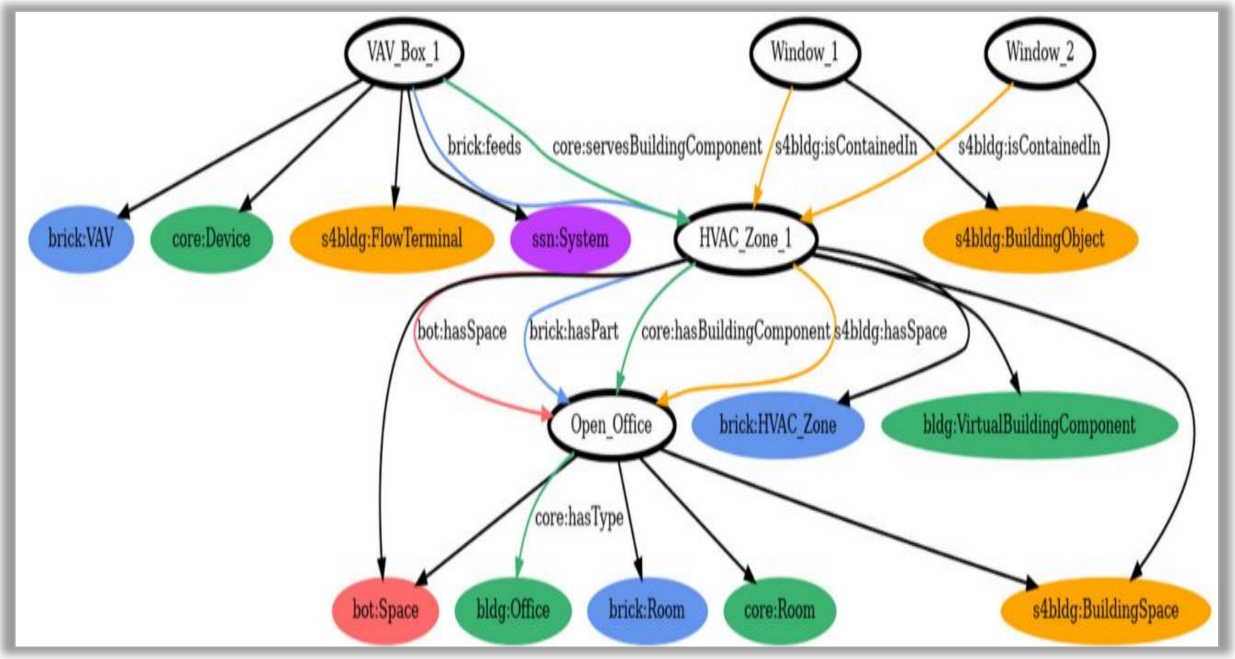


Figure 2-21 The representation differences and gaps between different schemas in model building (Source: (Pritoni et al., 2021))

This research well illustrated the difference and gaps in the way of building concepts and relationships between different schemas when building the application ontologies under three use cases in energy assessment, fault detection, and control optimisation. However, these three use cases focus on the building operation stage, which only represents a small part of the possible application scenarios. Future research should further cover other use application scenarios.

2.8.4 Smart buildings application

The application of ontological technologies provides more opportunities in improving the intelligence of buildings. The application of Semantic Web technologies in smart buildings aspect is analysed below.

- ifcOWL ontology

Industry Foundation Classes (IFC) is a standardized, digital description of the AEC industry, which is an open international standard (ISO 16739-1:2018), now managed by the buildingSmart International organization (Pauwels and Terkaj, 2016). It specifies a data schema and an exchange file format structure to improve the sharing of information among different applications used by the various stakeholders and facilitate interoperability in the AEC industry (Beetz, van Leeuwen and De Vries, 2009). The IFC data model is avaiable in EXPRESS schema and XSD schema (Pauwels *et al.*, 2017). It has three fundamental classes: the IfcObjectDefinition, the IfcRelationship, and the IfcPropertyDefinition. Each of these classes has its subclasses, as shown in Figure 2-22.

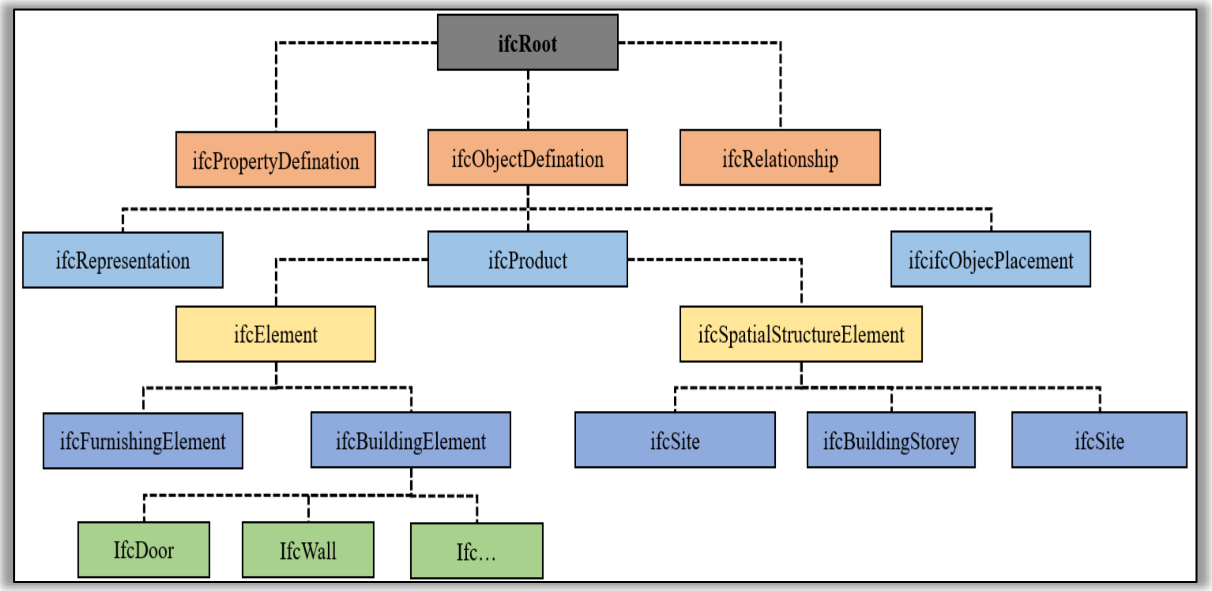


Figure 2-22 IFC model structure

The ifcOWL ontology provides a Web Ontology Language (OWL) representation of the IFC data model (Pauwels *et al.*, 2017), that allows the building data can be easily integrated with other data in the AEC industry, like the material data, product supplier data, sensor data, and so on. The development of ifcOWL provides fundamental support for achieving data interoperability in different applications in the AEC industry and beyond. It has been widely applied in various studies. The recent research from González et al. (2021) adopted the ifcOWL ontology to represent the relevant concepts of the indoor navigation system by adding new terms, SWRL rules and SQWRL query into it to support the indoor navigation system.

But ifcOWL also has some limitations, for example not every concept of IFC schema has been converted into ifcOWL, and the scope of ifcOWL is mainly focused on the architecture engineering domain. So Li et al. (2021) have explored the application of ifcOWL for code ontology development in the rail engineering field. Based on the conventional ifcOWL ontology, Li et al. (2021) extended and converted the ifcOWL into code ontology, which ifcOWL is used to represent the code concepts in rail engineering.

- DogOnt ontology

The DogOnt is a novel modelling approach for the intelligent domotic environment (IDE) (Bonino and Corno, 2008), consisting of DogOnt OWL ontology and a set of DogOnt SWRL rules. The DogOnt ontology is designed to support device/network independent descriptions of houses, with a focus on interoperability between connected domotic applications. As shown in Figure 2-23, this ontology contains 5 main categories: Building Thing, Building Environment, State, Functionality, and Domotic Network Component.

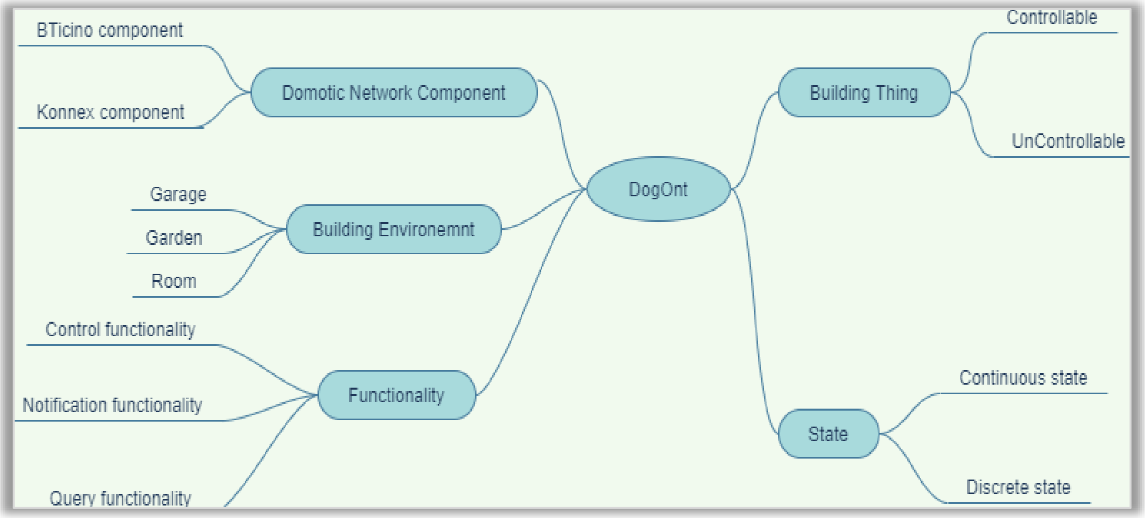


Figure 2-23 The overview of DogOnt ontology (Bonino, Castellina and Corno, 2008)

- ThinkHome ontology

The ThinkHome system consists of two main parts, a comprehensive knowledge base ontology and a multi-agent system, which was designed to optimise energy efficiency and user comfort. The ThinkHome ontology represents all relevant concepts that are related to the optimization of energy-efficient and intelligent control mechanisms in a smart home, including building automation services, system process, building layout and materials, users activities, environmental impact, thermal comfort, climate, etc (Reinisch *et al.*, 2011).

- SSN ontology

The Semantic Sensor Network (SSN) ontology, developed by the W3C Semantic Sensor Network Incubator group, to 'describe sensors and their observations, involved procedures, the characteristics of interest, samples used to do so, observed properties, and actuators'. The development of the initial SSN ontology was based on a Stimulus Sensor Observation (SSO) ontology design pattern (Compton *et al.*, 2012; Haller *et al.*, 2017). To streamline the ontologies, a lightweight ontology, the Sensor, Observation, Sample, and Actuator (SOSA) ontology was developed as a replacement of the SSO core to provide a light-weight central building block for the most recent version of SSN ontology, and it can be used standalone (Janowicz *et al.*, 2019).

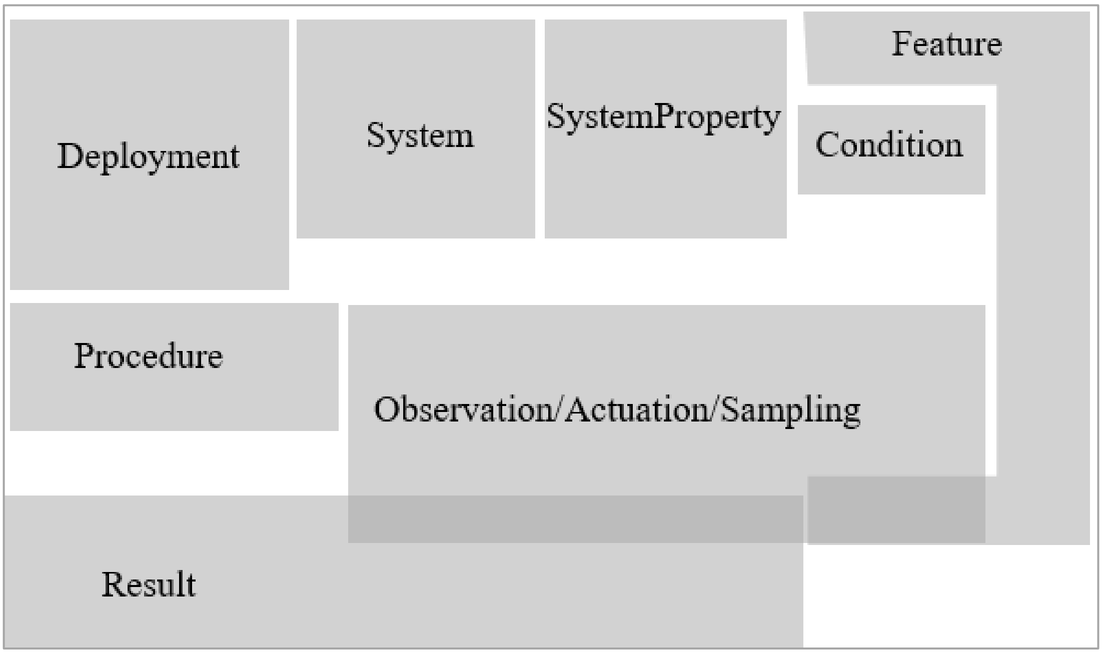


Figure 2-24 Overview of the SOSA/SSN ontology modules

Figure 2-24 shows the conceptual ontology models in SSN and SOSA. By following this pattern, the main classes and properties inside the ontology modules from the Observation perspective can be seen in Figure 2-25, where, the SOSA-related concepts and properties are shown in green, while SSN-only components are shown in blue.

Different from the original SSN and SSO, the scope of SSN ontology and SOSA ontology is not limited to sensors and their observations, but also includes the classes and the corresponding properties for actuators and sampling. The SOSA provides a flexible but coherent pattern to represent entities, relationships, and activities involved in sensing, sampling, and actuation (Haller *et al.*, 2017), and they are the most frequently cited ontologies in the Internet of Things (IoT) domain (Pritoni *et al.*, 2021).

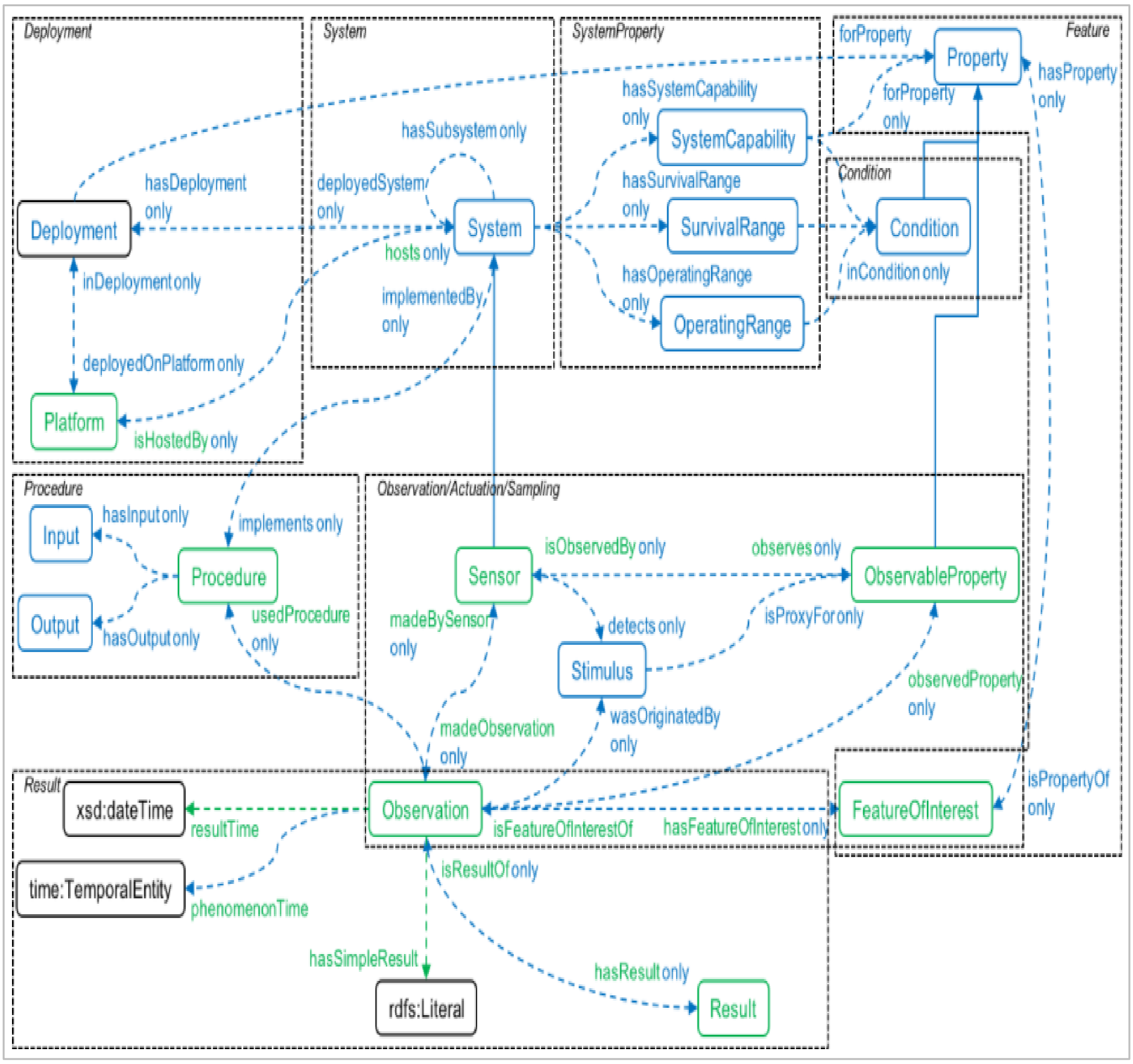


Figure 2-25 Overview of the SSN classes and properties in observation (Haller et al., 2017)

- BOnSAI ontology: Smart Building Ontology for Ambient Intelligence

Based on the methodology of the Ontology Development 101 Guide (Noy and McGuinness, 2001), Stavropoulos *et al.* (2012) developed the BOnSAI ontology to incorporate ambient intelligence in Smart Buildings, which modelled the service-oriented concepts of the ambient intelligence systems. By directly importing from some existing ontologies, like CoDAMoS ontology (Preuveneers and Berbers, 2005) and OWL-S upper ontology, the concepts of BOnSAI were extended to include Functionality (services, operations, inputs, outputs, environmental conditions, etc), Context (location, room temperature, etc.), Quality of Services (service latency, response time, etc.), Hardware (devices, sensors, actuators, etc), User Preferences (moods, preference profile, etc.).

Service interoperability provided by the BOnSAI ontology enhances the development of service-based intelligent systems in smart buildings. But this ontology does not describe the relations between the sensors and other building assets, and there is a missing description of some common functional building intelligent systems, like the HVAC or lighting system (Balaji *et al.*, 2018).

- SBOnto ontology: Smart Building Ontology

The SBOnto (Smart Building Ontology) is an ontology to formalize the basic knowledge of the smart building. It describes the general rules of smart buildings and consists of 6 main classes, the Device (sensors, actuators, etc.), State (e.g. On/Off), Architecture (building components, e.g. windows, doors, walls), Environment (Garden, Bedroom), Furniture, Network (Wired and Wireless), as shown in Figure 2-26. Different from other ontologies which are limited to specific building types, SBOnto ontology aims to be applicable to any type of building, whether residential or commercial buildings (Žáček and Janošek, 2017).

- SmartHome Ontology:

Along with the application of advantaged IoT technologies in smart homes, a large amount of heterogeneous data have been generated by the variety of smart home applications. To this end, a general domain ontology has been developed by Tao, Ota and Dong (2017) to enable high-efficiency data management and application. A general ontology model and a data semantic fusion model are designed. The general ontology model is designed to formalize domain knowledge with the top-level ontology concepts of User, Applications, Service, Home device, Technology, and the data semantic fusion model, as shown in Figure 2-27, is designed to effectively data update and query with four layers, Data Space Adaptation Layer, Ontology Description Layer, Semantic Processing Layer and Application Service Layer.

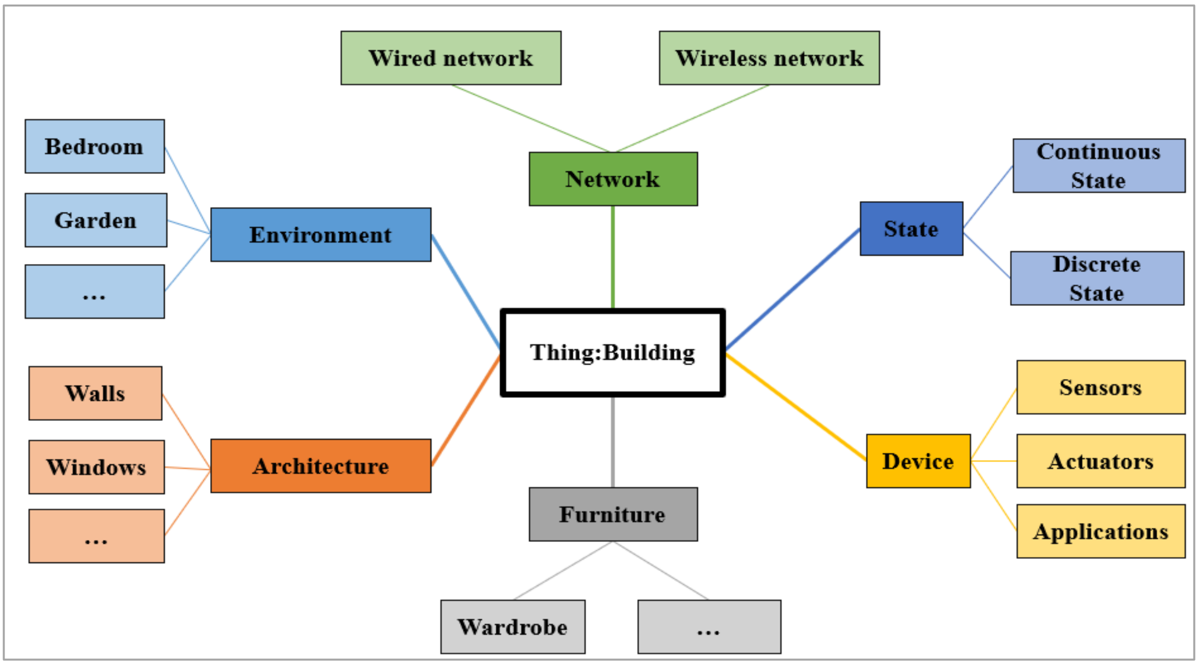


Figure 2-26 The main structure of SBOnto ontology

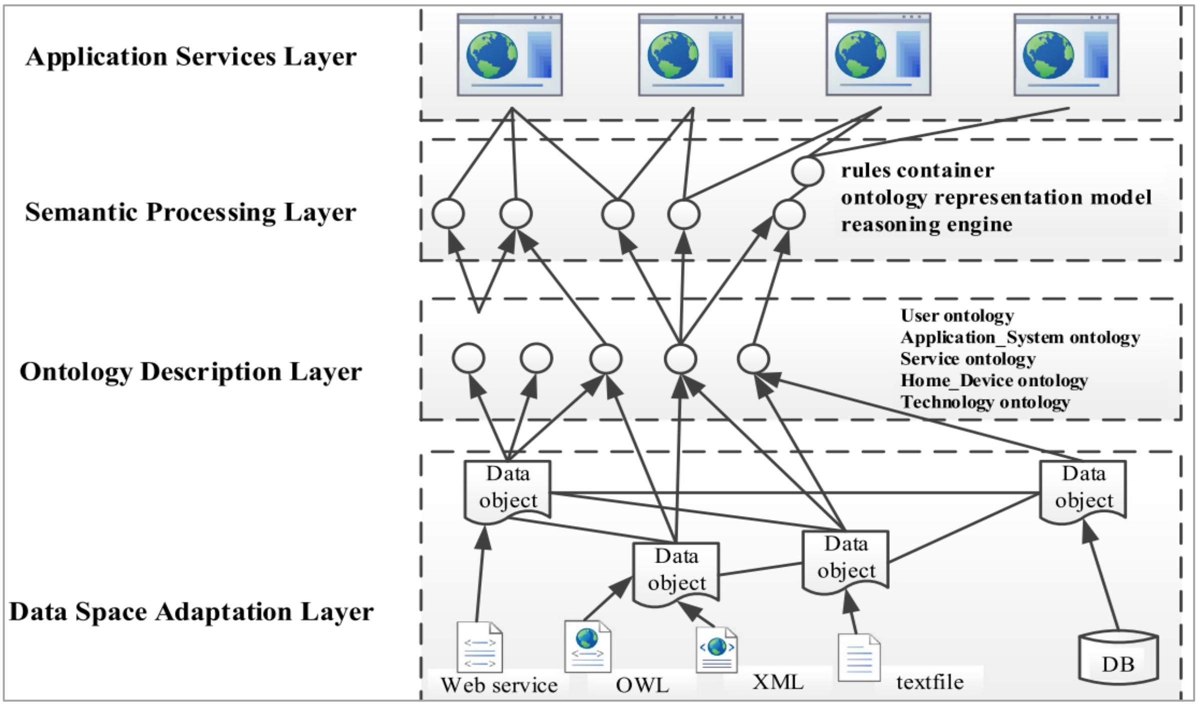


Figure 2-27 Designed data semantic fusion model (Source: (Tao, Ota and Dong, 2017))

- SAREF: Smart Appliance REFerence ontology

The SAREF ontology (Smart Appliance REFerence ontology), which was created 'as a shared model of consensus to facilitate the matching of existing assets in the smart appliances domain', aims to enable semantic interoperability for smart appliances (Daniele, den Hartog and Roes, 2015). It specified the core concepts in the smart appliances domain, as well as the relations between them. The basic structure of SAREF ontology is shown in Figure 2-28. The development of SAREF was based on the fundamental principles of reuse and alignment, modularity, extensibility, and maintainability.

Since many studies have been already done in the smart appliances domain, there is no need to establish new ontology concepts for this domain, so the SAREF was developed from the reuse and alignment of concepts and relationships that are defined in existing assets (Daniele, den Hartog and Roes, 2015). The modularity of SAREF allows the separation and recombination of different modules of the ontology depending on specific requirements. The good extensibility of SAREF allows it to be widely extended into some other domains, for example, the SAREF4ENER for the Energy domain, the SAREF4ENVI for the Environment domain, the SAREF4BLDG for the Building domain, the SAREF4CITY for the Smart Cities domain (ETSI, 2019b), the SAREF4INMA for Industry and Manufacturing domains (ETSI, 2019c), the SAREF4AGRI for Smart Agriculture and Food Chain domain (ETSI, 2019a), etc., maintained by the ETSI (European Telecommunications Standards Institute) (ETSI, 2020).

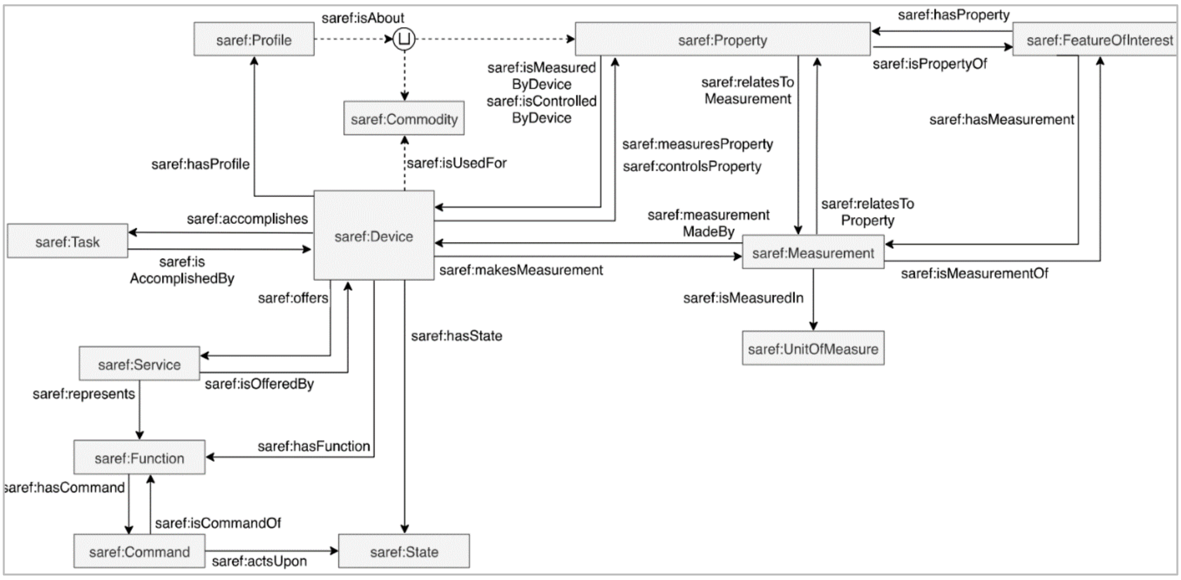


Figure 2-28 Overview of the SAREF ontology

- DNAS Framework

Hong, D'Oca, Turner, *et al.* (2015) pointed out that the behaviour and activities of occupants are pivotal factors affecting the efficient utilization of energy in buildings. In their first study

(Hong, D'Oca, Turner, *et al.*, 2015), an ontology-driven DNAS ('Drivers-Needs-Actions-Systems') framework is developed to represent the occupant behaviours that would directly or indirectly impact energy consumption in buildings. Drivers represent the environmental factors that trigger the occupant's behaviours or activities. Needs are the occupant's requirements that must be met to ensure they are satisfied with their environment, these requirements are categorized into Physical and Non-physical sub-elements. Actions are the interactions with systems or activities performed by occupants to meet their requirements. Systems stand for the devices or mechanisms within a building with which the occupants can interact to achieve environmental comfort satisfaction. Each of these four components has several sub-categories. Six energy-related mechanisms from the literature are described in the DNAS ontology: window opening and closing, shade and blind operation, lighting system control, thermostat and HVAC adjustment, electrical equipment usage, space occupancy. This standardized framework provides a shared common understanding model to the relevant domain researchers to observe, model, and simulate energy-related occupant behaviours in buildings.

As shown in Figure 2-29, the second part of their study implemented the DNAS framework into an XML (Extensible Markup Language) schema, the obXML ('occupant behavior XML') schema, for supporting the development of occupant information modelling and integration with building simulation tools (Hong, D'Oca, Taylor-Lange, *et al.*, 2015). It is also the first schema that provides a standardized data model to describe occupant behaviours with a focus on energy simulation (Pritoni *et al.*, 2021). Later on, they developed the occupant behaviour models to improve a robust and interoperable integration with multiple building performance simulation programs (Hong *et al.*, 2018; Putra, Hong and Andrews, 2021).

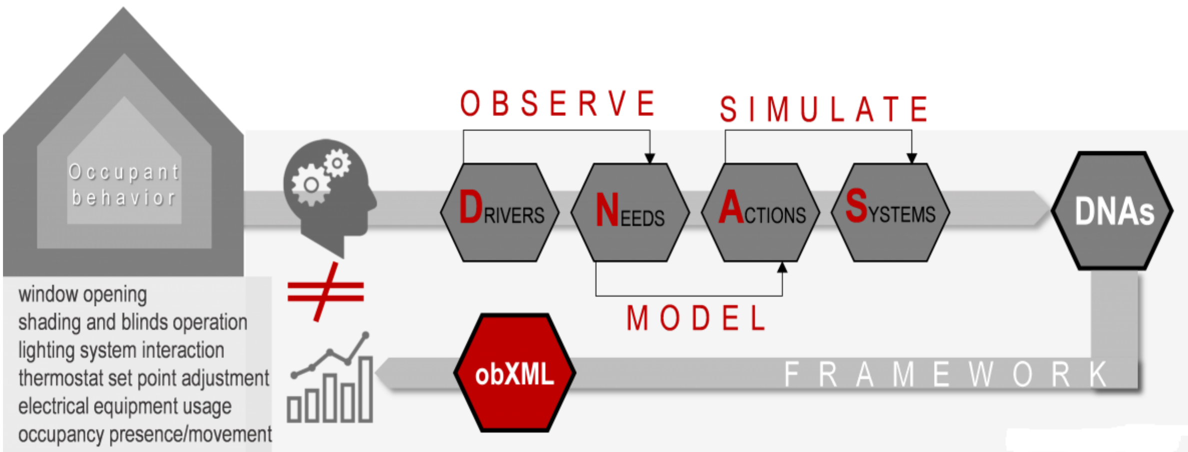


Figure 2-29 DNAS framework (obXML)(Source: (Hong, D'Oca, Turner, *et al.*, 2015))

Several studies on the application of ontology in human behaviours with a focus on the energy impact in buildings have been reviewed below.

- Onto-SB: Human Profile Ontology for Energy Efficiency in Smart Building

The Onto-SB (Human Profile Ontology for Energy Efficiency in Smart Building) ontology is a domain ontology for smart building (Degha *et al.*, 2018), which aims to contain maximum contextual information on the smart building domain and human profile. Following the ontology reusing development principles, some important concepts were imported from existing ontologies, for example, the concepts of the human profile, environment parameters, services, appliances, place, energy sources concepts, profile, etc. The relations between the main concepts in Onto-SB are shown in Figure 2-30.

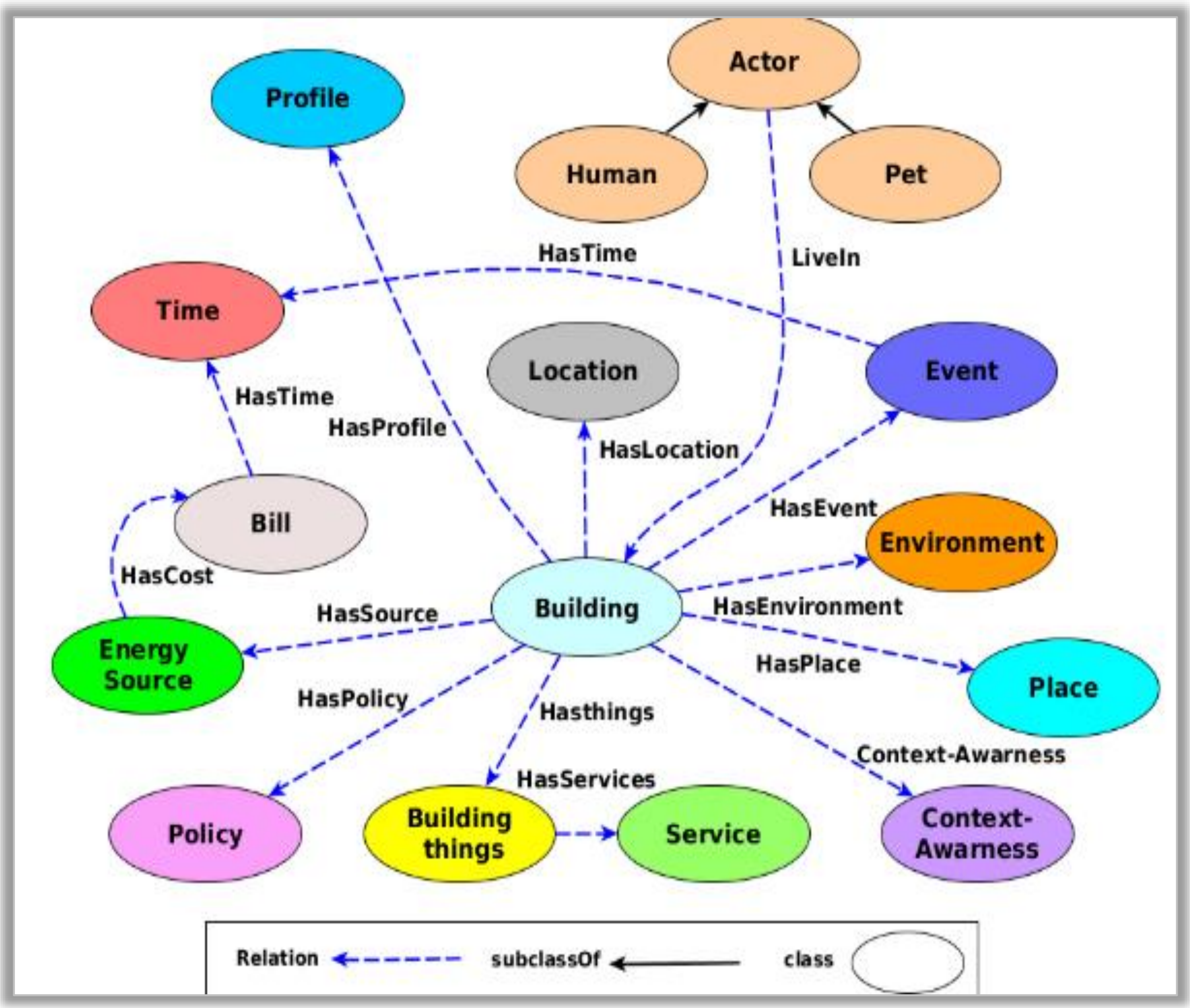


Figure 2-30 Main concepts relations in Onto-SB (Source: (Degha et al., 2018))

This ontology also developed a series of energy efficiency and inhabitant comfort optimization-related SWRL rules based on Onto-SB to enable reasoning using in Onto-SB ontology. For example, the following SWRL rule in Table 2-1 stated that based on the value from the temperature sensors to switch off the HVAC device when the temperature value is at a certain value range.

Table 2-1 Onto-SB ontology SWRL rule example

TemperatureSensor(?t) ^ Devices-Value(?t, ?val) ^ swrlb:greaterThan(?val, 25) ^ DeviceLocateInPlace(?z, ?t) ^ HeatingDevice(?h) ^ DeviceLocateInPlace(?z , ?h) -> Device-State(?h, "off").

This ontology provides the idea of developing the energy-saving and occupant comfort-optimization related SWRL rules, but this research still needs to develop more rules to exploit maximum user profile concepts with a focus on energy saving in smart buildings.

- OnCom (Orozco *et al.*, 2019)

The OnCom refers to a Wireless Sensor Network (WSN)-based thermal comfort measurement approach, which is developed based on ontology technology and the emotional state analysis of occupants, it aims to optimise the occupant's thermal comfort in smart buildings (Orozco *et al.*, 2019). S

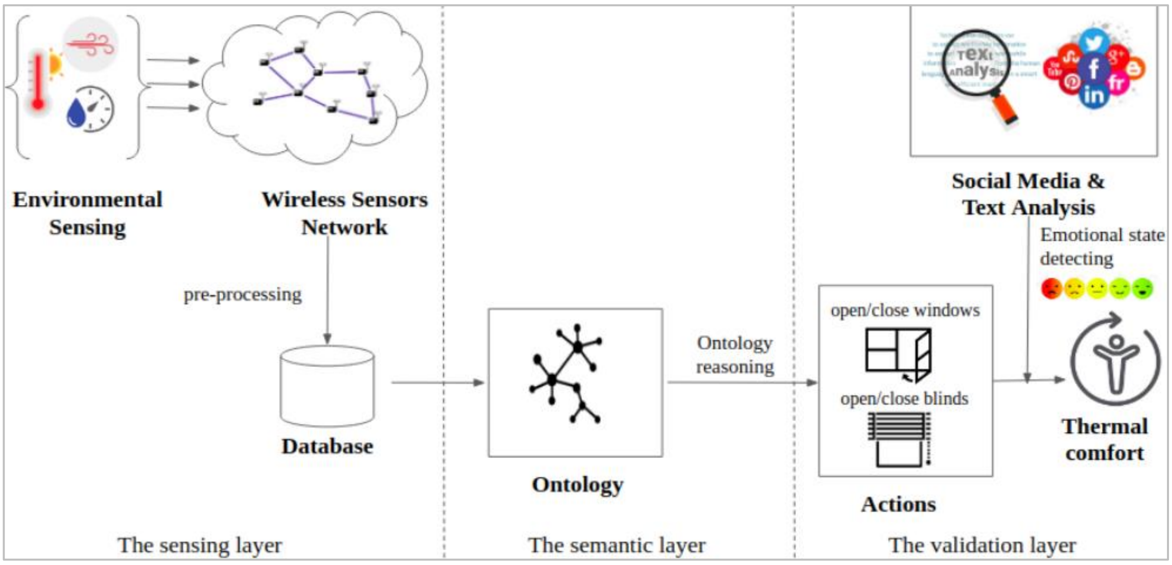


Figure 2-31 OnCom System (Source: (Orozco et al., 2019))

As shown in Figure 2-31, a three layers system (sensing layer, semantic layer, and controller layer) has been designed to manage the thermal comfort of the indoor environment. The first principle for building ontology is to consider reusing the existing ontologies, in this research, the authors adopted the existing ontologies of ‘sosa’, ‘event’, ‘geo’ and ‘qudt’ to describe the observation and actuation in this system.

The authors developed some inference SWRL rules into this system so that the system can automatically execute some commands to reach the indoor thermal comfort level in real-time, such as close or open windows, turn on or off lights, close or open blinds, etc. For example, the following SWRL rule in Table 2-2 explains one of the pre-defined rules in this research, it can be read as if the indoor environment has monitored light parameters value is lower than the satisfactory value then turn the light on:

Table 2-2 Sosa ontology SWRL rule example

sosa:Observation(?x) ^ oncom:EnviromentsSpace(?en) ^ sosa:hasFeatureOfInterest(?x,?en) ^ qudt-1-1:QuantityValue(?r) ^ sosa:hasResult(?x, ?r) ^ oncom:LUM(?r, ?c) ^ oncom:ObservableProperty(?obp) ^ oncom:LMIN(?obp, ?lmin)^swrlb:lessThan(?c,?lmin) ^ sosa:Actuation(?a) ^ sosa:hasLocation(?a, ?en) ^oncom:Lights(?lig) ^ sosa:hasFeatureOfInterest(?a, ?lig) ^ oncom:ParametersOutside(?po) ^ oncom:LUM(?po, ?lo) ^ swrlb:lessThan(?lo, ?lmin) →oncom:TurnON(?a, true) ^ oncom:ACTIVATE(?lig, true)

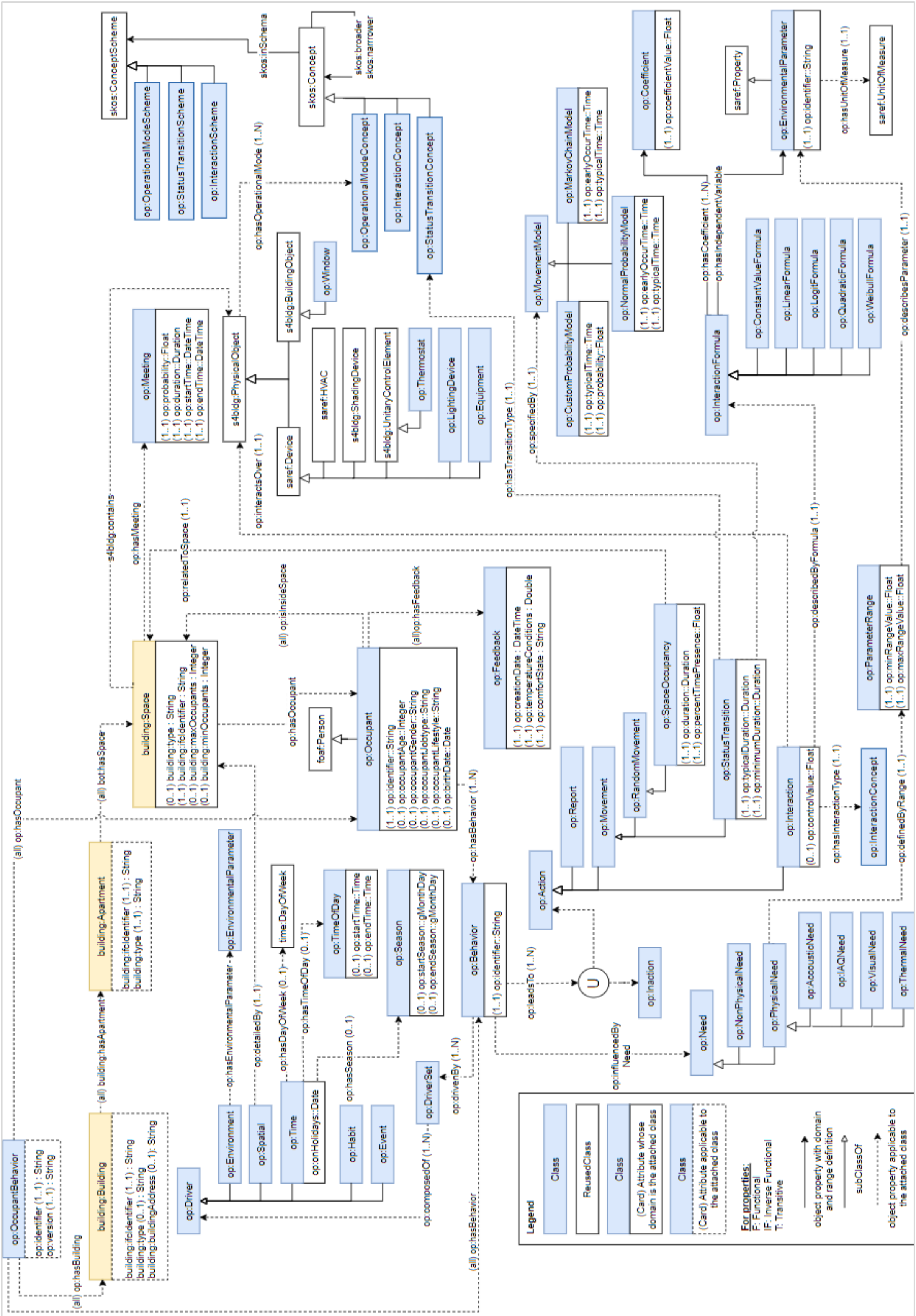


Figure 2-32 Occupancy profile ontology (Source: Chávez-Feria et al. (2020))

- Occupancy Profile (OP) Ontology

Based on the obXML (Occupancy Behavior XML) schema (Hong, D'Oca, Turner, *et al.*, 2015), Chávez-Feria *et al.* (2020) developed the Occupancy Profile (OP) ontology, as shown in Figure 2-32, to represent the occupant behaviours and activities inside buildings with a focus on the energy impact of their actions (Chávez-Feria, Poveda-Villalón and García-Castro, 2020). Instead of creating new classes, the authors resued the concepts available in SAREF ontologies, to allocate with the systems adopted from the obXML schema (Chávez-Feria, Poveda-Villalón and García-Castro, 2020).

The application of ontology technology in human behaviours that are reviewed, most of the existing literature however is focused on the development of a common abstraction language for sensors and energy-related concepts. There are still lots of research examples in the application in the construction domain, for example, the construction noise control domain (Xiao et al., 2018), the building construction cost estimation domain (Liu, Li and Jiang, 2016), delay analysis in construction (Bilgin, Dikmen and Birgonul, 2018). These above-reviewed ontologies are specialized in AEC industry domains with different focuses, for example, the knowledge representation and management of the AEC industry, sustainability development, human behaviours analysis with the focus on energy impact, and so on. However, the application of ontology is not just limited to the AEC industry, the advantages of data interoperability and flexible data exchange capabilities have promoted the wide application of ontology in other fields. For example, recent research from Patel *et al.* (2021) has proposed a Covid-19 Impact on Banking Ontology (Covid19-IBO) to examine the impact of Covid-19 on the performance of the Indian banking sector and gives initiatives at both the tactical and the strategic levels.

2.9 Review summary of Semantic Web and ontology development

The Semantic Web technology is investigated in Section 2.5; ontology as the key technology of Semantic Web has been illustrated in Section 2.6 from the perspectives of the ontology definition, ontology languages, ontology query technology, ontology editors, and so on; the methodologies for building ontologies are compared in Section 2.7, after analysing the advantages and disadvantages of each method, the '7-Steps' methodology from the Noy and McGuinness (2001) is adopted in this research; in Section 2.8, the application of ontologies in different fields, especially the AEC industry domain, is reviewed in detail, the limitations of current studies are given as well.

As reviewed above, the Semantic Web technologies have the ability to overcome interoperability problems between humans and machines. An ontology represents knowledge in a machine-readable format and achieves the interaction not only between different machines but also between machines and people. Ontology plays a key role in the knowledge management engineering (KEM) domain, not only in knowledge representation, but also in the fields of knowledge sharing, reuse, and retrieval.

The AEC industry is a knowledge-intensive field, however, due to the poor interoperability of data, the knowledge information in this industry is scattered and fragmented, it is time-consuming and error-prone to acquire information. Benefiting from the advantage of Semantic Web technology in knowledge conceptualization, ontology has been widely taken as an effective method for problem-solving, knowledge management, information representation and extraction, and logical inference in the AEC industry. Based on the critical review of Semantic Web technologies application in the AEC industry, several key findings are listed as follows:

- The application range of ontology technology in the AEC industry can be summarized in the following fields of the compliance checking domain, the safety and risk management, sustainability development domain, building management system and performance evaluation domain, smart building and applications, human behaviours in buildings analysis. And most of the ontologies were developed to represent structured knowledge or data from the domain of sensors systems or application systems.
- There are few human behaviours and profile ontologies, but these studies focus on developing lightweight ontologies to explore the user behaviours and profile concepts with the main focus on energy efficacy, but not a heavyweight ontology with rule-based reasoning.
- The emerging smart building assessment schemas information is heterogeneous and is represented in different data formats in different information systems. Knowledge sharing and semantic interoperability is therefore a literature gap for post-occupancy evaluation. The above-reviewed ontologies are specialized in different AEC industry domains, however, there is a missing ontology model for the post-occupancy evaluation

domain, especially the occupant satisfaction assessment in smart buildings. Instead of a taxonomy of technologies, the smart building assessment should be based on measurable performance criteria.

As analysed above, the application range of ontology covers various fields in the AEC industry. However, most of the existing ontologies in the AEC industry are lightweight ontologies that mainly focus on building a structured representation of the domain-specific knowledge or information, without delving into rule-based reasoning and query functions of the advanced Semantic Web technologies. Moreover, there is not a comprehensive POE ontology yet, especially within the building occupant satisfaction domain, which is the starting point of this research.

The Semantic Web approach offers great interoperability to integrate different knowledge, and therefore has been applied for framework development. The ontology not only has the ability of knowledge representation but also provides the reasoning and query functions. In this work, an ontology-based post-occupancy evaluation framework for smart buildings is introduced with the focus on developing an actual-performance-based smart building assessment ontology. The proposed heavyweight ontology model realizes the structural representation, sharing, interoperability and retrieval of fragmented assessment knowledge data, and achieves the automatic building assessment based on the SWRL rules and queries in the proposed ontology. The next section discusses the proposed ontology building methodology and processes.

Chapter 3 Methodology

This chapter presents the research methodologies of this study. The overview of the developed methodologies is illustrated in Section 3.1; the main OWL ontology knowledge base development methods are explained in Section 3.2; the POE-related SWRL rules and SQWRL query development methods are explained in Section 3.3 and Section 3.4; Section 3.5 is the summary of this chapter.

As analysed in Chapter 2, the application ranges of ontology cover various fields in the AEC industry. However, most of the existing ontologies in the AEC industry are lightweight ontologies that mainly focus on building a structured representation of the domain-specific knowledge or information, without delving into rule-based reasoning and query functions of the advanced Semantic Web technologies. Moreover, there is not a comprehensive POE ontology yet, especially within the building occupant satisfaction domain, which is the starting point of this research.

This research aims to develop an ontology-based post-occupancy evaluation framework dedicated to building performance evaluation with the ultimate aim of optimizing building operation guidelines and improving occupant use experience quality and well-being. A heavyweight OWL ontology has been developed to capture the fragmented knowledge of building assessment in the POE domain, with the benchmarking evaluation axioms encoded in SWRL (Semantic Web Rule Language) rules to enable automatic rule-based reasoning and query. This ontology model also enables effective POE-related knowledge retrieving and sharing, and promotes its implementation in the POE domain. A field study has been conducted based upon the Building Use Studies (BUS) Methodology to validate the proposed ontology framework. This research concludes that the ontology has the feasibility for occupant satisfaction assessment at the building operation stage to provide a multi-criteria optimised solution.

3.1 Research methodology overview

Figure 3-1 gives an overview of the developed methodologies and the corresponding development processes in this research. Overall, the methodology development includes literature review for knowledge acquisition, ontology development based on the ‘7-Steps’ methodology, case study data collection through the BUS Methodology, framework validation and implementation. The more detailed development processes of each method are explained in the following sections.

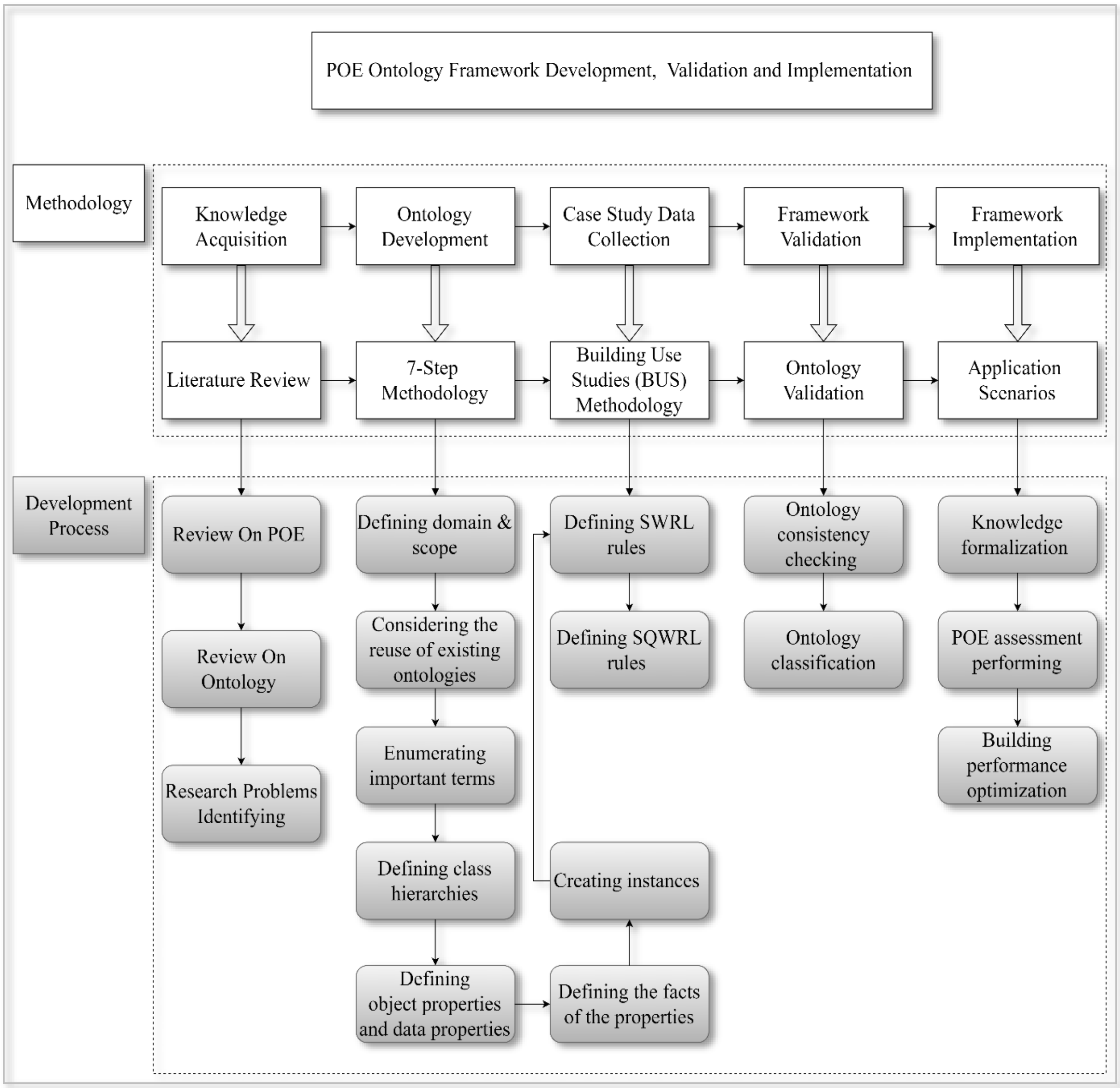


Figure 3-1 Methodology development process

3.1.1 Knowledge acquisition from literature review

Based on a state-of-the-art literature review on post-occupancy evaluation (POE) and ontology, this research identified the research gaps and then determined the appropriate methodologies for the proposed ontology-based POE framework design.

The sources for literature review include but are not limited to journal articles, books, reports, conference proceedings, encyclopedia articles, cases, statutes, theses, ontology libraries, web pages, and so on. In this research, both the primary and secondary sources are adopted

Based on the findings from the literature review, the research aim of building a post-occupancy evaluation ontology framework with the focus on building occupants' satisfaction evaluation of building performances is determined.

3.1.2 Ontology framework development

The ontology framework development mainly includes two parts: the OWL knowledge base development, and the SWRL rules and SQWRL query development.

3.1.2.1 OWL knowledge base development

After determining the research aim, a heavyweight POE OWL ontology has been established based on proposed methods and tools. The '7-Steps' methodology from Noy and McGuinness (2001) is used to develop the POE ontology knowledge base. Specifically, the POE ontology proposed in this research adopts some concepts from the existing OP ontology (Chávez-Feria *et al.*, 2020) and the DNAS framework (Hong, D'Oca, Taylor-Lange, *et al.*, 2015), which realizes the principle of ontology reusing. Based on the existing ontologies, five core components are proposed in the POE ontology, namely Needs, Parameters, Results, Actions and Systems. The expanded corresponding sub-concepts are explained in detail in Chapter 4. In addition, the practical guides from Horridge (2011) and DeBellis (2021) for building OWL ontologies by using ontology editor Protégé have been followed, to build the proposed POE ontology from scratch within the software Protégé 5.5.0 and a set of plug-ins.

3.1.2.2 SWRL rules and SQWRL query development

Based on the asserted axioms in the OWL knowledge base developed above, the SWRL rules for determining the evaluation constraints conditions in POE assessment are defined. After determining the POE constraints SWRL rules, the user-need-driven SQWRL query rules are defined according to the information users wanted to query in the ontology.

3.1.3 Case study

A case study through BUS Methodology is carried out to validate the proposed ontology and to illustrate the feasibility and effectiveness of the ontology framework implementation in different application scenarios.

3.2 OWL ontology knowledge base development methodology

As many ontology experts declare in different studies, there is no single correct way or methodology to design an ontology for any given domain, and ontology design is a creative process. The ontology development is influenced by developers' understanding of the ontology domain and intended usage, as well as their own ontology development experiences (Noy and McGuinness, 2001; Gangolells and Casals, 2012).

The various ontology development methodologies have been reviewed in Chapter 2, for example, Grüninger and Fox's methodology TOVE (Grüninger and Fox, 1995), Uschold and King's method (Uschold and King, 1995), METHONTOLOGY (Fernández-López, Gómez-Pérez and Juristo, 1997; Fernández-López *et al.*, 1999), On-To-Knowledge Methodology OTKM (Staab, Studer and Gmbh, 2001) and Simple-Knowledge Engineering Methodology (SKEM) (Noy and McGuinness, 2001), etc. The main development processes of these methods are summarized in Table 3-1, the Simple-Knowledge Engineering Methodology (SKEM), developed by Noy and McGuinness (2001), also known as the '7-Steps' method, is the most prevailing one.

Table 3-1 Ontology development methodologies

Methodology	Time	Development processes
Uschold and King's methodology	1995	Identify the purpose and scope of ontology → Ontology building (ontology capture, coding, integrating existing ontology) → Evaluation → Documentation
Grüninger and Fox's Methodology (TOVE)	1995	Motivating Scenarios → Informal Competency Questions (CQ) → Formal Terminology → Formal CQ ⇄ Formal Axioms → Completeness Theorems

METHONTOLOGY	1997	Planning → Specification → Knowledge Acquisition → Conceptualization → Formalization → Integration → Implementation → Evaluation → Documentation → Maintaining
On-To-Knowledge Methodology （OTKM）	2001	Feasibility study→ Ontology kickoff→ Refinement→ Evaluation → Maintenance
Simple-Knowledge Engineering Methodology (SKEM) / 7-Steps Method	2001	Determine the domain and scope → Consider reusing existing ontologies →Enumerate import terms → Define the classes → Define the slots → Define the facets of the slots → Create instances

In this research, the SKEM method from the document ‘‘Ontology Development 101: A Guide to Creating Your First Ontology’’ (Noy and McGuinness, 2001) has been referred to as the main ontology design method. The reasons for adopting this method are listed below:

- It is a standardized general ontology-design guide with expert experiences: this ontology design guide is developed on the expert’s own ontology-development experiences, and it applies to most ontology design scenarios.
- Easy to understand for beginners: this ontology design guide provides a starting point for beginners, there is no prior experience required for beginners to crating their first ontology.
- Detailed step-by-step instruction: this methodology provides users with detailed ontology development steps from scratch.
- A practical example for reference: a practical wine ontology design example has been implemented in the ontology-editing environment of Protégé to illustrate the detailed ontology design process.

As shown in Figure 3-2, the conventional ‘7-Steps’ method has been adopted as the basis to develop an enhanced method for heavyweight ontology development in this research. Different from the conventional ‘7-Steps’, the extra steps of ontology SWRL rules editing, SQWRL rules reasoning and query are integrated into the ontology structure procedure to develop an enhanced design method for heavyweight ontology.

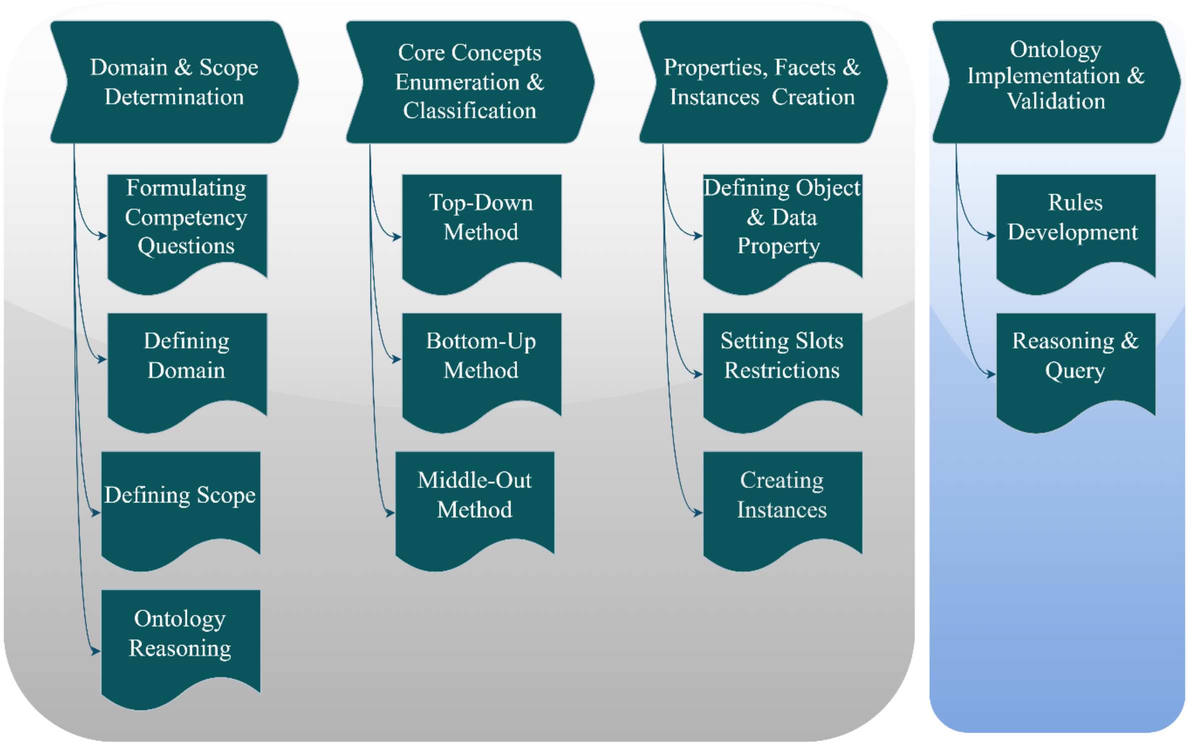


Figure 3-2 Enhanced '7-Steps' ontology development method

After determining the ontology design method, the open-source ontology editor Protégé 5.5.0 has been chosen to construct the ontology model in this research. The reason to use Protégé is that it is a Java-based platform that is extensible and provides a plug-and-play environment, making it a flexible base for rapid prototyping and application development, and it is available to download at: https://protege.stanford.edu/. A set of plug-ins is available in Protégé for users in different development scenarios, and it provides a user-friendly ontology editor and framework with a suite of tools for building intelligent systems (Musen, 2015b).

The protégé tutorial documentation from the Gene Ontology Consortium (2017) offers guidance for using this software with the illustration of its layout of tabs and panels, which can also be configurable by the user. In addition, Horridge (2011) and DeBellis's (2021) OWL ontologies building practical guide provides a detailed illustration of ontology construction processes by demonstrating how to build a pizza ontology. Hence, these technical and practical guides have been followed to create the proposed ontology in this research.

Based on the above review, the general requirements for ontology development have been explained. Developing a fully functional heavyweight ontology involves several tasks, such as knowledge extraction, SWRL rules editing, SQWRL query rules defining, etc. The detailed ontology development procedure includes the following key tasks, and the used development techniques and tools are summarized in Table 3-2:

- Extract knowledge from the POE-related assessment standards to build the OWL knowledge body and SWRL rules.
- Reusing concepts from the existing ontologies. This work aims to develop an occupant-satisfaction oriented post-occupancy evaluation ontology, but the studies on occupants centred ontologies are still limited. In this research, the proposed ontology model is constructed by referring to the DNAS (Drivers-Needs-Actions-Systems) framework and the Occupant Profile (OP) ontology. The structure of the DNAS framework provides valuable ideas for the development of ontology's main structure in this study. In addition, some reusable concepts are adopted from the Occupant Profile (OP) ontology to develop the classes and properties.
- Building survey data and information are integrated into the ontology model to define the axioms and inference SWRL rules.
- The rule engine is used to realize the OWL knowledge and SWRL rules inference and assert new facts into the ontology as axioms after.
- Pellet reasoner is applied to check the consistency of ontology and classify the subclass relations between every named class to create the complete class hierarchy.
- The SWRL rules and SQWRL query rules are edited in the rules editor, SWRL Tab
- The SQWRL Tab is used to execute the assessment query.

The development of ontology involves many aspects, such as the establishment of a knowledge base, the development of SWRL rules, ontology inference and so on, each of which applies different ontology techniques and tools. Overall, the main techniques and their related references involved in this research are summarized in following Table 3-2.

Table 3-2 Key references for ontology development

Techniques	Components	Practical Guide References
Ontology editor	Protégé 5.5.0	https://protege.stanford.edu/ (Protégé, 2016)

Ontology design methodology	7-Steps / SKEM	‘‘Ontology Development 101: A Guide to Creating Your First Ontology’’ (Noy and McGuinness, 2001)
Ontology construction method	Practice guide	‘‘A Practical Guide To Building OWL Ontologies Using Protégè 4 and CO-ODE Tools Edition 1.3’’ (Horridge, 2011)
		‘‘A Practical Guide to Building OWL Ontologies Using Protégé 5.5 and Plugins Edition 3.0’’ (DeBellis, 2021)
Ontology language	OWL	‘‘OWL Web Ontology Language Guide W3C Recommendation 10 February 2004’’ (Smith, Welty and McGuinness, 2004)
Ontology class hierarchy visualization	OWLViz	Protégé 5.5.0 plug-in. ‘‘User-Friendly Ontology Editing and Visualization Tools: The OWLeasyViz Approach’’ (Catenazzi, Sommaruga and Mazza, 2009)
Ontology schema visualization	VOWL	Protégé 5.5.0 plug-in. Visual Notation for OWL Ontologies (VOWL)
Ontology rules language	SWRL	‘‘Writing Rules for the Semantic Web Using SWRL and Jess’’ (O’Connor et al., 2005)
SWRL rules editor	SWRLTab	Protégé 5.5.0 plug-in. ‘‘The SWRLTab: An Extensible Environment for working with SWRL Rules in Protege-OWL’’ (O’Connor and Das, 2005)
Validation engine	Pellet Reasoner	Protégé 5.5.0 plug-in.
Reasoning engine	DroolsTab	Protégé 5.5.0 plug-in.
Query language	SQWRL	‘‘SQWRL: A query language for OWL’’ (O’Connor and Das, 2009)
Query interface	SQWRLTab	Protégé 5.5.0 plug-in

3.3 SWRL rules and SQWRL query development methodology

This development process can be divided into two parts, the SWRL rules development and SQWRL query development.

3.3.1 SWRL rules development

As introduced in Chapter 2, the SWRL is a standard ontological rule language, it provides users with the ability to write Horn-like rules that can be expressed in terms of OWL concepts (classes, properties, individuals), the rules are combined with the OWL knowledge base to reason about OWL individuals of ontology (O'Connor, Shankar, *et al.*, 2007; Hu *et al.*, 2013; Munir and Sheraz Anjum, 2018; Wu *et al.*, 2021). The SWRL rules are developed based on the OWL knowledge base, which provides ontology with the inference function that allows the new inferred facts to be generated after the execution of SWRL rules in the rule engine, and then new facts can be returned to the knowledge base as the new facts of the OWL ontology. The rule engine reasoning logical processes are shown in Figure 3-3.

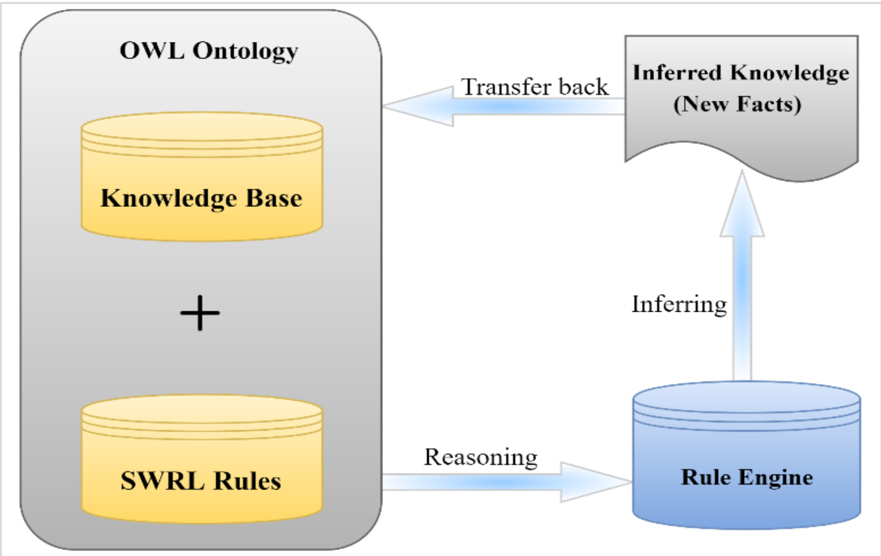


Figure 3-3 SWRL rule engine reasoning logical processes

In this section, the SWRL rules are defined by following the formats of Equation ***(1)***, Equation ***(2)***, and Equation ***(3)***, as explained in Chapter 2. Figure 3-4 presents an example of the expression of OWL atoms in SWRL rules. The OWL entities of class, object property and data property are edited into SWRL rules with their corresponding SWRL atoms. For example, the 'EvaluatedBuilding' class is expressed as EvaluatedBuilding(?eb) in SWRL rules, and the 'isEvaluatedBy' object property which is used to describe the relationship between the 'EvalautedBuilding' class and the 'EvaluationCriteria' class, is expressed as isEvaluatedBy(?eb, ?ec), etc.

In this study, the used built-ins include the math built-ins of swrlb:multiply, swrlb:add, swrlb:subtract, swrlb:abs and swrlb:divide; the comparison built-ins of swrlb:lessThan, swrlb:lessThanOrEqual, swrlb:greaterThan and swrlb:greaterThanOrEqual; and the query operator of sqwrl:select.

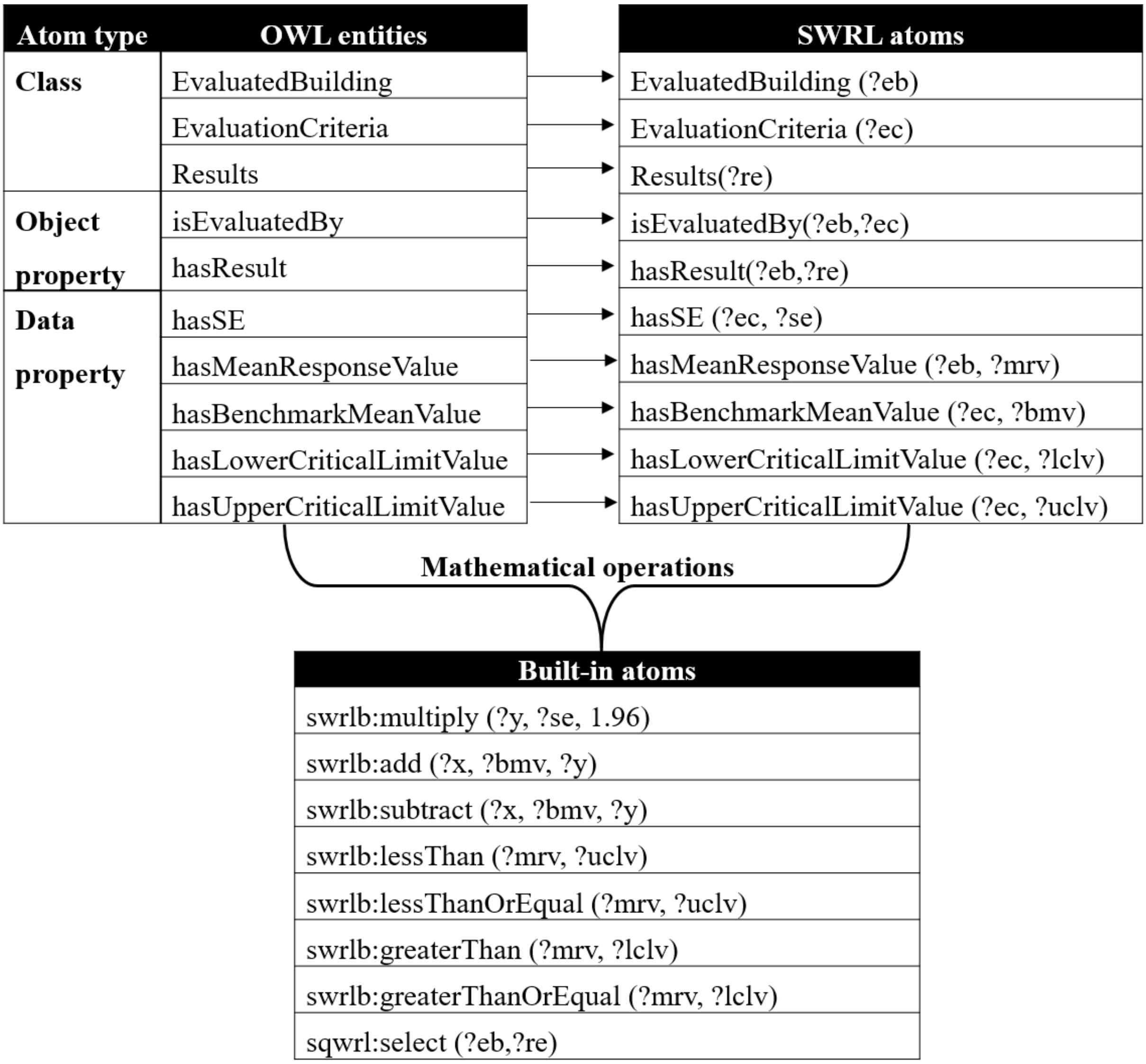


Figure 3-4 Example of the expression of OWL atoms in SWRL rules

The defined SWRL rules are stored in SWRLtab as part of the knowledge base. As the SWRL provides deductive reasoning capabilities by adding rules to extend OWL-DL, after defining the SWRL rules into the ontology knowledge base, the new facts knowledge are generated when executing the reasoner, then the new facts can be inferred from the existing OWL knowledge base, the new knowledge can be written back to the OWL ontology for further SWRL rules reasoning (O'Connor, Shankar, *et al.*, 2007).

Based on the defined SWRL rules, the SQWRL query rules are developed to retrieve information or knowledge needed by users from the OWL ontology. The development of query rules is presented in the following section.

3.3.2 SQWRL query development

SQWRL (Semantic Query-Enhanced Web Rule Language) is an SWRL-based query language that is used for querying information from OWL ontologies (O'Connor, 2016). The SQWRL query rules can be edited in the SWRLTab plug-in, the left side of SQWRL query rules has the same pattern as the SWRL antecedent with its associated semantics. The SQWRL operator in the consequent, *sqwrl:select,* is the core of SQWRL query rules, which is used to select every parameter that matches the query conditions when the query rules are triggered (DeBellis, 2021).

The following equation illustrates the pattern specification of SQWRL query rules:

$$p(arg_1, arg \ldots arg_n) \wedge \ldots p(arg_1, arg_2 \ldots arg_n) \;\; -> sqwrl{:}select(arg_1, arg_2 \ldots arg_n) \qquad (5)$$

where p is a predicate symbol defined in OWL and $arg_1, arg_2 \ldots arg_n$ are the expression of specific terms or parameters.

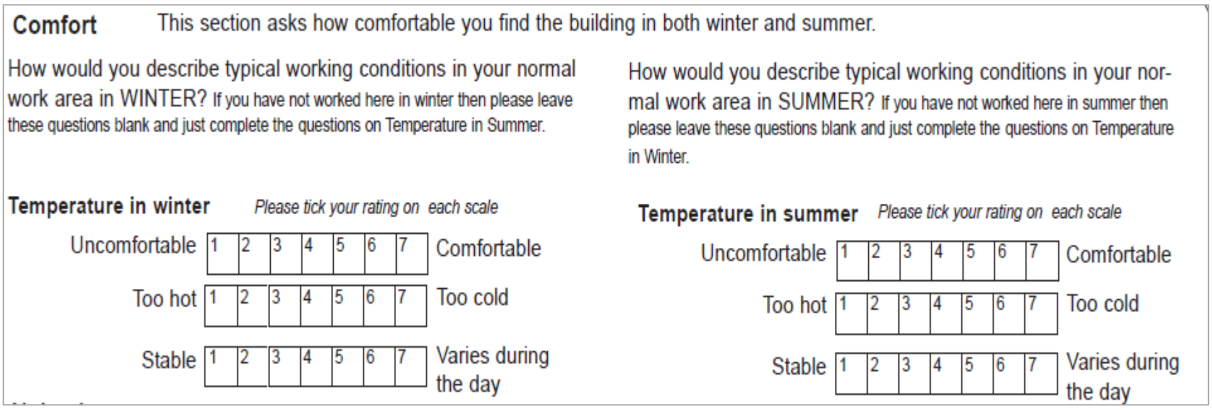
As introduced in Chapter 2, the queries are operated in combination with SWRL rules to retrieve the knowledge which is inferred by SWRL rules. For instance, in this case, a set of SWRL rules are coded in the POE ontology knowledge base to define the lower and upper critical limit values of evaluation criteria. Based on these rules, the SQWRL query can be written to retrieve the corresponding value for evaluation criteria in POE ontology.

3.4 Case study based upon the licensed BUS Methodology

In this research, a field case study is carried out based upon the licensed Building Use Studies (BUS) Methodology to illustrate the validity and feasibility of this ontology framework. The BUS Methodology is a post-occupancy evaluation tool, involving tailored surveys and benchmarking against appropriate standards for the building use type and geographical location. Compared to other questionnaire methodologies, the BUS methodology provides a standardised questionnaires prototype and benchmarks, it also gives dataset comparison. And the building performance-related variables contained in the BUS Methodology meet the scope of this research. By following the license agreement from the BUS Methodology organization, to better develop the benchmarks by incorporating the investigated data into the benchmarks dataset, a license agreement has been signed in advance of using this methodology (Leaman, 2011).

3.4.1 Building Use Studies (BUS) Methodology

The Building Use Studies (BUS) Methodology is a method of evaluating the building performance in meeting the needs and the satisfaction of building occupants, which was developed and refined in the 1990s (UK Green Building Council, 2013). It can be used to identify both good and poor building performance features. The BUS Methodology adopts a paper-based and web-based questionnaire survey method to assess building performance in various quantitative and qualitative questions in the aspects of background information (e.g. gender, room layout, or time stay in buildings, etc.), thermal comfort, ventilation, IEQ, lighting, noise, personal control, perceived productivity, health, and so on. As shown in Figure 3-5, building occupants can rate various performance-related building parameters based on a 7-point semantic differential scale using two adjectives with a neutral point (e.g. '1=too cold and 7= too hot'), they can also provide written feedback (Khoshbakht, Gou, Xie, et al., 2018). It is an efficient tool that has been widely used in many POE studies (R. Cohen et al., 2001; Jin Woo, 2017; Parkinson et al., 2018; Zhao and Yang, 2021, 2022). It is also a recognised tool for post-occupancy evaluation contributing toward achieving building performance labelling and certification such as BREEAM, LEED, WELL Standard, and Soft Landings (BUS METHODOLOGY, 2017).



Comfort This section asks how comfortable you find the building in both winter and summer.

How would you describe typical working conditions in your normal work area in WINTER? If you have not worked here in winter then please leave these questions blank and just complete the questions on Temperature in Summer.

Temperature in winter *Please tick your rating on each scale*

Uncomfortable	1	2	3	4	5	6	7	Comfortable
Too hot	1	2	3	4	5	6	7	Too cold
Stable	1	2	3	4	5	6	7	Varies during the day

How would you describe typical working conditions in your normal work area in SUMMER? If you have not worked here in summer then please leave these questions blank and just complete the questions on Temperature in Winter.

Temperature in summer *Please tick your rating on each scale*

Uncomfortable	1	2	3	4	5	6	7	Comfortable
Too hot	1	2	3	4	5	6	7	Too cold
Stable	1	2	3	4	5	6	7	Varies during the day

Figure 3-5 BUS Methodology questionnaire question example

The licensed BUS Methodology questionnaire has been applied to carry out a survey assessment in two non-domestic buildings, the Eastern Gateway (ESGW) building and Michael Sterling (MCST) building, they are selected from a university in the United Kingdom, these two buildings are functionally similar (Kawneer UK Limited, 2014). The collected assessment data is analysed by comparison with similar buildings' benchmark values from the database of BUS Methodology, more details about the data analysis are given in Chapter 5.

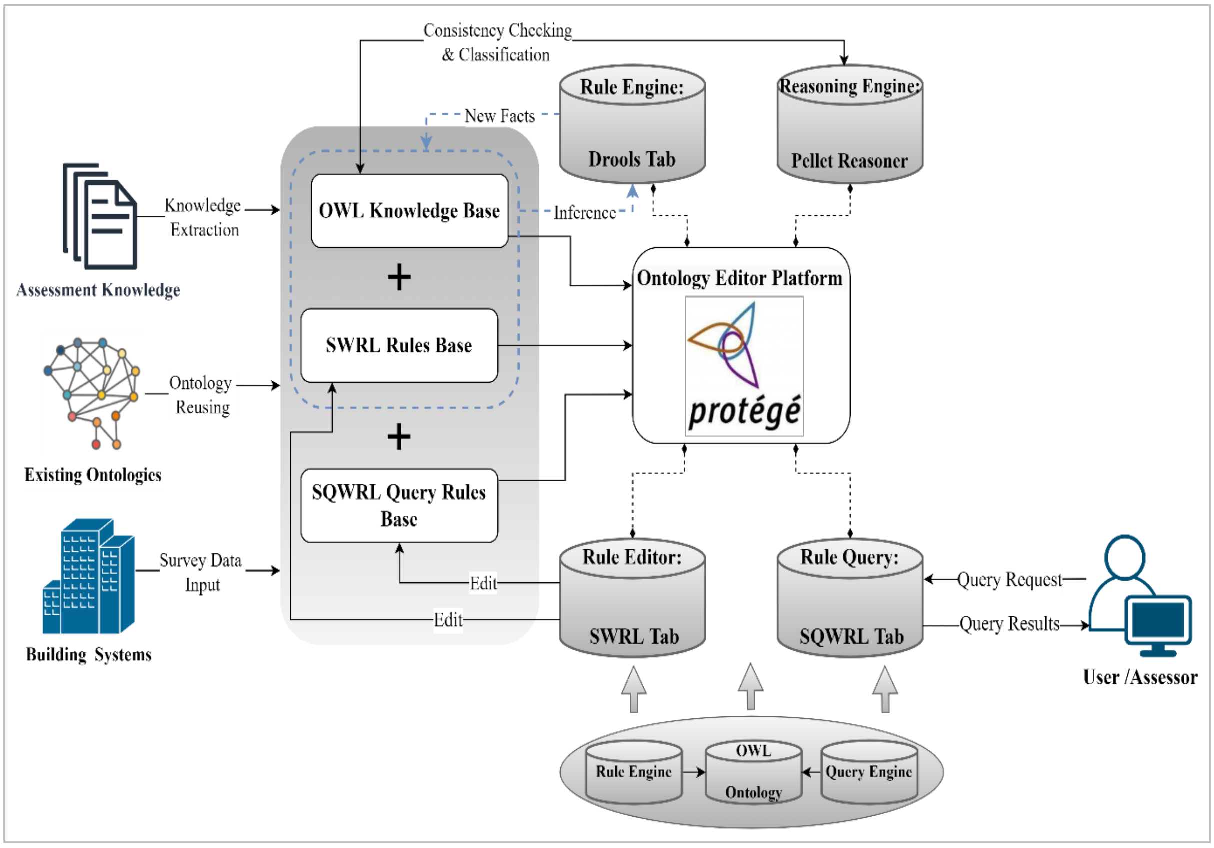


Figure 3-6 The development of the POE ontology framework

3.4.2 Framework validation and implementation

To validate the proposed ontology, the Pellet reasoner is used to carry out a consistency checking and classification of ontology, and to infer implicit new axioms of ontology from the given explicit facts in the knowledge base. Also, the Drools rule engine can be used here to infer new knowledge, transfer the inferred rule engine knowledge to the OWL knowledge, and write back to the ontology model.

Figure 3-6 presents the development of the proposed POE ontology framework in detail, including the POE knowledge acquisition and formalization, SWRL rules and SQWRL query definition, as well as the used technologies and tools, etc.

3.5 Summary

This chapter illustrates the developed methodologies for this research. Overall, the methodology development includes a start-of-the-art literature review, a fully functional heavyweight POE ontology development based on the enhanced '7-Steps' methodology, case study data collection based upon the BUS Methodology, framework validation and implementation.

Based on the findings from the literature review, the research gaps are identified, and the developed methodologies for developing an ontology-based POE framework are analysed. The ontology framework development mainly includes three parts: the OWL knowledge base development, SWRL rules development and SQWRL query development.

Based on the combination of pre-defined POE constraints conditions SWRL rules and the user-need-driven SQWRL queries, a case study of POE assessment is carried out in the proposed framework, to illustrate the feasibility and effectiveness of the ontology framework implementation in different application scenarios.

The next chapter presents the detailed development process of the proposed framework.

Chapter 4 Framework design and development

This chapter presents the development of the proposed framework of this research. The overview of the proposed ontology-based post-occupancy evaluation framework is illustrated in Section 4.1; the OWL ontology knowledge base structure processes are explained step by step in Section 4.2; in Section 4.3, the POE related rules are edited into SWRL rules to support the ontology reasoning in this research; the SQWRL queries are edited in Section 4.4; the ontology reasoning and querying examples are given in Section 4.5; the chapter summary is given in Section 4.6.

4.1 Overview of the proposed ontology-based POE framework

As shown in Figure 4-1, the development of the proposed ontology-based post-occupancy evaluation framework consists of three main steps: Step 1 ontology development, Step 2 ontology application validation, and Step 3 optimization of actions that buildings need to take in the future to meet building occupant needs.

- Step 1: Ontology development: this is the core step of the framework development, including knowledge extraction, survey building data collection and input, and assessment SWRL rules editing based on the existing assessment knowledge. This research builds the ontology model in the environment of Protégé 5.5.0 software. The assessment knowledge is extracted from the POE related standards and some existing ontologies to enumerate core concepts and define the properties, values, and axioms of these concepts. The collected survey building data and information from the survey are integrated into the ontology model as the values of instances and also is edited into SWRL rules to set the assessment constraints for ontology applications and validations.
- Step 2: Ontology application validation: rules-based reasoning and querying. Based on the edited ontology knowledge and SWRL rules from Step 1, the new OWL knowledge is generated in the rule engine after the logical reasoning, and the evaluation can be conducted by using the querying rules with the SQWRL query engine.
- Step 3: Building operation feedback and optimization: according to the evaluation results from Step 2, the report of assessment negative results and corresponding

optimization suggestions is generated, and the optimization suggestions are feedbacked to the facility manager to adjust the operation mode of the building operating systems.

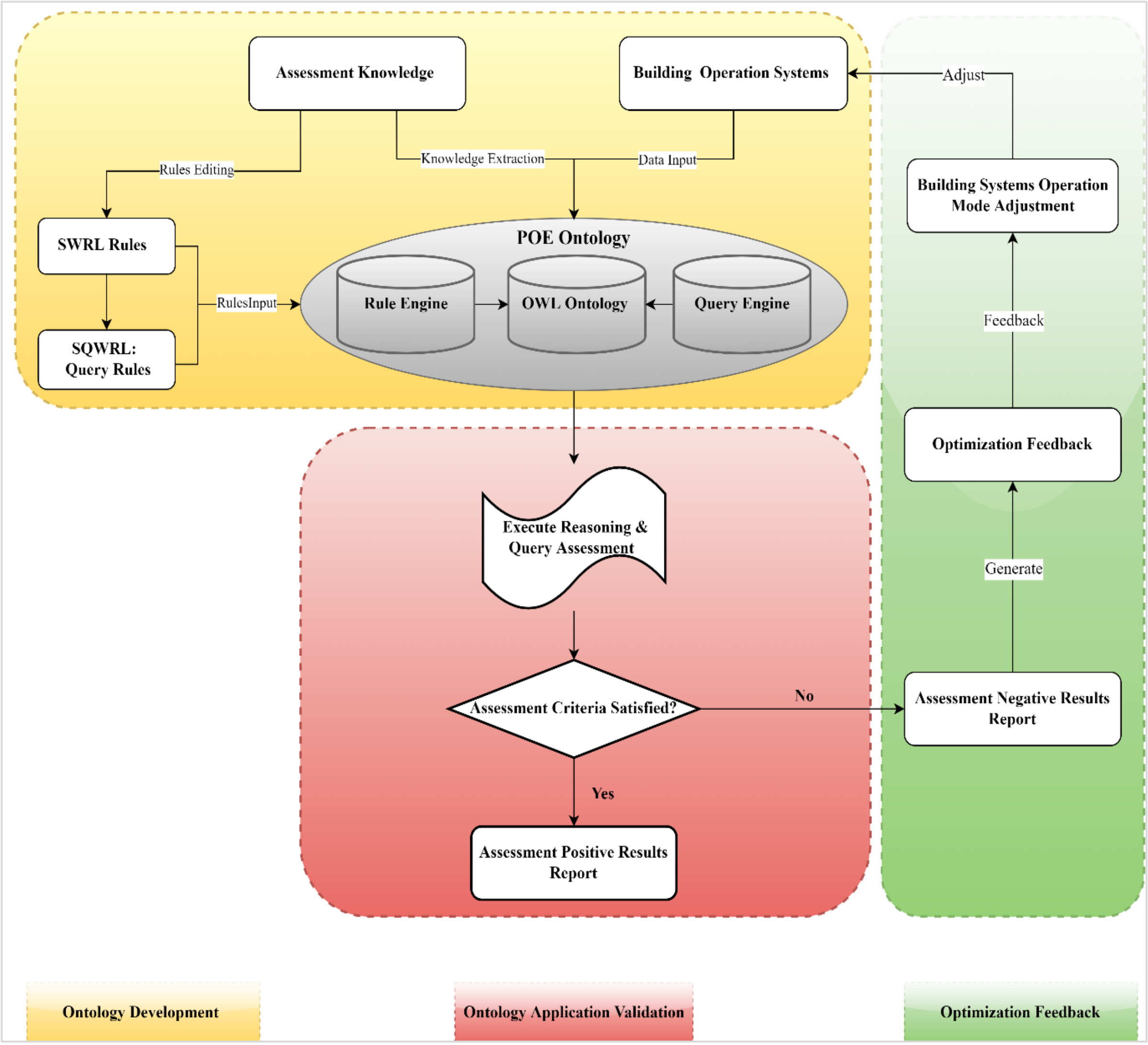


Figure 4-1 Proposed ontology-based post-occupancy evaluation framework

As shown in Figure 4-2, by following the development method of the DNAS framework (Hong, D'Oca, Taylor-Lange, *et al.*, 2015), in this research, there are five core components developed in the proposed ontology, namely Needs, Parameters, Results, Actions and Systems. The full list of the top-level classes of the proposed POE ontology is given in Table 4-2.

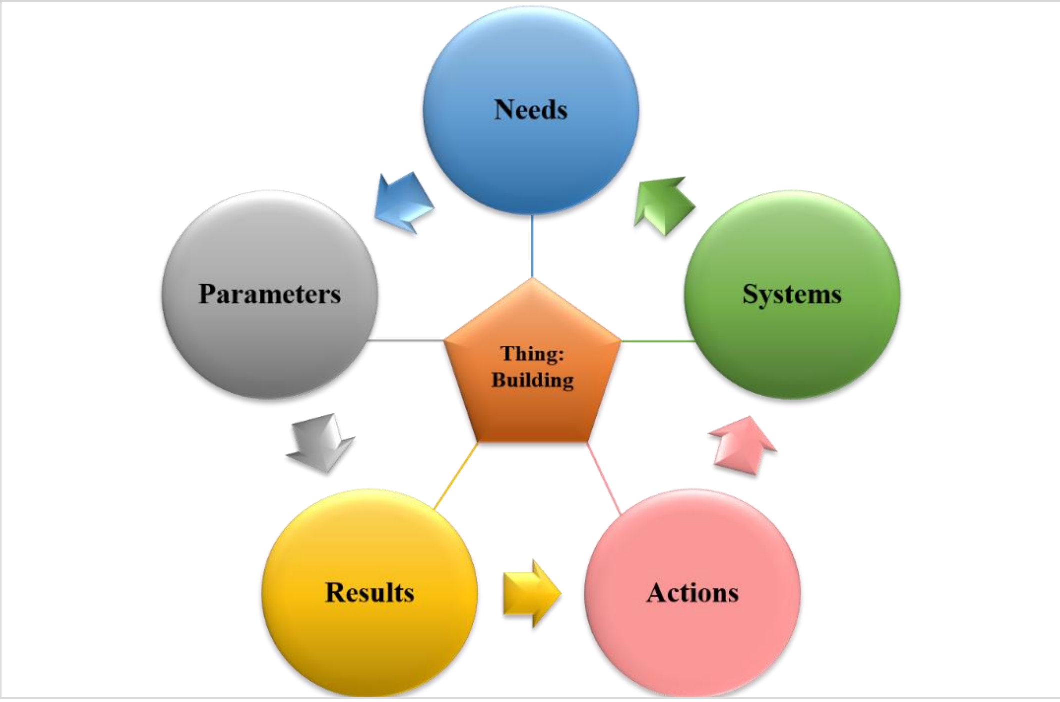


Figure 4-2 The core ontology development components

The expanded corresponding sub-concepts are partly shown in Figure 4-3. Needs are occupants' physical and non-physical requirements from buildings, such as the physical needs of indoor air quality, thermal comfort, and acoustic need; and the non-physical requirements of cleaning, safety, productivity, space, etc; the sub-concepts of the Needs class are presented in Table 4-4 and Figure 4-7. Parameters are these building performance evaluation criteria in the aspects of comfort, indoor environmental quality, temperature, building design, system smart levels, etc; the contained sub-classes of the parameters are listed in Table 4-3 and the hierarchy structure of this class is shown in Figure 4-8. Result refers to the results of evaluation generated from the assessment process, e.g. too hot, satisfaction, unsatisfactory, etc; the full list of instances of the Result class is shown in Figure 4-10. Action means the interactions with building operation systems that the facility manager adopted from the result feedback report to adjust the systems operation mode to achieve a better environmental satisfaction and comfort; the subclasses of the Actions class are listed in Table 4-5 and the developed instances of each subclass are presented in Figure 4-9. System refers to the building systems, such as heating systems, window systems, etc; the listed building systems instances and the related relations are illustrated in Figure 4-13.

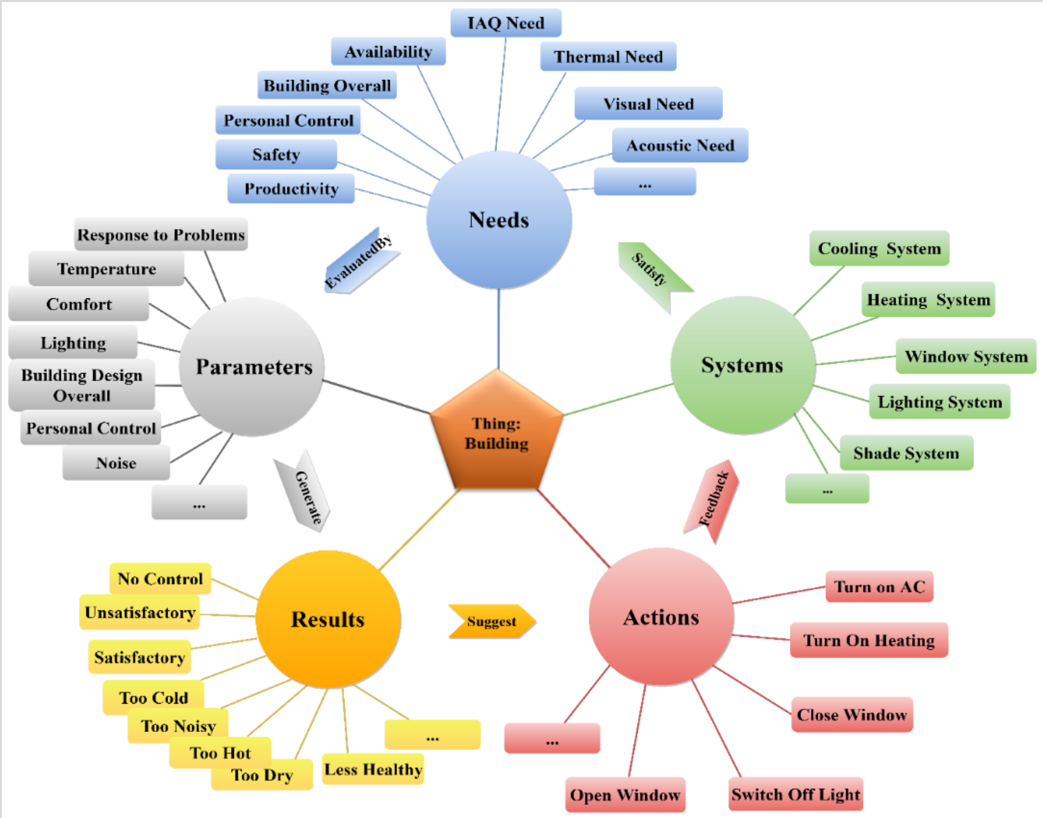


Figure 4-3 Ontology top-level concepts

The logic relations can be read as building occupants have needs of the environment they stay in, occupants' needs are evaluated by building parameters, the evaluation generates the result, the result suggests the corresponding adjustment action, and the suggested action will feedback to the system. Then the systems adjust the operation mode to satisfy the occupants' needs. Based on this The following few sections present the details of each step's development process.

4.2 OWL ontology development

Many ontology experts declare that there is no single correct way or methodology to design ontologies for any given domain, and ontology design is a creative process. Ontology development is influenced by developers' understanding of the ontology domain and intended usage, as well as their own ontology-development experiences (Noy and McGuinness, 2001; Gangolells and Casals, 2012). By following the enhanced '7-Steps' methodology illustrated in Figure 3-2 and the above ontology model development structure, the following subsections present the detailed development procedures of the proposed post-occupancy evaluation (POE) ontology.

4.2.1 Domain and scope of the proposed ontology

The development of ontology starts with defining its domain and scope. According to the ontology development guide from Noy and McGuinness (2001), listing and answering some basic questions before starting to build an ontology that can help determine the domain of ontology. In this research, some basic questions are listed below, such as what domain will be covered by the ontology, why to develop the ontology, what kind of information will be provided by the ontology, etc.

- What is the purpose to develop this ontology?

    To make up for the lack of ontology research in the POE domain, especially with the focus on occupant satisfaction. Developing a heavyweight ontology to capture the fragmented knowledge of building assessment in the POE domain with the ultimate aim of optimizing building operation guidelines and improving occupant use experience quality and well-being.

- What are the domains of this ontology covers?

    - This ontology is developed for the post-occupancy evaluation of buildings, especially covers the occupant's satisfaction assessment in the aspects of building design overall, indoor environmental quality (IEQ), thermal comfort, personal control, convenience, health and productivity, smart readiness levels, etc.

- What are the sources of knowledge and data needed to develop this ontology?

    - The related knowledge from the building assessment standards, practical guide, the existing ontologies concepts, the survey data from buildings.

- Are there any other reusable ontologies?

    - There are no existing ontologies that can be reused directly, but some ontology structure development ideas are borrowed from the DNAS framework, and some concepts are adopted from the OP ontologies.

- Who will use and maintain the ontology?

    - This proposed ontology can be used by building performance assessment specialists, companies, or facility managers.

The questions listed above are some general questions to help define the ontology domain, there are more specific questions for each development step to check whether the ontology can represent the desired domain knowledge.

In addition, according to Grüninger and Fox's Methodology (TOVE) (1995), a set of competency questions can help determine the scope of ontology. The answer to the competency questions varies from domain to domain, and they may change during the ontology development. In this research, the proposed ontology should be able to answer the following possible competency questions(CQs):

- What types of buildings should be evaluated?
- Which assessment methods should be applied?
- What are the constraints for the evaluation items?
- What kind of tasks are included in the post-occupancy evaluation event?
- Which aspects of building performance should be included in evaluation activities?
- What are the corresponding actions that should be taken if the assessment result failed?
- What are the smartness levels of the building systems?
- What are the occupant satisfaction levels for assessed building performance?
- ……

Here, we are not trying to enumerate all the relevant questions, but judging from the above competency questions, this ontology will represent the corresponding information including building types, building evaluators, evaluation approaches, evaluation criteria, assessment tasks, green label certification levels, smartness levels of building technologies and so on.

4.2.2 Considering the reuse of existing ontologies

The advantages of data interoperability and flexible data exchange capabilities have promoted ontologies integration and reuse when developing new ontologies. After determining the domain and scope of the proposed ontology, it is almost always worth considering whether there are relevant ontologies that already existed that can be directly reused or extended (Noy and McGuinness, 2001). For example, in the field of the AEC industry, the ifcOWL is an OWL representation of the IFC data model which describes the building elements, material properties, costs, organizations and so on (Pauwels and Terkaj, 2016), and the Brick ontology can be used as a uniform metadata schema for BMS of smart buildings (Balaji et al., 2016, 2018). Ontology reuse can not only save the time and effort of ontology engineers while building ontologies, but also avoid the redundancy of domain knowledge (Simperl, 2009).

Based on the broad literature review in Chapter 2, many existing ontologies have been developed to support knowledge representation and management within the building evaluation domain, but there are not many ontologies that are occupant-centred developed. Several schemas represent occupants' activities and their behaviours in buildings, for example, the obXML schema/DNAS Framework (Hong, D'Oca, Turner, *et al.*, 2015), Onto-SB: Human Profile Ontology for Energy Efficiency in Smart Building (Degha *et al.*, 2018), OnCom (Orozco *et al.*, 2019), 'Occupant Behavior in Dynamic Environments' (OBiDE) framework (Arslan, Cruz and Ginhac, 2019), Occupancy Profile (OP) ontology (Chávez-Feria *et al.*, 2020). As the first XML schema that presents energy-related occupant behaviours analysis, obXML (Hong, D'Oca, Turner, *et al.*, 2015) laid the foundation for the development of other occupant-behaviour-related ontologies, the development of OP ontology and OBiDE framework are both developed based on the obXML schema (Pritoni *et al.*, 2021). The OP ontology represents the occupant behaviours and activities inside buildings with a focus on the energy impact of their actions. Instead of developing new classes, the authors reused the concepts available in the SAREF series of ontologies, to allocate with the systems adopted from the obXML schema (Chávez-Feria, Poveda-Villalón and García-Castro, 2020).

Since the obXML schema and OP ontology are the two most relevant ontologies to this research, so, in this study, the proposed ontology model is constructed by referring to the DNAS framework and the OP ontology. The structure of the DNAS framework provides valuable ideas for the main structure development of POE ontology, and some reusable concepts are adopted from the OP ontology to develop the classes and properties in this research.

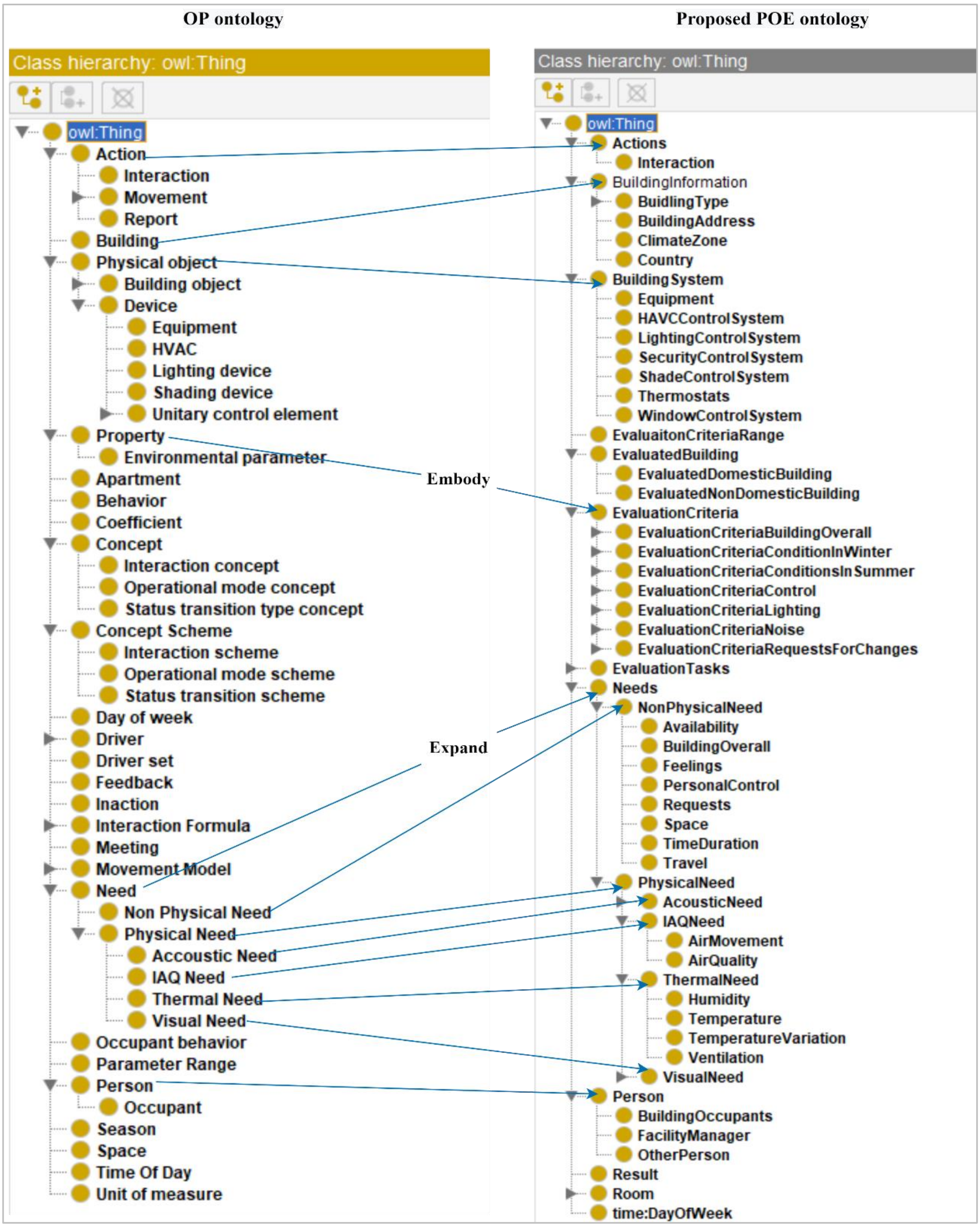


Figure 4-4 Top-level classes mapping from OP ontology

The OP ontology offers a general framework to represent the energy-impact related occupant's behaviours inside buildings, and to understand how the occupant's behaviours would affect building energy performance, but it does not provide development details in terms of occupants' needs identification and the building elements parameters specification, and so on. The mapping relationships of the main top-level classes between the OP ontology and the proposed POE ontology are shown in Figure 4-4.

The class of '*Need'* in OP ontology has two sub-classes of *'Non Physical Need'* and *'Physical Need'*, and there are another 4 subclasses of '*Acoustic need', 'IAQ Need', 'Thermal Need'* and *'Visual Need'* under the class of *'Physical Need'*. The proposed POE ontology develops the *'Needs'* class by reusing the concept of need of the OP ontology but expanding to more specific occupant's needs. For example, the *'NonPhysicalNeed'* class has the subclasses of *'Availability', 'Space', 'PersonalControl'*, etc.; the *'Physical Need'* class has the same 4 subclasses of '*Acoustic need', 'IAQ Need', 'Thermal Need'* and *'Visual Need'*, but each of these 4 subclasses have their expanded subclasses, for example, the *'IAQNeed'* class has its subclasses of *'AirMovement'* and *'AirQuality'; 'ThermalNeed'* class has its subclasses of *'Humidity', 'Temperature', 'TemperatureVariation', 'Ventilation',* and so on.

The concepts from the need classes are be measured by different building environment elements, for example, four factors are recognized as being related to thermal comfort, namely temperature, temperature variation, humidity and ventilation. The visual need can be influenced by the glare and light. Moreover, these factors can all be subdivided into more specific items, like the glare can be subdivided into glare from lights and the glare from the sun and sky, the availability needs include the availability of meeting rooms, the availability of furniture, and so on. In addition, a set of evaluation criteria have been developed in this model to measure the satisfactory levels of the occupant needs. The following sections will give more details on these class concepts.

4.2.3 Enumerating important terms in the ontology

At the beginning of building an ontology, it is always useful for ontology engineers to list all the relevant terms in the given domains. When listing the important concepts, there is no need to worry about the overlap between these listed concepts. In this step, the priority is to get a full list of concepts first, these concepts include but not are limited to class individuals, the properties or the attributes of concepts, and so on. For example, in this proposed building post-occupancy evaluation ontology, the important related concepts may include the evaluated building type and layout, the evaluation tasks, methods and the evaluators; the evaluation criteria, such as the indoor environment quality, security and safety, open space designs, thermal and visual comfort, the building operation management system, and so on. This step is the essential preparation work for the next two steps.

According to the domain of ontology being built, there are some domain related questions that can be used to help determine the important concepts. In this research, the questions are listed below:

- What are the POE-related terms this ontology will present?

The building post-occupancy evaluation related concepts include evaluation criteria of temperature, thermal comfort, humidity, safety, productivity, health, building design, etc; the building systems of HVAC systems, lighting systems, etc; building occupants, occupants needs, assessors, evaluation results, different building types, and so on.

- What are the relationships or properties between these terms?

The internal relations of the concepts can be defined as object properties, there are many different relationships between these concepts, for example, the relationship between the building occupants and needs is *hasNeeds*, more specifically, *hasNeedAirQuality*, *hasNeedTemperature*, etc. Except for the object property, there is data property, which defines the relations between concepts and the data-type values. For example, the evaluation criteria have a lower critical limit value, upper critical limit value, etc.

- What attributes do these terms have?

In this research, the attributes of concepts are described in data properties of them with different datatypes. A simple example of some listed concepts and their related properties and attitudes is shown in Table 4-1. The detailed lists of concepts and their properties are given in the following sections.

Table 4-1 Enumerating important terms example

Class	Items	Properties	Attributes
EvaluationCriteria			
	CriteriaTemperatureInWinter	hasBenchmarkMeanValue	4.48
		hasSE	0.06

		hasUpperLimitValue	Mean +1.96SE
		hasLowerLimitValue	Mean - 1.96SE
EvaluatedBuilding	EvaluatedBuilding	hasResultTSHOT	-
		hasBuildingSystem	-
		hasAction	-
		hasAssessor	
		……	……
Needs	NonPhysicalNeed		
	AcousticNeed		
	IAQNeed		
	ThermalNeed		
	……	……	……

Uschold and Gruninger (1995; 1996) introduced the terms enumeration in ontology capture step of their ontology development methodology, which includes three tasks: ‘‘ (1) identification of the key concepts and relationships in the domain of interest, (2) production of explicit unambiguous text definitions for such concepts and relationships, (3) identification of terms to refer to such concepts and relationships’’ (Fernández-López, 1999; Darlington and Culley, 2008; Gangolells and Casals, 2012). In the guideline of ‘Ontology Development 101’, the first two tasks are simply expressed as developing the class hierarchy and defining properties of concepts (slots), they are the most important steps in building an ontology (Noy and McGuinness, 2001).

4.2.4 Define the classes and the class hierarchy

After listing all the relevant domain concepts in the previous step, this step adopts the approaches from Uschold and King (1995) to classify the concepts into different class hierarchies: top-down, bottom-up, middle-out (Uschold and Gruninger, 1996; Noy and McGuinness, 2001; Abanda and Tah, 2008; Darlington and Culley, 2008).

- Top-down approach: This approach starts with defining the most general concepts in the given domain firstly and then descending into more specialized concepts (Uschold and Gruninger, 1996; Corcho, Fernández-López and Gómez-Pérez, 2003). For example, this research can start with creating the general class of occupants needs, then specialize this class by creating some subclasses of it: temperature need, humidity need, air quality need, visual need, and so on. This method results in better control of the level of detail, but also arises a risk of less stability and increases the re-work and greater effort (Fernández-López and Gómez-Pérez, 2002).

- Bottom-up approach: By contrast with the top-down method, the most specific concepts are firstly defined in this approach, and then generalised to groups. For example, this work can start by defining the classes of end user or the third party, and then create a common superclass for these listed classes i.e. assessor. However, this method results in a very high level of detail, and increases the risk of inconsistencies, rework and greater effort (Fernández-López and Gómez-Pérez, 2002).
- Middle-out approach: It is a mixed approach, which combines the top-down and bottom-up methods. The more salient concepts will be identified first, and then generalized and specialized the initially defined concepts appropriately. This method makes a balance control of the level of detail, and it is easier to spot commonalities and reduce the inaccuracies, re-work and overall (Noy and McGuinness, 2001).

However, none of these three approaches is inherently better than the others, the approaches used to build the ontology are strongly dependent on ontology developers' experience and their understanding of the given domain. The mixed approach is often the easiest and most widely used among most developers (Noy and McGuinness, 2001).

In this study, a middle-out approach is used with defining the more salient concepts first, then generalise and specialise these concepts into different taxonomic class hierarchies by following the rule of ''if class A is a superclass of class B, then every signal instance of class B is also an instance of class A''. The relationship between class B and class A is a 'kind of ' or an "is-a" relation.

In the developed POE ontology, the general classes include *EvaluatedBuilding, EvaluationCriteria, Needs, Person, BuildingSystem, BuildignInformation, results, Actions, Room, Season, TimeOfDay, Time:DayOfWork, ParameterRange,* and *EvaluationTasks.* Some of the top-level classes have sub-classes, for example, according to the different types of buildings evaluated, the class of *EvaluatedBuidling* is divided into *EvaluatedDomesticBuidling* and *EvaluatedNonDomesticBuilding*; the *Needs* class includes the subclass of *NonPhysicalNeed* and *PhysicalNeed*, etc.

In this study, the top-level classes are shown in Table 4-2, most of them can be further divided into more specific subclasses, the top-level class hierarchy of the proposed ontology is shown in Figure 4-5. Next, the more detailed subclasses hierarchies of the superclass are introduced separately.

Table 4-2 Top-Level classes of the POE ontology

SuperClass	**Subclass**
EvaluatedBuilding	EvaluatedDomesticBuilding
	EvaluatedNonDomesticBuilding
EvaluationCriteria	EvaluationCriteriaBuildingOverall
	EvaluationCriteriaConditionInWinter
	EvaluationCriteriaConditionsInSummer
	EvaluationCriteriaControl
	EvaluationCriteriaLighting
	EvaluationCriteriaNoise
	EvaluationCriteriaRequestsForChanges
Needs	NonPhysicalNeed
	PhysicalNeed
Person	BuidlignOccupants
	FacilityManager
	OtherPeople
BuildingSystem	Equipment
	HVACControlSystem
	LightingControlSystem
	SecurityControlSystem
	ShadeControlSystem
	Thermostats
	WindowControlSystem
Results	

Actions	Interaction
	Feedback
	BehaviourChange
BuildingInformation	BuidlignType
	BuildignAddress
	ClimateZone
	Country
EvaluationTasks	Assessor
	EvaluationMethod
	EvaluationTime
PerformanceResultGap	
Room	
FunctionalityLevel	
DayOfWeek	
ParameterRange	
Season	

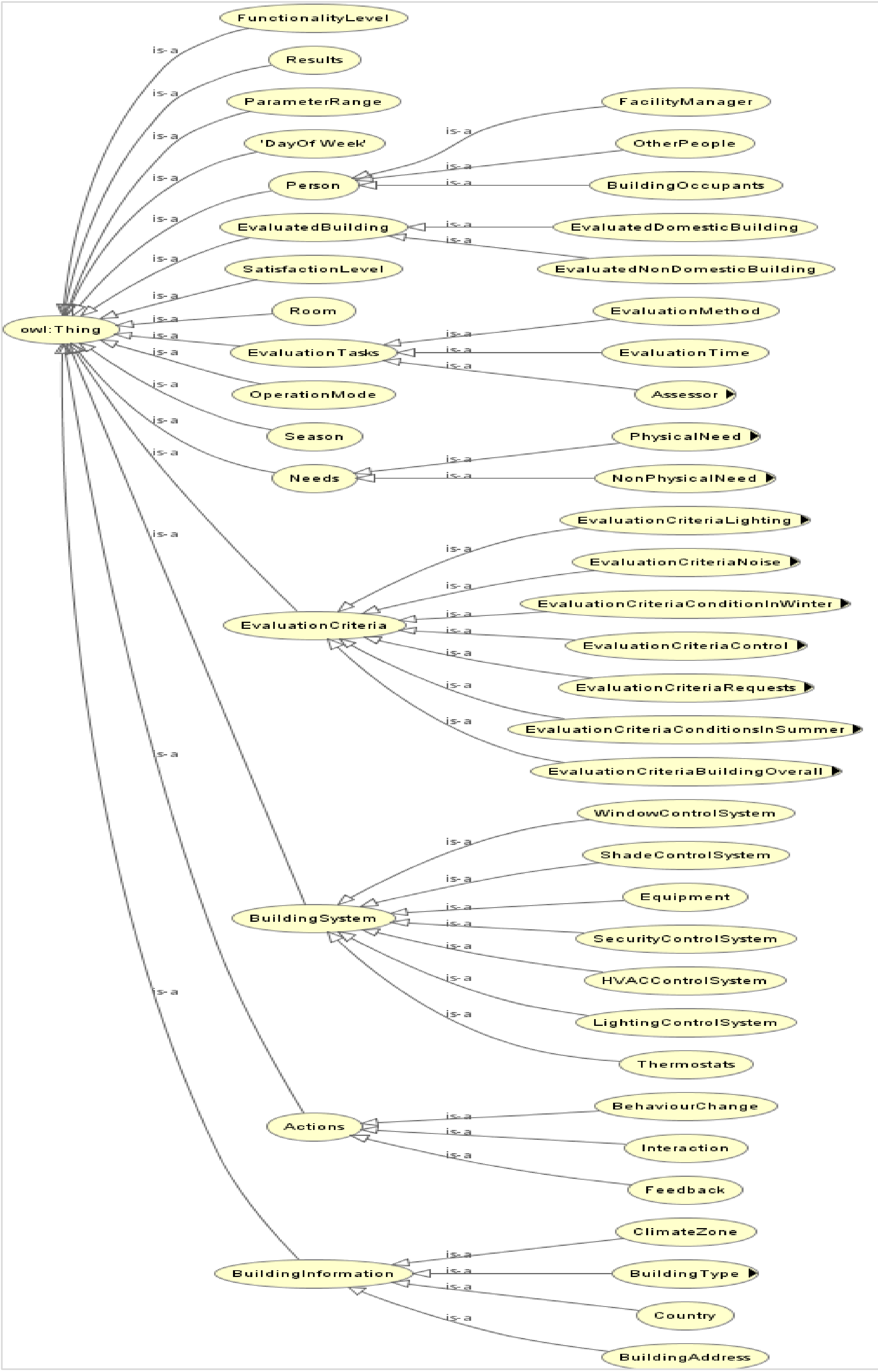


Figure 4-5 Top-level class hierarchy of the proposed POE ontology

The *EvaluationCriteria* class has seven subclasses, namely: *EvaluationCriteriaBuildingOverall*, *EvaluationCriteriaControl*, *EvaluationCriteriaLighting*, *EvaluationCriteriaConditionInWinter*, *EvaluationCriteriaNoise*, *EvaluationCriteriaRequests*, and *EvaluationCriteriaConditionInSummer*. Most concepts of this *'EvaluationCriteria'* class are developed based on the Building Use Studies (BUS) methodology. Each of the subclasses has its subclasses, take the *EvaluationCriteriaConditionsInSummer* as an example, which means the building conditions in the summer season, including the air quality, temperature, humidity, ventilation, and so on; their corresponding concepts in the ontology are *CriteriaAirQualityInSummer*, *CriteriaTemperatureInsummer*, *CriteriaHumidityInSummer*, *CriteriaVentilationInSummer*, respectively. The full list of evaluation criteria concepts is shown in Table 4-3, the class hierarchy is shown in Figure 4-6, and it shows the *'is-a'* relationship, which also can be called a *'has subclass'* relationship

Table 4-3 EvaluationCriteria class hierarchy

Class	**Subclass**	**Subclass**
Evaluation Criteria	EvaluationCriteria BuildingOverall	CriteriaAdequacyOfSpaceAtWorkArea
		CriteriaAvailabilityOfMeetingRooms
		CriteriaBuildingDesignOverall
		CriteriaCleaning
		CriteriaFacilitiesMeetNeeds
		CriteriaGreenBuildingCertification
		CriteriaImageToVisitors
		CriteriaOverallComfort
		CriteriaPerceivedHealth
		CriteriaPersonalSafety
		CriteriaProductivityAtWork
		CriteriaSpaceUse
		CriteriaSRI
		CriteriaStorageArragements
		CriteriaUsabilityOfFurniture
	EvaluationCriteria	CriteriaAirMovementInWinter
		CriteriaAirQualityInWinter

	ConditionInWinter	CriteriaConditionsInWinterOverall
		CriteriaHumidityInWinter
		CriteriaTemperatureInWinter
		CriteriaTemperatureVariationInWinter
		CriteriaThermalComfortInWinter
		CriteriaVentilationInWinter
	EvaluationCriteria ConditionInSummer	CriteriaAirMovementInSummer
		CriteriaAirQualityInSummer
		CriteriaConditionsInSummerOverall
		CriteriaHumidityInSummer
		CriteriaTemperatureInSummer
		CriteriaTemperatureVariationInSummer
		CriteriaThermalComfortInSummer
		CriteriaVentilationInSummer
	EvaluationCriteriaControl	CriteriaPersonalControlOverCooling
		CriteriaPersonalControlOverHeating
		CriteriaPersonalControlOverLighting
		CriteriaPersonalControlOverNoise
		CriteriaPersonalControlOverVentilation
	EvaluationCriteriaLighting	CriteriaAmountOfArticialLight
		CriteriaAmoutnOfNaturalLight
		CriteriaGlareFromLights
		CriteriaGlareFromSunAndSky
		CriteriaLightingOverall
	EvaluationCriteriaNoise	CriteriaFrequencyOfUnwantedInterruptions
		CriteriaNoiseFromColleagues
		CriteriaNoiseFromOutside
		CriteriaNoiseOverall
		CriteriaOtherNoiseFromInside
	EvaluationCriteria Requests	CriteriaEffectivenessOfResponseToRequests
		CriteriaSpeedOfResponseToRequests

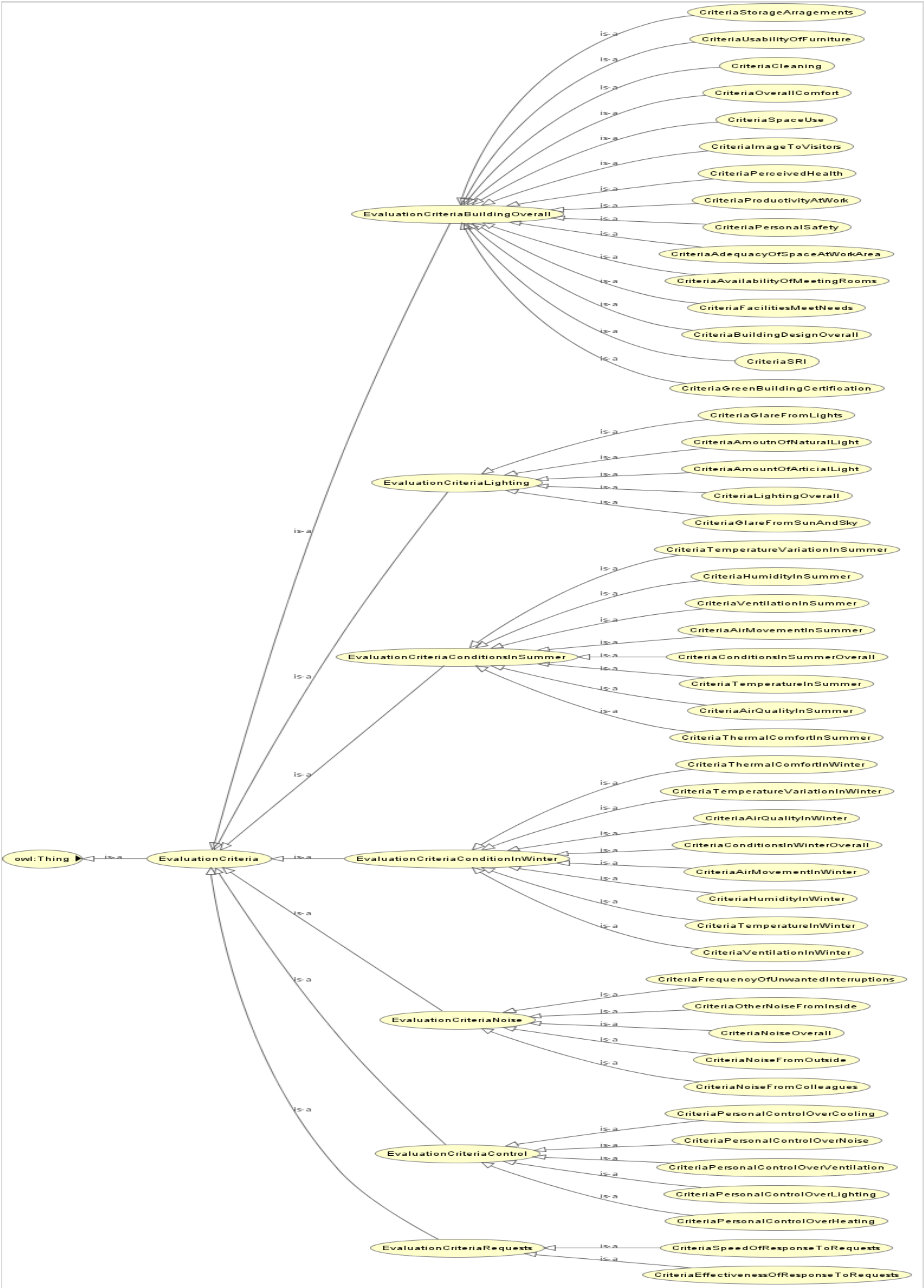


Figure 4-6 EvaluationCriteria class hierarchy

The *'Needs'* class is the class to represent the occupant's requirements in buildings, which is categorized into the subclasses of '*NonPhysicalNeed*' and '*PhysicalNeed*'. This *Needs* concept is adopted from the existing obXML schema and the OP ontology, but extended and enriched with more specific sub-concepts. As shown in Table 4-4, the '*NonPhysicalNeed*' is comprised of the 7 sub-classes of the *'Availablity', 'BuildingOverall', 'Feelings', 'PersonalControl', 'Requests'*, *'TimeDuration',* and *'Travel'*. The need *'Availablity'* refers to the availability of meeting rooms, storage arrangement, the usability of furniture, facilities to meet needs, etc; the *'BuildingOverall'* covers the requirements on building design, cleaning, image to visitors, etc; *'Wellbeing'* refers to the occupant's senses of safety, health, productivity; *'PersonalControl'* is the class to represent the personal control over heating, cooling, lighting, noise and ventilation; *'Requirements'* means the occupant requirements on the speed and effectiveness of response for requesting changes to the heating, lighting, ventilation or air-conditioning/cooling, etc; *'Space'* space at the desk, space at the work area, occupation density, number of occupants in office; *'TimeDuration'* refers to the days of occupants spend in buildings, hours per day of occupants spend at desk or computer; *'Travel'* represents the mode of occupants travel, journey to work, journey to home.

The '*PhysicalNeed*' is comprised of the 5 sub-classes of the *'AcousticNeed', 'IAQNeed', 'ThermalNeed', 'VisualNeed'* and *'Space'*, and the 5 sub-classes are decomposed into more specific sub-classes. *'AcousticNeed'*, includeing*'Interruption*' and '*Noise*', represents the occupants' needs satisfaction with the frequency of unwanted interruption and the noise in buildings. *'IAQNeed'* is the indoor air quality need, comprised of '*AirMovement*' and '*AirQuality*', which means the requirements for air movement and quality in buildings. *'ThermalNeed'* is comprised of 4 sub-classes of '*Humidity*', '*Temperature*', '*TemperatureVariation*' and '*Ventilation*'. *'VisualNeed'* has child classes of '*Glare*' and '*Light*', which express the needs on the amount of light and the glare from the lighting and sun, etc. In addition, the IAQ needs and thermal needs also vary from season to season, with summer and winter being mainly considered in this research. The *'Needs'* class hierarchy is shown in Figure 4-7.

Table 4-4 Needs class hierarchy

Class	SubClass	Subclass	Subclass
Needs	NonPhysicalNeed	Availability	

		BuildingOverall	
		Wellbeing	
		PersonalControl	
		Requirements	
		TimeDuration	
		Travel	
	PhysicalNeed	AcousticNeed	Interruption
			Noise
		IAQNeed	AirMovement
			AirQuality
		ThermalNeed	Humidity
			Temperature
			TemperatureVariation
			Ventilation
		VisualNeed	Glare
			Light
		Space	

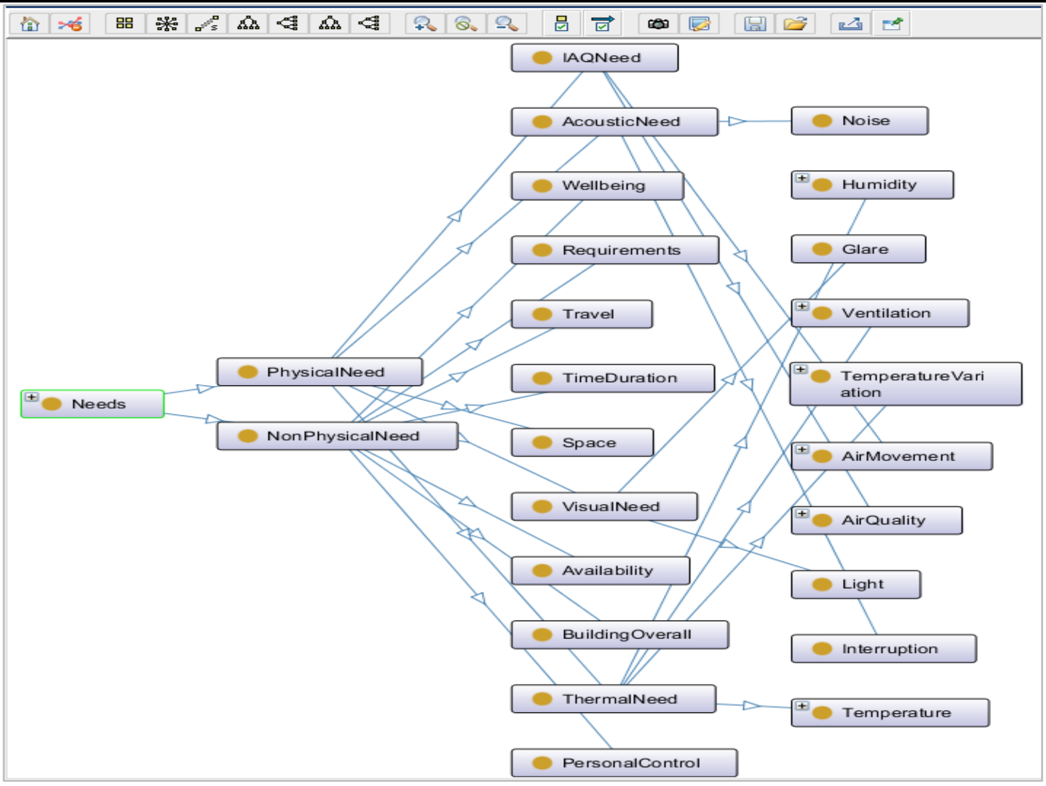


Figure 4-7 Needs class hierarchy structure

The ‘*ParameterRange’* class represents the benchmark, minimum and maximum building occupant needs’ comfort values with respect to building evaluation parameters. As shown in Figure 4-8, the association of ‘*ParameterRange’* class with the class of *‘Needs’* and *‘EvaluationCriteria’* is expressed as the occupants have the comfort value threshold, that is the parameter range. The parameter range of each building evaluation criterion is different.

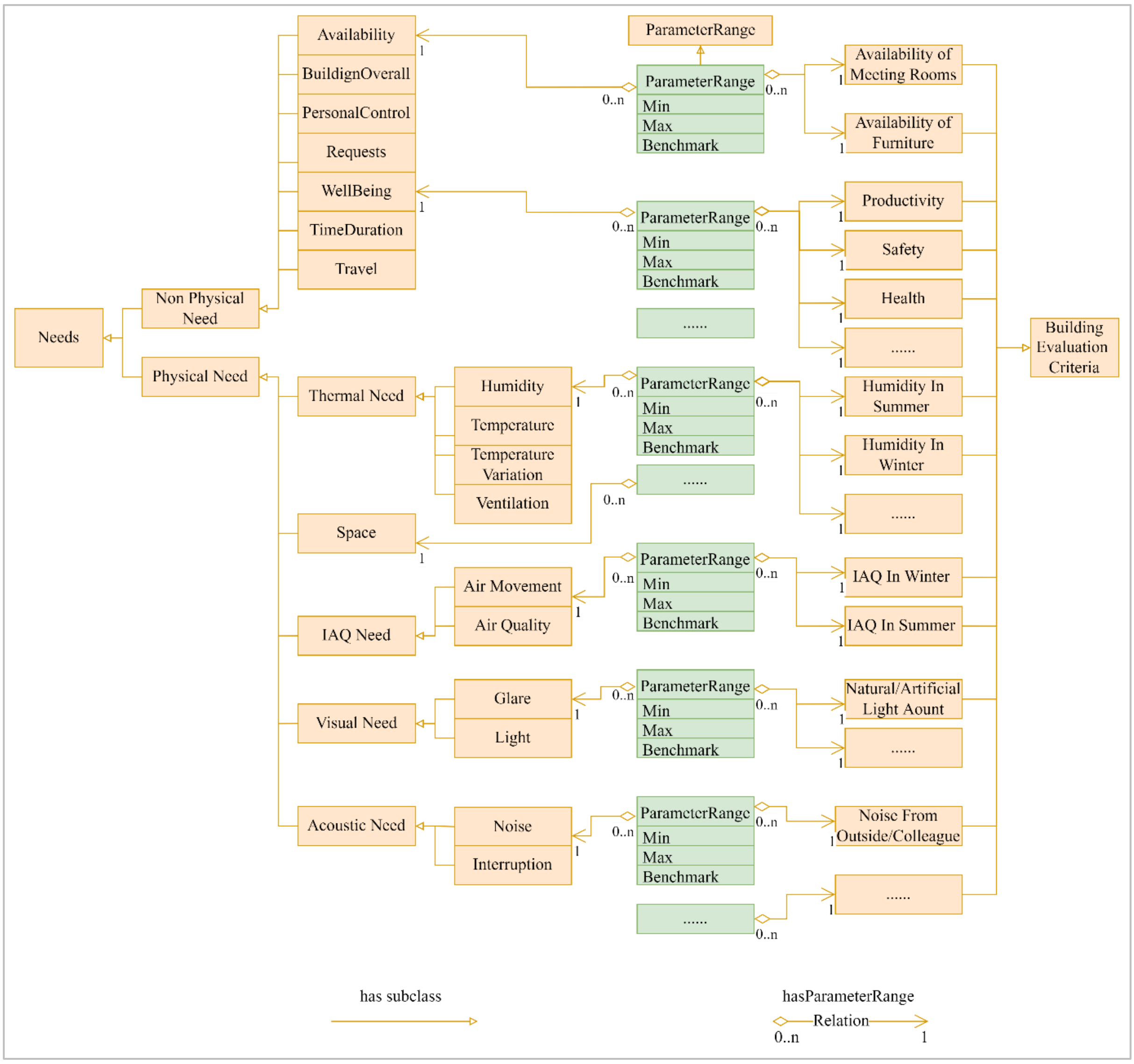


Figure 4-8 Parameter range association with the class of needs and evaluation criteria

As shown in Table 4-5, the ‘*Actions*’ class has three child elements of *‘Feedback’*, *‘BehaviourChange’* and *‘Interaction’*. The *‘Interaction’* class represents the interactions between the people and building systems based on the assessment results to meet the needs of building occupants, including adjusting the settings of the systems, such as turning up/down the set value of temperature, or increasing the frequency of window opening times, etc. The

'BehaviourChange' class represents the behaviours change of building occupants to meet their requirements because of conditions in buildings, for example, using an extra fan in summer or heating generator in winter, or humidifiers for a dry indoor environment. The '*Feedback*' class indicates the occupant's feedback to the facility manager to complain about the conditions in buildings. The class hierarchy and its instances are shown in Figure 4-9.

Table 4-5 Actions class hierarchy

Class	SubClass
Actions	Feedback
	BehaviourChange
	Interaction

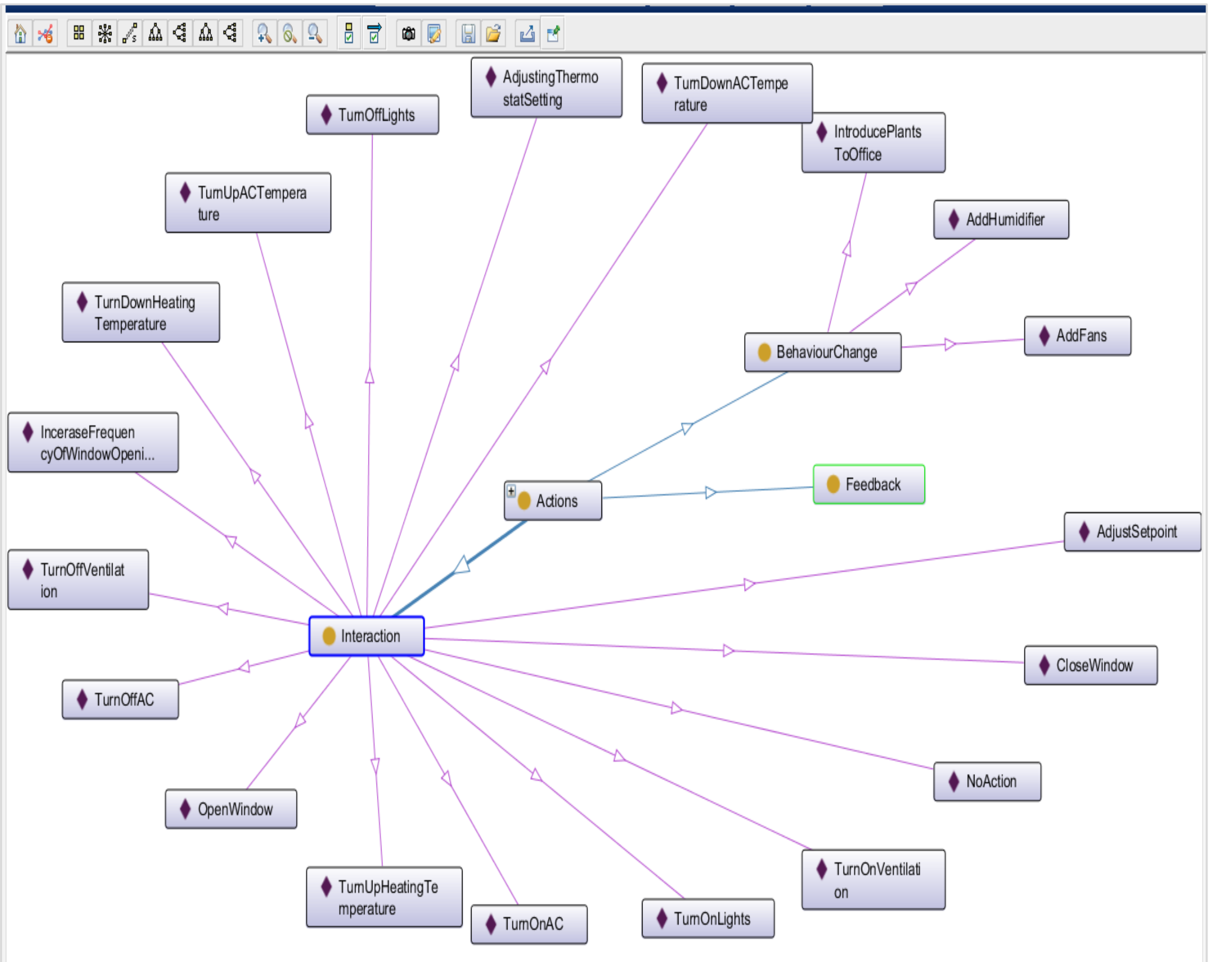


Figure 4-9 Actions class hierarchy with instances structure

The '*Result*' class represents the post-occupancy evaluation results of buildings performances, it has the instances of TooCold, TooHot, LessProdustivity, TooDry, Comfort, VeryWell, TooHumid, Still, Poor, TooLittle, Stable, TooNoisy, Outstanding, etc. For different building performances, the POE results are presented in three levels: Poor, Average, and Outstanding. Poor level means the assessed building performance elements fail to reach the lower boundary of a parameter range description, which leads to a negative impact on building performance assessment results; the average level is the satisfactory level, that is, the results are in the general satisfaction parameters range, outstanding level refers to the result of the assessment building criteria exceeds the upper boundary of parameters range, which adds value to the building performance. The three different assessment levels and their corresponding results and evaluation elements are shown in Table 4-6 below.

Table 4-6 Assessment results levels

Assessment Elements	Poor Level	Average Level	Outstanding Level
Building Design Overall/ Noise/Thermal Comfort/ Conditions In Winter or Summer/ Cleaning, etc	Unsatisfactory	Satisfactory	Outstanding
Perceived Health	Less Healthy	Healthy	More Healthy
Productivity At Work	Less Productivity	Average	More Productivity
Space Using	Used Ineffectively	Used Effectively	Outstanding
Safety/ Image/ Needs/ Usability Furniture/	Poor	Good	Outstanding
Ventilation	Stuffy	Satisfactory	Fresh
Temperature	Too Hot/ Too Cold	Satisfactory	Outstanding
Humidity	Too Dry/ Too Humid	Satisfactory	Outstanding
Air Movement	Still / Draughty	Satisfactory	Outstanding
Indoor Air Quality	Smelly	Satisfactory	Odourless
Temperature variation	Variable	Satisfactory	Stable
Control over Heating / Coolding/Lighting/Noise	No Control	Satisfactory	Full Control
Lighting/ Noise	Too Little/Too Much	Satisfactory	Outstanding

As shown in Figure 4-10, the two classes of *'Actions'* and *'EvaluatedBuilding'* are related to the *'Result'* class, the relationships between them are '*hasResult*' and *'leadTo'*, that is, the buildings from the *'EvaluatedBuilding'* class '*hasResult*' from the *'Result'* class, then the results from the *'Result'* class *'leadTo'* the corresponding actions from the *'Actions'* class. The relations properties of classes are detailed explained in the next section.

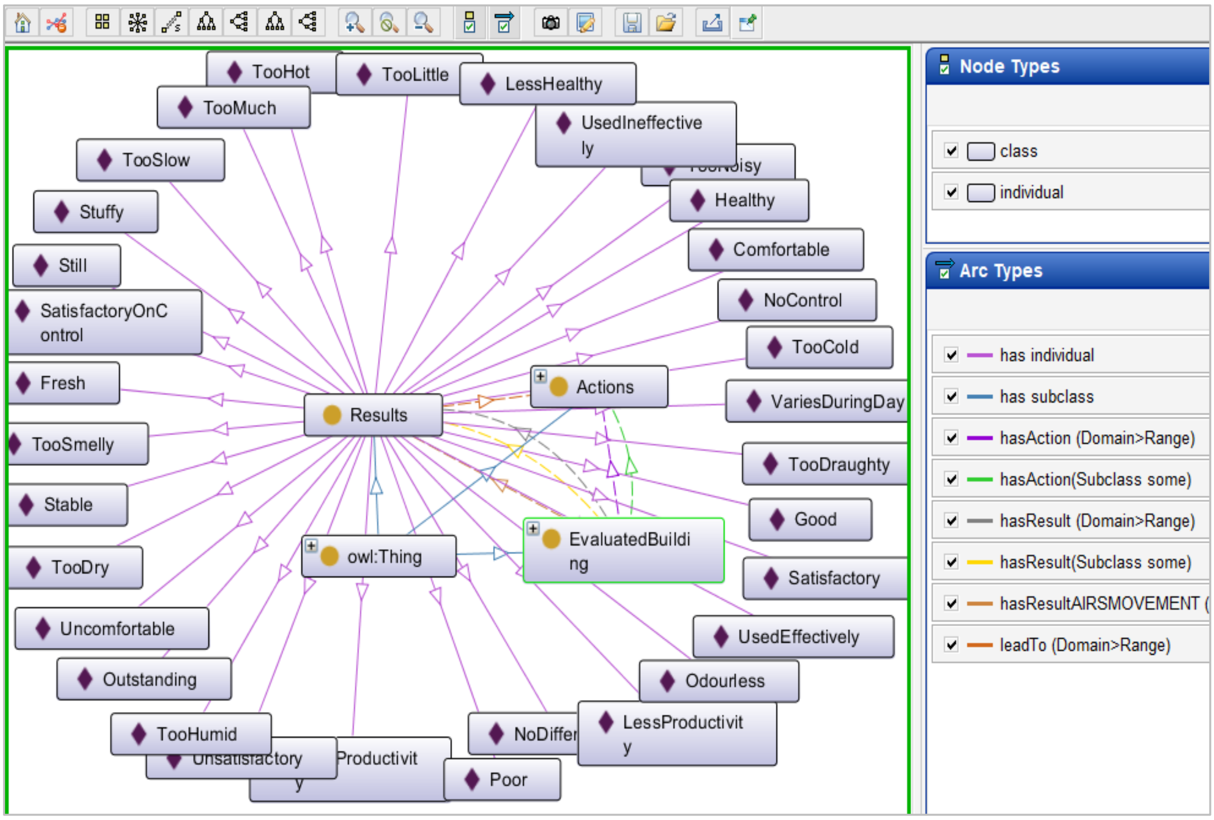


Figure 4-10 Result class structure

The *'BuildingInformation'* class introduces the basic building information including the building types, building address, the country and climate zone, as shown in Table 4-7.

Table 4-7 BuildingInformation class

SuperClass	**SubClass**	**Sub-SubClass**
BuildingInformaiton	BuildingType	DomesticType
		NonDomesticType
	BuildingAddress	
	Country	
	ClimateZone	

As shown in Figure 4-11. The relationships between these classes are indicated with different arc types, the meaning of each arc type is illustrated in the Arc Types tab. For example, the internal relationship between the *'Country'* class and *'ClimateZone'* class is *'hasClimateZone'* property, which is indicated with dashed lines.

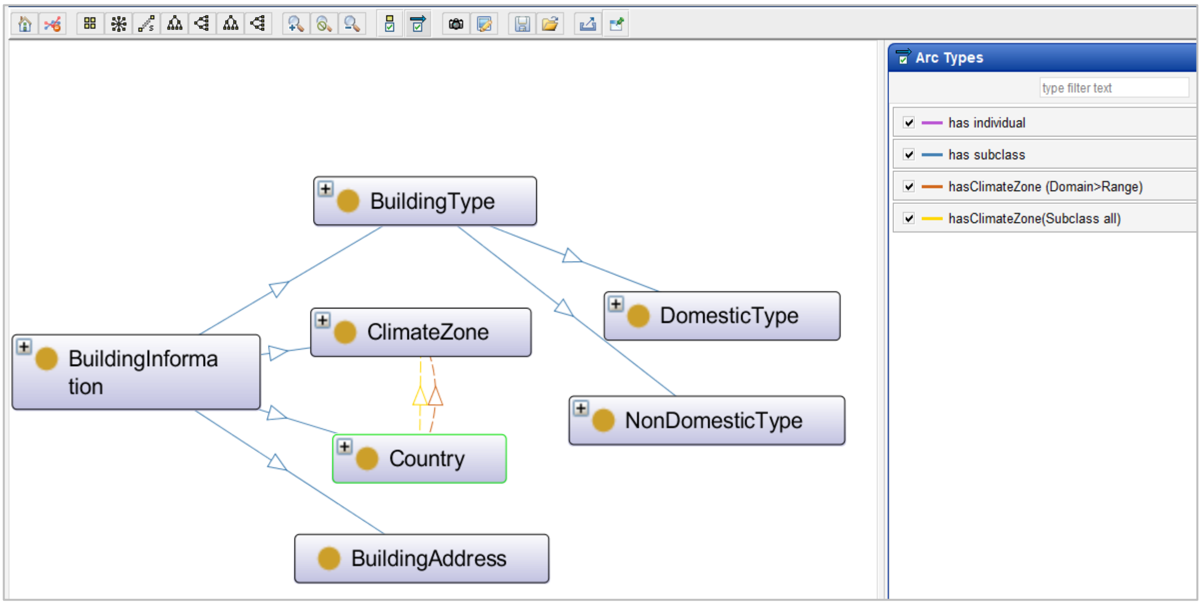


Figure 4-11 Building information class structure

The class *'EvaluationTasks'* represents evaluation tasks including the assessor, the evaluation time, and the evaluation methods. Figure 4-12 shows the structure and some instances.

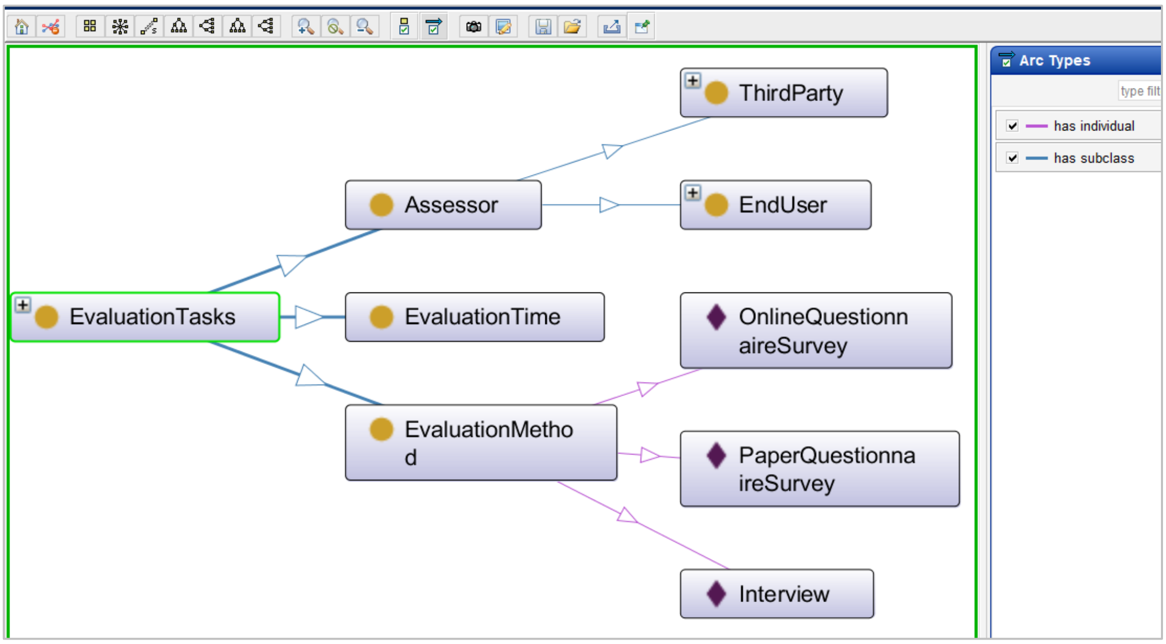


Figure 4-12 Evaluation task class structure

In this model, the methods of the online questionnaire survey, paper-based questionnaire survey and interview are developed as the evaluation methods to conduct a case study to instantiate the proposed ontology model. More information on the case study is presented in Chapter 5.

The *'BuildingSystem'* class represents the building equipment and operating systems that can be adjusted to meet the building occupants' satisfaction needs. This class concept is adopted from the obXML schema, which includes *'WindowControlSystem'*, *'Equipment'*, *'HVACControlSystem'*, *'LightingControlSystem'*, *'Thermostats'*, *'ShadeConytolSytem'*, *'SecurityControlSystem'*. The '*BuildingSystem*' class has multiple relations with other classes, the related classes and the corresponding relation properties are shown in Figure 4-13.

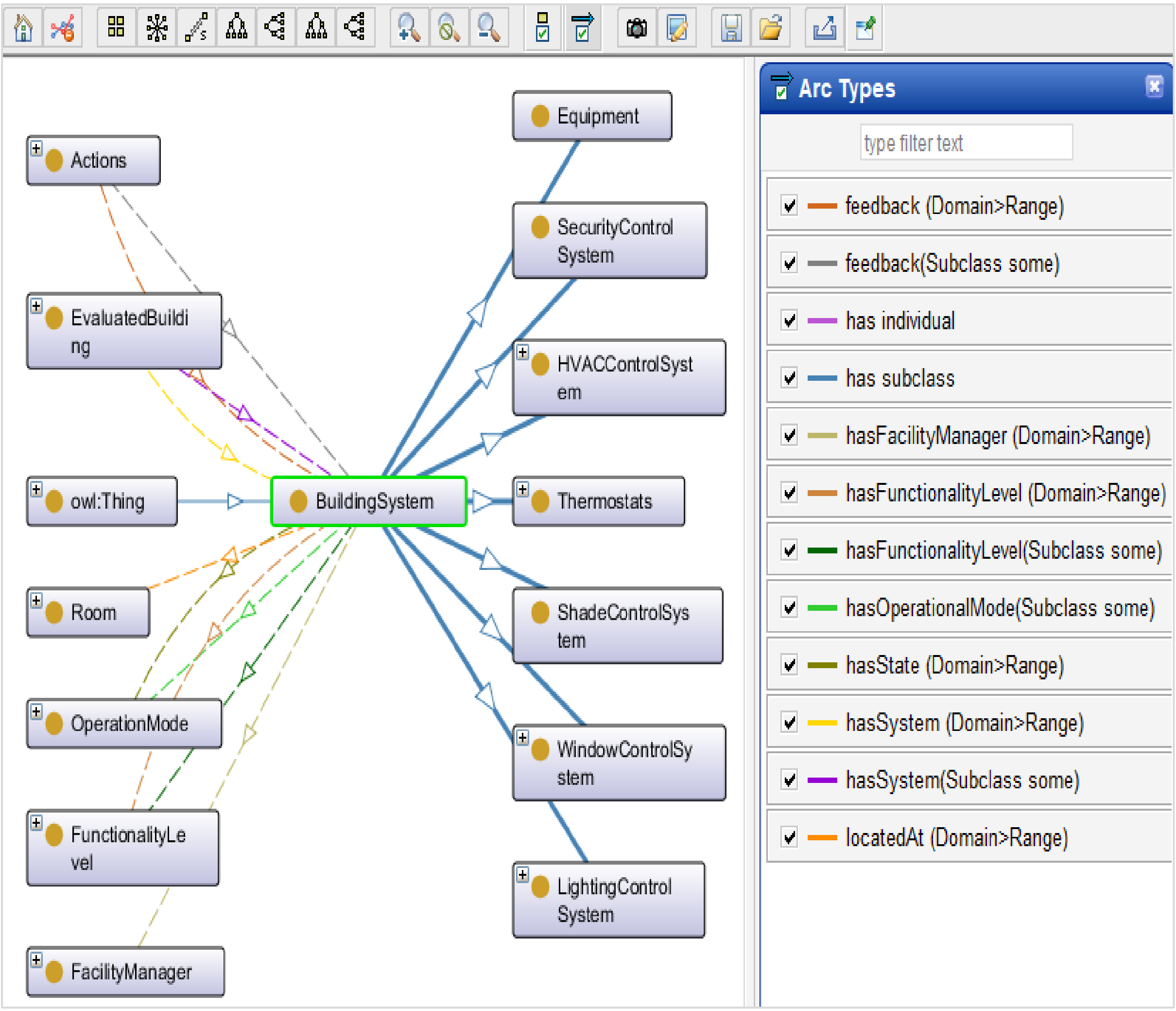


Figure 4-13 Building system class and related relations

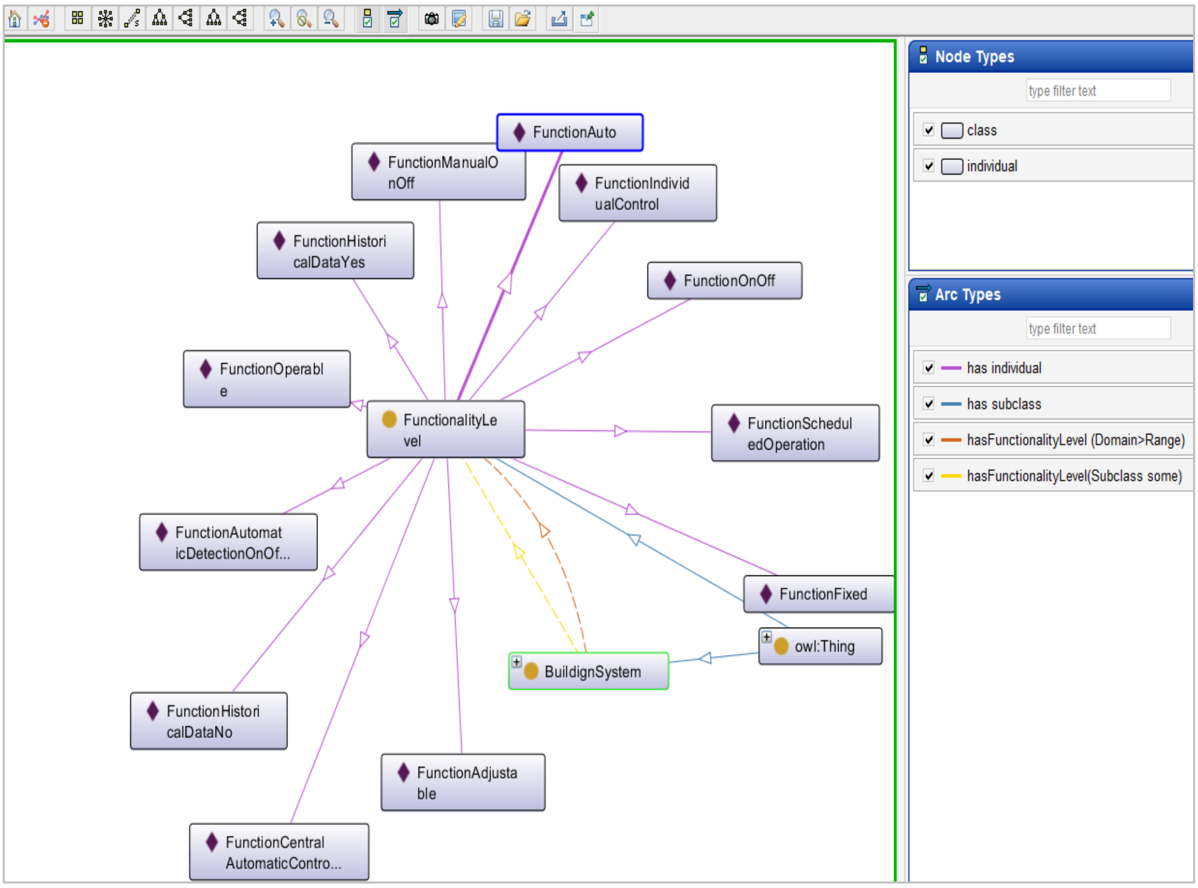


Figure 4-14 Functionality level instances

The ‘*FunctionalityLevel’* class is used to describe the functionality levels of building systems or device operations. This class is integrated with the concepts adopted from the smart readiness indicator (SRI) schema (Ma and Verbeke, 2021). For example, in terms of the heating system, for the heating emission control ability, it can be central automatic control or individual room control; for the heat generator control, it may have an on/off simple model or it has complicated multi-stage scheduled. This functionality level also indicates the smart readiness level of the building systems.

In addition to the above-explained class categories, there are some other simple classes. For example, the *‘Person’* class refers to the people in buildings, including building occupants, facility managers or other people like visitors or cleaners, etc; the *‘Room’* class means the room where building occupants stay; the *‘Season’* class represents the season of the year, Spring, Summer, Autumn, Winter; *‘DayOfWeek’* class refers to the different day of a week, which is relative to the building systems operation mode and the frequency of building occupant using buildings.

So far, all the classes and the class hierarchies have been illustrated in this section. After defining the classes, the internal relationships of concepts have to be described by defining the logical restriction between them, which can help to infer the classes and their instances.

4.2.5 Define the properties of classes-slots

Without defining the properties of classes, the classes alone in the ontology can not provide sufficient information to answer the proposed competency questions. After defining the class hierarchies in the previous step, the properties of the defined classes are analysed in this section, and these proposed properties become slots attached to classes and all subclasses of a class

There are mainly three types of properties introduced here, the object property that defines the internal relations between entities; the data property that defines the relations between entities and the data-type values; and the annotation property that presents the annotations on classes, properties, and individuals, etc. The following sub-sections describe the different types of properties presented in this research.

4.2.5.1 Annotation property

Annotation property is used to present informal documentation annotations on entities, such as classes, properties, individuals, ontology headers, etc. There are five annotation properties are defined by OWL, they are: o*wl:versionInfo*, *rdfs:label*, *rdfs:comment, rdfs:seeAlso, rdfs:isDefinedBy*. For example, the most used *rdfs:comment* annotation property is an instance of *rdf:Property*, which is used to provide a human-readable description of entities (W3C, 2014).

Here is an example of the annotation properties in the developed ontology model of this reserch:
‘‘<!-- http://purl.org/dc/elements/1.1/creator -->

<owl:AnnotationProperty rdf:about="http://purl.org/dc/elements/1.1/creator"/>

<!-- http://purl.org/dc/elements/1.1/title -->

<owl:AnnotationProperty rdf:about="http://purl.org/dc/elements/1.1/title"/>’’

The used ontology annotation properties and ontology prefixes are shown in Figure 4-15.

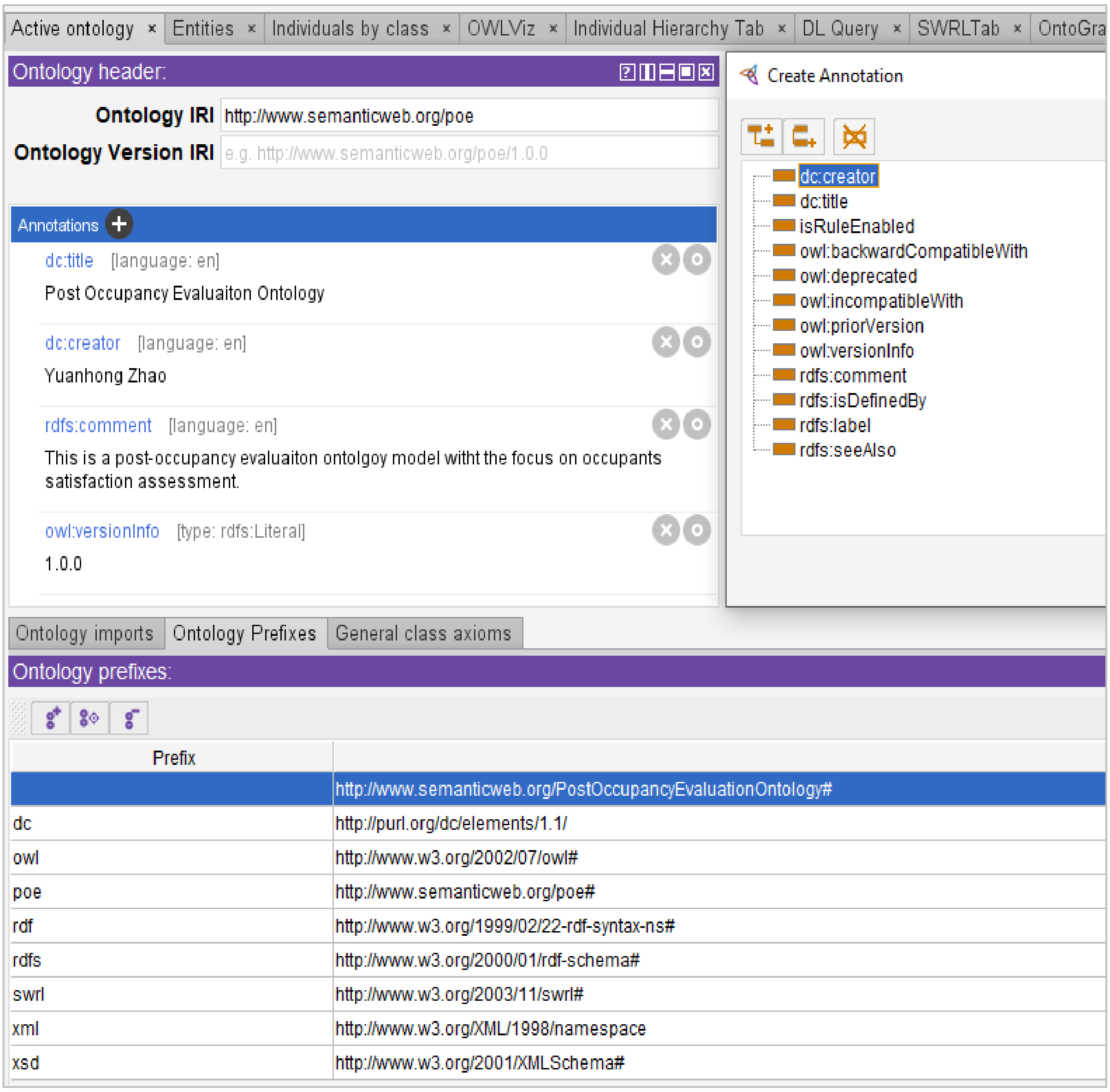


Figure 4-15 Annotation properties

4.2.5.2 Object property

Object property can be regarded as a binary predicate that associates entities with each other. It is often but not always necessary to define the domain and range of object properties. The domain of a class specifies the class of individuals that act as the subjects of each object, and the range of a class specifies the class of individuals that act as objects of the property predicate.

As shown in Figure 4-16, the *owl:topObjectProperty* is the superclass of all the defined object properties. These object properties are used to define the relations between classes and their corresponding individuals, for example, the relation between the class *'EvaluatedBuilding'* and class '*Actions*' is defined by the object property of '*hasAction'*, which can be read as

'*EvaluatedBuilding hasAction Actions'*, the other object properties can be explained in the same manner. The object properties can also be restrained with different characteristics of Functional, Inverse functional, Transitive, etc, where the Functional characteristic means the object property can only have one value for individuals. For example, as the object property of *'inBuilding'* has the domain of *'Room'* class and range of *'EvaluatedBuilding'* class, and has been defined as Functional, that means individuals from class *'Room'* can only have one corresponding individual from class *'EvaluatedBuilding'* as one room can only be in one building.

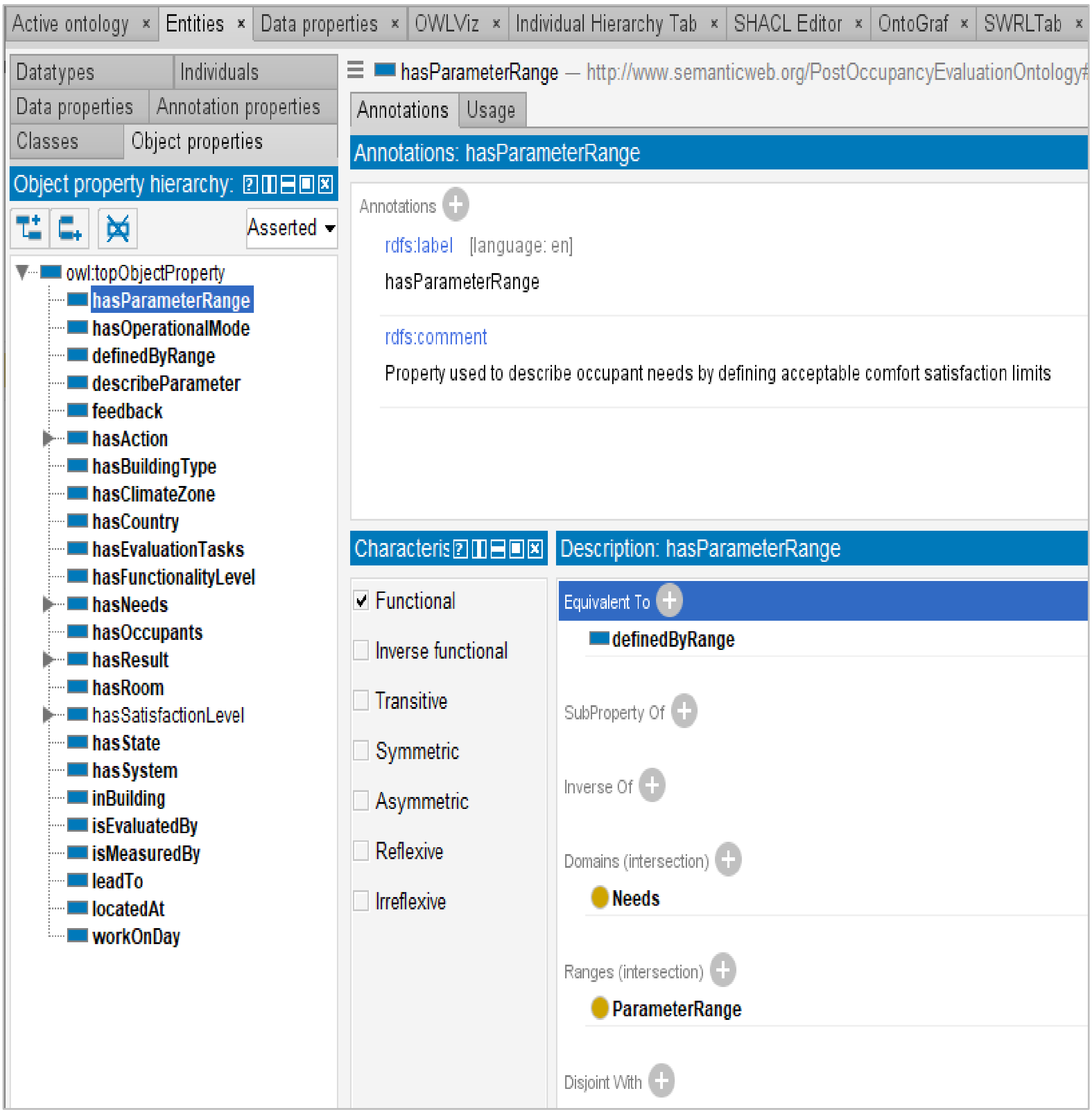


Figure 4-16 The overview of the top object properties hierarchy

As shown in Figure 4-16, the annotation property has been used to present annotations on properties, the *rdfs:comment* is the most used one in this research.

The explanation of each top object property is explained in the following Table 4-8.

Table 4-8 Top object properties

owl:topObjectProperty	**Explanation**
feedback	This property declares the relationship between its domain class '*Actions'* and its range class of *'BuildingSystem'*. The statement is '*Actions feedback BuildingSystem'*, which can be read as the proposed actions are feedback to building systems to adjust or maintain the operation mode to satisfy the building occupants needs.
hasAction	This property defines the relationship between classes of *'EvaluatedBuilding'* and *'Actions'*. The *'EvaluatedBuilding'* class is its domain, and its range is the '*Actions*' class. The statement is read as '*EvaluatedBuildings hasAction Actions'*. This property has a group of sub-properties, such as the *'hasActionSummerIAQ'*, *'hasActionSummerHumidity', 'hasActionWinterIAQ', 'hasActionWinterTemperature',* etc.
hasBuildingType	This property is used to state the relation of the *'EvaluatedBuilding'* class (domain) and the *'BuildingType'* class (range). The statement is *'EvaluatedBuilding hasBuildingType BuildingType'*.
hasClimateZone	This property defines the relationship between the classes of *'Country'* (domain) and *'ClimateZone' (range)*. The statement is expressed as *'Country hasClimateZone ClimateZone'*. And it is a functional property since one country can only have one climate zone.
hasCountry	This property states the relationship between the classes *'EvaluatedBuilding'* (domain) and the *'Country'* (range). It is a functional property since one building can only have one country.

isEvaluatedBy	This property states the relationship between the classes *'EvaluatedBuilding'* and the *'EvaluationCriteria'*. The domain of this property is the *'EvaluatedBuilding'* class and the range is the '*EvaluationCriteria*' class. The statement is *'EvaluatedBuilding isEvaluatedBy EvaluationCriteria'*
hasParameterRange	This property states the relationship between the *'Needs'* class (domain) and the *'ParameterRange'* class (range). It is used to describe the acceptable comfort satisfaction range of the needs.
describeParameter	This property is used to describe the relationship between the class of *'ParameterRange'* (domain) and the *'EvaluationCriteria'*(range). It declares the function of the parameter range class is to describe the building evaluation parameters.
hasFunctionalityLevel	This object property is used to align a certain functionality level with the building systems. Its domain is the *'BuildingSystem'* class and the range is the *'FunctionalityLevel'* class. The statement is *'BuildingSystem hasFunctionalityLevel FunctionCentralAutomaticControl'*.
hasNeeds	This property defines the relationship between classes of *'BuildingOccupants'* and *'Needs'*. The domain of this property is the *'BuildingOccupants'* class and the range is the *'Needs'* class. The statement is read as '*BuildingOccupants hasNeeds Needs'*. It has a set of sub-properties, such as *'hasNeedAirQuality', 'hasNeedHumidity', 'hasNeedLight', 'hasNeedTemperature'*, etc.
hasOccupants	This property is used to define the relationship between the class of *'Room'* (domain) and the class of *'BuildignOccupants'* (range). The statement is '*Room hasOccupants BuildingOccupants'*.
hasResult	This object property is used to state the relationship between the class *'EvaluatedBuilding'* and the class *'Results'*. It has the domain of *'EvaluatedBuilding'* class and the range of *'Results'*

	class. The expression is *'EvaluatedBuilding hasResult Results'*. It has a series of sub-properties, such as the *'hasResultAIRSMOVEMENT', 'hasResultCLEANING', 'hasResultSAFETY', 'hasResultSIAQ', 'hasResultTSHOT' 'hasResultSHUMIDITY', etc.*
hasRoom	This object property is used to state the relationship between the class *'Person'* and the class *'Room'*. It has the domain of *'Person'* class and the range of *'Room'* class. The expression is *'Person hasRoom Room'.*
hasState	This object property is used to align a certain operation state with the building systems. Its domain is the *'BuildingSystem'* class and the range is the *'OperationMode'* class. The statement is *'BuildingSystem hasState OperationMode'.*
hasSystem	This property states the relationship between the classes *'EvaluatedBuilding'* (domain) and the *'BuildingSystem'* (range). The statement is *'EvaluatedBuilding hasSystem BuildingSystem'.*
inBuilding	This object property defines the relationship between the classes of *'Room'* and *'EvaluatedBuilding'*. It has the domain of *'Room'* class and the range of *'EvaluatedBuilding'* class. It has been defined as Functional, which means the individuals from the class of *'Room'* can only have one corresponding individual from the class of *'EvaluatedBuilding'* since one room can only be in one building.
locatedAt	This property states the relationship between the class *'BuildingSystem'* (domain) and the class of '*Room'* (range). The statement is *'BuildingSystem locatedAt Room'.*
workOnDay	This object property defines the relationship between the class of *'BuildignOccupants'* (domain) and the class of *'DayOfWeek'* (range).
isMeasuredBy	This object property is used to align evaluation criteria with the needs. Its domain is the *'Needs'* class and the range is the

	'EvaluationCriteria' class. The statement is *'Needs isMeasuredBy EvaluationCriteria'*.
leadTo	This object property is used to illustrate the relationship between the class *'Results'* and the class *'Actions'*. It has the domain of the *'Results'* class and the range of *the 'Actions'* class. The expression is *'Results leadTo Actions'*.

The *'hasNeeds'* object property has the domain of the *'BuildingOccupants'* class and the range of the *'Needs'* class. The *'hasNeeds'* object property has a set of sub-properties as shown in Figure 4-17, which means the building occupants have needs on air movement, air quality, temperature, availability, senses, humidity, light, personal control, glare, requests, noise, temperature variation, travel, ventilation, time duration, interruption, and space.

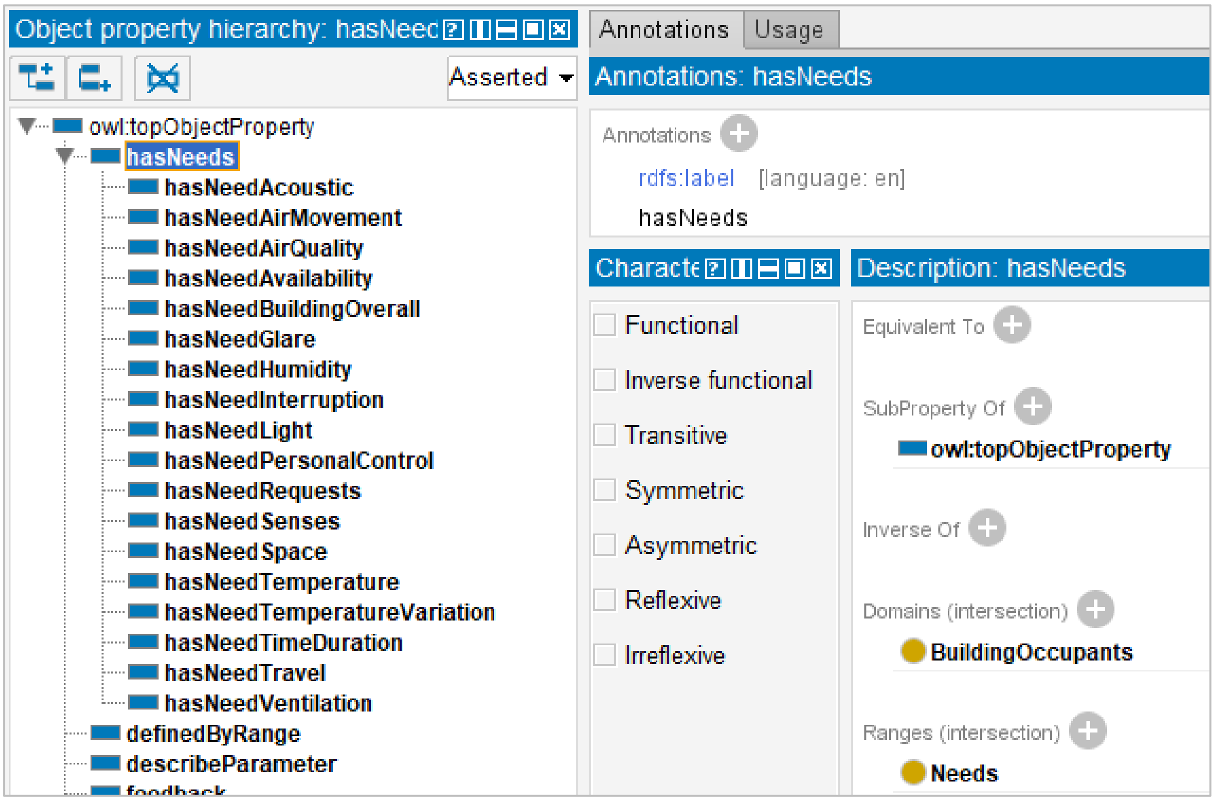


Figure 4-17 The 'hasNeeds' object property hierarchy

The *'hasResult'* property has the domain of the *'EvaluatedBuilding'* class and the range of the *'Results'* class. The sub-properties of the *'hasResult'* property is shown in Figure 4-18. The annotation property has been used for annotations of each property.

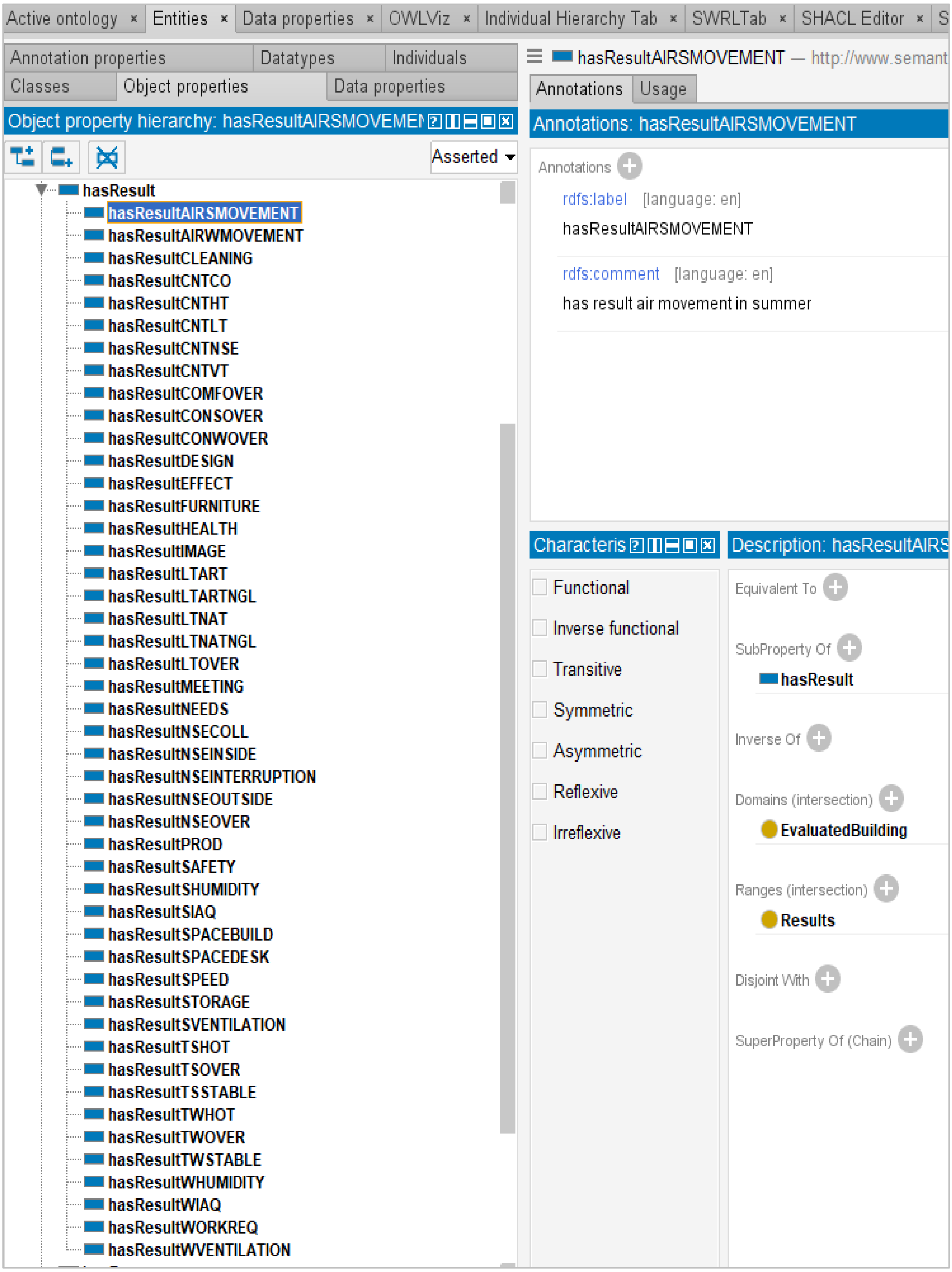


Figure 4-18 The sub-properties of the 'hasResult' object property

The explanation of sub-properties of the '*hasResult*' object property is shown in Table 4-9.

Table 4-9 The explanation of 'hasResult' sub-properties

Sub-properties	Explanation
hasResultAIRSMOVEMENT/ hasResultAIRWMOVEMENT	has result on air movement in summer (AIRSMOVEMENT)/ winter (AIRWMOVEMENT)
hasResultCLEANING	Has result on cleaning (CLEANING)
hasResultCNTCO/ hasResultCNTHT/ hasResultCNTLT/ hasResultCNTNSE /hasResultCNTVT	Has result personal control over cooling (CNTCO)/ heating (CNTHT)/ lighting (CNTLT)/ noise (NSE)/ ventilation (CNTVT)
hasResultCOMFOVER	has result on overall comfort within the building environment (COMFOVER)
hasResultCONSOVER/ hasResultCONWOVER	has result on conditions in summer overall (CONSOVER)/ winter overall (CONWOVER)
hasResultDESIGN	has result on design overall (DESIGN)
hasResultEFFECT	has result on effectiveness of response to requests (EFFECT)/speed of response to requests (SPEED)
hasResultFURNITURE	has result on usability of furniture (FURNITURE)
hasResultHEALTH	has result on perceived health (HEALTH)
hasResultIMAGE	has result on image to visitors (IMAGE)
hasResultLTART/ hasResultLTNAT/ hasResultLTOVER	has result on amount of artificial light (LTART)/ natural light (LTNAT)/ lighting overall (LTOVER)
hasResultLTARTNGL/ hasResultLTNATNGL	has result on glare from lights (LTARTNGL)/ sun and sky (LTNATNGL)
hasResultMEETING	has result on availability of meeting rooms (MEETING)
hasResultNEEDS	has result on facilities meet needs (NEEDS)
hasResultNSECOLL/ hasResult NSEOUTSIDE/ hasResult NSEINSIDE/ hasResultNSEOVER	has result on noise from colleagues (NSECOLL)/ outside (NSEOUTSIDE)/ other noise from inside (NSEINSIDE) / noise overall (NSEOVER)

hasResultNSEINTERRUPTION	has result on frequency of unwanted interruptions (NSEINTERRUPTION)
hasResultPROD	has result on productivity at work (PROD)
hasResultSAFETY	has result on personal safety (SAFETY)
hasResultSHUMIDITY/ hasResultWHUMIDITY	has result on the humidity in summer (SHUMIDITY) / winter (WHUMIDITY)
hasResultSIAQ/ hasResultWIAQ	has result on indoor air quality in summer (SIAQ) / winter (WIAQ)
hasResultSPACEBUILD	has result on building space use (SPACEBUILD)
hasResultSPACEDESK	has result on the adequacy of space at work area (SPACEDESK)
hasResultSTORAGE	has result on storage arrangements (STORAGE)
hasResultSVENTILATION/ hasResultWVENTILATION	has result on ventilation in summer (SVENTILATION) / winter (WVENTILATION)
hasResultTSHOT/ hasResultTWHOT	has result on the temperature in summer (TSHOT)/ winter (TWHOT)
hasResultTSOVER/ hasResultTWOVER	has result on thermal comfort in summer (TSOVER)/ winter (TWOVER)
hasResultTSSTABLE/ hasResultTWSTABLE	has result on temperature variation in summer (TSSTABLE)/ winter (TWSTABLE)
hasResultWORKREQ	has result on how well facilities meet needs on work (WORKREQ)

4.2.5.3 Data Property

The data property is a binary predicate that connects entities with literal data. There are many datatypes of data properties such as string, integer, Boolean, float, rational or decimal, etc. The '*owl:topDataProperty'* is the superclass of all the data properties (W3C, 2014). Normally, the attributes are often defined in datatype properties (Zhang and Issa, 2013). In this research, the defined data properties are shown in Figure 4-19, and the explanation of the top data properties is given in Table 4-10.

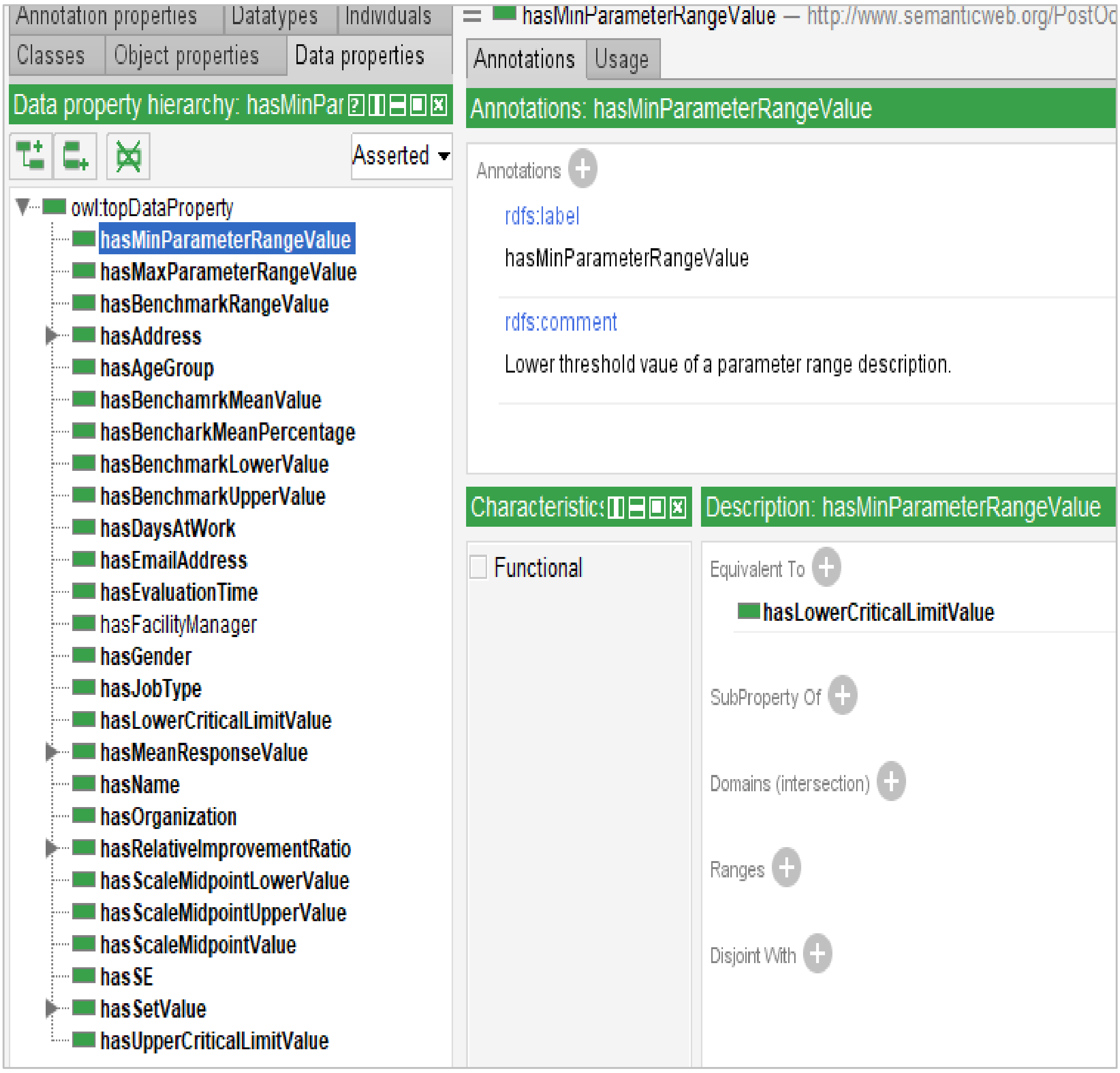


Figure 4-19 The overview of the top data property hierarchy

Table 4-10 Data properties explanation

Owl:topDataProperty	Explanation
hasMeanResponseValue	This property is used to align the evaluated building values with certain evaluation criteria individuals. For this data property, the domain is the *'EvaluatedBuilding'* class, and the range is xsd:decimal. According to the evaluation criteria, this data property has a group of corresponding sub-properties. For example, *'BuildingEasternGateway hasMeanResponseValueSIAQ*

	4.58', or 'BuildingEasternGateway hasMeanResponseValueAIRWMOVEMENT 3.21', etc.
hasScaleMidpointValue	This data property is used to assign a pre-determined midpoint value to every evaluation criteria. The domain is the *'EvaluationCriteria'* class, and the range is xsd:decimal. For example, *'CriteriaCleaning hasScaleMidpointValue 4.00', 'CriteriaPersonalSafety hasScaleMidpointValue 4.00, etc.*
hasSE	This data property is used to assign a pre-determined Standard Error (SE) value to certain evaluation criteria. The domain is the *'EvaluationCriteria'* class, and the range is xsd:decimal. For example, '*CriteriaAirMovementInSummer hasSE 0.07', 'CriteriaTemperatureInWinter hasSE 0.06', 'CriteriaThermalComfortInWinter hasSE 0.09', etc.*
hasScaleMidpointLowerValue	This data property is used to assign a scale midpoint lower value to certain evaluation criteria. The domain is the *'EvaluationCriteria'* class, and the range is xsd:decimal. In this research, the value of scale midpoint lower value is calculated by the scale midpoint value and Standard Error (SE) value of certain evaluation criteria. It is the value of the scale midpoint mean -1.96 X SE. The calculation of this value is defined in SWRL rules in the ontology model. For example, *'CriteriaPerceivedHealth hasScaleMidpointLowerValue 3.87, etc.*
hasScaleMidpointUpperValue	This data property is used to assign a scale midpoint upper value to certain evaluation criteria. The domain is the *'EvaluationCriteria'* class, and the range is xsd:decimal. In this research, the value of scale midpoint upper value is calculated by the scale midpoint value and Standard Error (SE) value of certain evaluation criteria. It is the value of the scale midpoint mean +1.96 X SE.

	For example, *'CriteriaPerceivedHealth hasScaleMidpointUpperValue 4.13, etc.*
hasBenchamrkMeanValue	This data property is used to assign a pre-determined benchmark mean value to certain evaluation criteria. The domain is the *'EvaluationCriteria'* class, and the range is xsd:decimal. For example, *'CriteriaPerceivedHealth hasBenchamrkMeanValue 3.74, 'CriteriaPersonalSafety hasBenchamrkMeanValue 5.95, etc.*
hasBencharkMeanPercentage	This data property is used to assign a benchmark mean percentage value to certain evaluation criteria. The domain is the *'EvaluationCriteria'* class, and the range is xsd:decimal. For example, *'CriteriaPersonalSafety hasBencharkMeanPercentage 47.24'*.
hasBenchmarkLowerValue	This data property is used to assign a benchmark lower value to certain evaluation criteria. The domain is the *'EvaluationCriteria'* class, and the range is xsd:decimal. In this research, the value of benchmark lower value is calculated by the benchmark mean value and the Standard Error (SE) value of certain evaluation criteria. It is the value of the benchmark mean -1.96 X SE. The calculation of this value is defined in SWRL rules in the ontology model. For example, *'CriteriaPerceivedHealth hasBenchmarkMeanLowerValue 3.61, etc.*
hasBenchmarkUpperValue	This data property is used to assign a benchmark upper value to certain evaluation criteria. The domain is the *'EvaluationCriteria'* class, and the range is xsd:decimal. In this research, the value of the benchmark lower value is calculated by the benchmark mean value and the Standard Error (SE) value of certain evaluation criteria. It is the value of the benchmark mean +1.96 X SE. The calculation of this value is defined in SWRL rules in the

	ontology model. For example, *'CriteriaPerceivedHealth hasBenchmarkMeanUpperValue 3.87, etc.*
hasRelativeImprovementRatio	This data property is used to describe the relative improvement ratio for the building in specific evaluation criteria. It describes the performance result gap between the evaluated buildings and the benchmark building database, also this value represents the potential improvement capability of evaluated buildings and also shows the impact degree of certain building performance on building occupants' satisfaction. It is defined by the response mean value of evaluated buildings and the benchmark mean value of evaluation criteria. It has a set of sub-data properties corresponding to the evaluation criteria. The domain is the *'EvaluatedBuilding'* class, and the range is xsd:decimal. For example, *'Evaluated BuildinghasRelativeImprovementRatioIAQ', hasRelativeImprovementRatioSAFETY etc.*
hasLowerCriticalLimitValue	This data property assigns a lower critical limit value to certain evaluation criteria. The domain is the *'EvaluationCriteria'* class, and the range is xsd:decimal. The value of this data property for each evaluation criteria is the lower value between the scale midpoint lower value and the benchmark lower value. For example, *'CriteriaLightingOverall hasLowerCriticalLimitValue 3.86', 'CriteriaHumidityInWinter hasLowerCriticalLimitValue 3.24' , etc.*
hasUpperCriticalLimitValue	This data property assigns an upper critical limit value to certain evaluation criteria. The domain of this property is the *'EvaluationCriteria'* class, and the range is xsd:decimal. The value of this data property for each evaluation criteria is the upper value between the scale

	midpoint upper value and the benchmark upper value. For example, *‘CriteriaLightingOverall hasUpperCriticalLimitValue 5.22’, ‘CriteriaHumidityInWinter hasUpperCriticalLimitValue 4.11’ , etc.*
hasMinParameterRangeValue	This property is applied to describe the entities in ‘*Needs’* class, it represents the minimum acceptable comfort range of needs. This attribute is equivalent to *hasLowerCriticalLimitValue.*
hasMaxParameterRangeValue	This property is applied to describe the entities in ‘*Needs’* class, it represents the maximum acceptable comfort range of needs. This attribute is equivalent to *hasUpperCriticalLimitValue.*
hasBenchmarkRangValue	This property is applied to describe the entities in ‘*Needs’* class, it represents the benchmark value that most of the buildings achieved in terms of meeting occupants' needs. This attribute is equivalent to *hasBenchamrkMeanValue.*
hasDaysAtWork	This property represents the days of the building occupants at work. The domain of this property is the *‘BuildingOccupants’* class, and the range is xsd:integer.
hasEmailAddress	This data property is used to define the email address of the person. The domain of this property is the *‘Person’* class, and the range is xsd:string. For example, *‘FacilityManager John hasEmailAddress John@gmail.com’.*
hasEvaluationTime	This data property is used to state the evaluation time of conduction of building assessment. The domain of this property is the *‘EvaluatedBuilding’* class, and the range is xsd:date. For example, *‘BuildingEasternGateway hasEvaluationTime 2020-10-01 ’.*
hasGender	This data property is used to state the gender of the person. The domain of this property is the *‘Person’* class,

	and the range is xsd:string. For example, *'Occupant_A hasGender Female'*.
hasJobType	This data property is used to state the job type of the person. The domain of this property is the *'Person'* class, and the range is xsd:string. For example, *'Occupant_A hasJobType DoctoralResearcher'*.
hasName	This data property is used to state the name of the person. The domain of this property is the *'Person'* class, and the range is xsd:string.
hasOrganization	This property is used to state the organization that the assessor works for. The domain of this property is the *'Assessor'* class, and the range is xsd:string.
hasSetValue	This data property is used to define the default set value of the different building systems or equipment. The domain of this property is the *'BuildingSystem'* class, and the range is xsd:string.
hasID	This data property is used to define the unique ID of the different building systems or equipment.
hasAddress	This data property is used to define the address of the evaluated buildings. The domain of this property is the *'EvaluatedBuilding'* class, and the range is xsd:string.
hasAgeGroup	This data property is used to state the age group of the person. The domain of this property is the *'Person'* class, and the range is xsd:string. For example, *'Occupant_A hasAgeGroup Over30'*.

The *'hasMeanResponseValue'* and *'hasRelativeImprovementRatio'* properties are defined to align evaluated building values with certain evaluation criteria individuals. Corresponding to the different evaluation criteria, a set of sub-properties of these two properties are defined to assign assessment values to the corresponded criteria. As shown in Figure 4-20, including the '*hasMeanResponseValueSIAQ', 'hasRelativeImprovementRatioSIAQ', etc.*

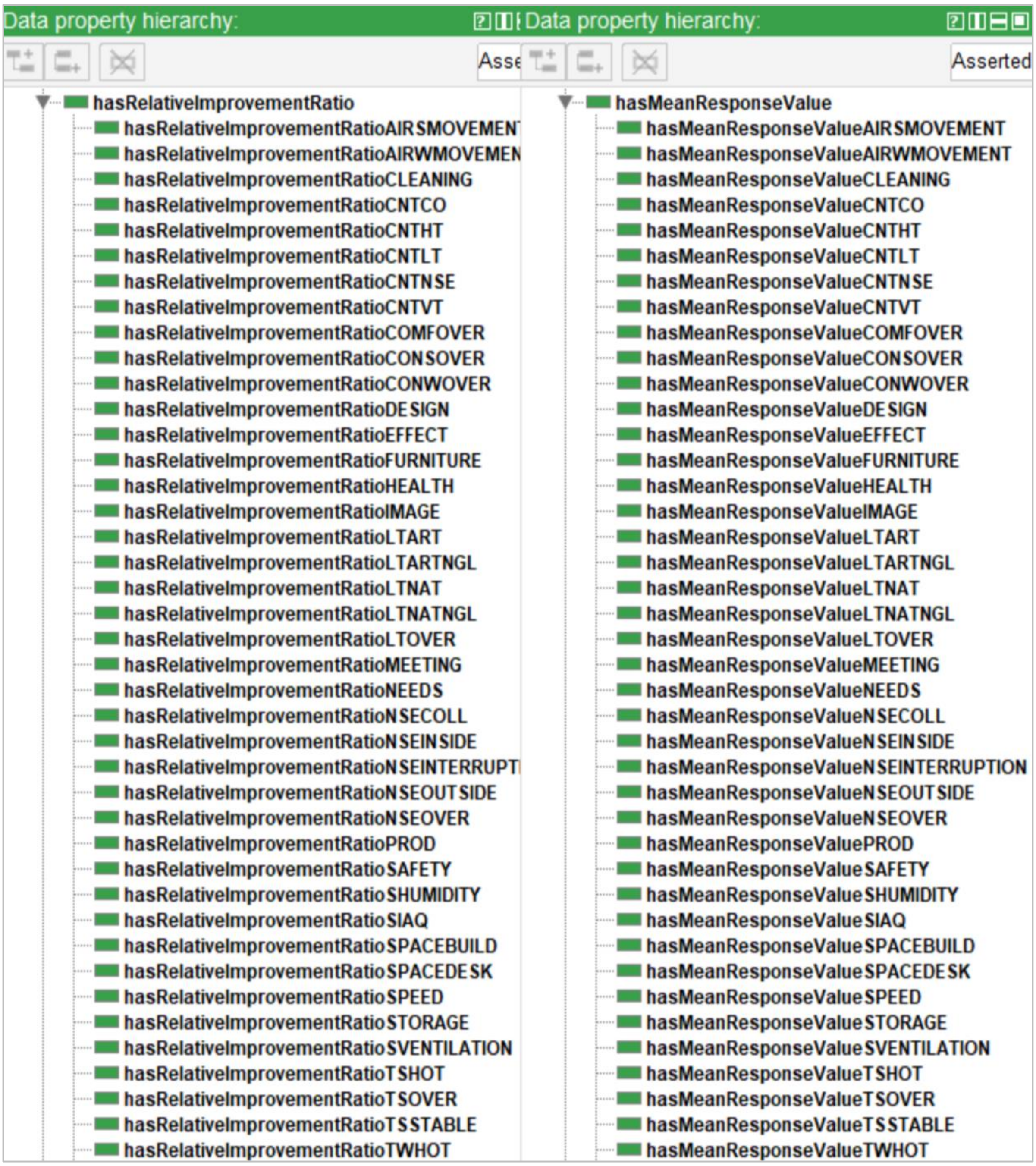


Figure 4-20 The hasMeanresponseValue and hasRelativeImprovementRatio data properties

4.2.6 Define the facets of the slots

The facets of slots can also be explained as the restrictions of properties, including the properties' value type, allowed value, the cardinality of the values, and so on. The cardinality facet defines the number of values that properties can have; the value type facet includes string, boolean, instance, etc. The allowed values can be a specific numerical value. The determination of instance type facet includes defining the domain and range of slots. For example, as shown in Figure 4-21, the *CriteriaAirMovementInSummer* has a value of hasSE 0.07, the type of the value is decimal. The *CriteriaAirMovementInSummer* has *exactly* 1 *hasLowerCriticalLimitValue*, here, the cardinality restriction of the value is 1.

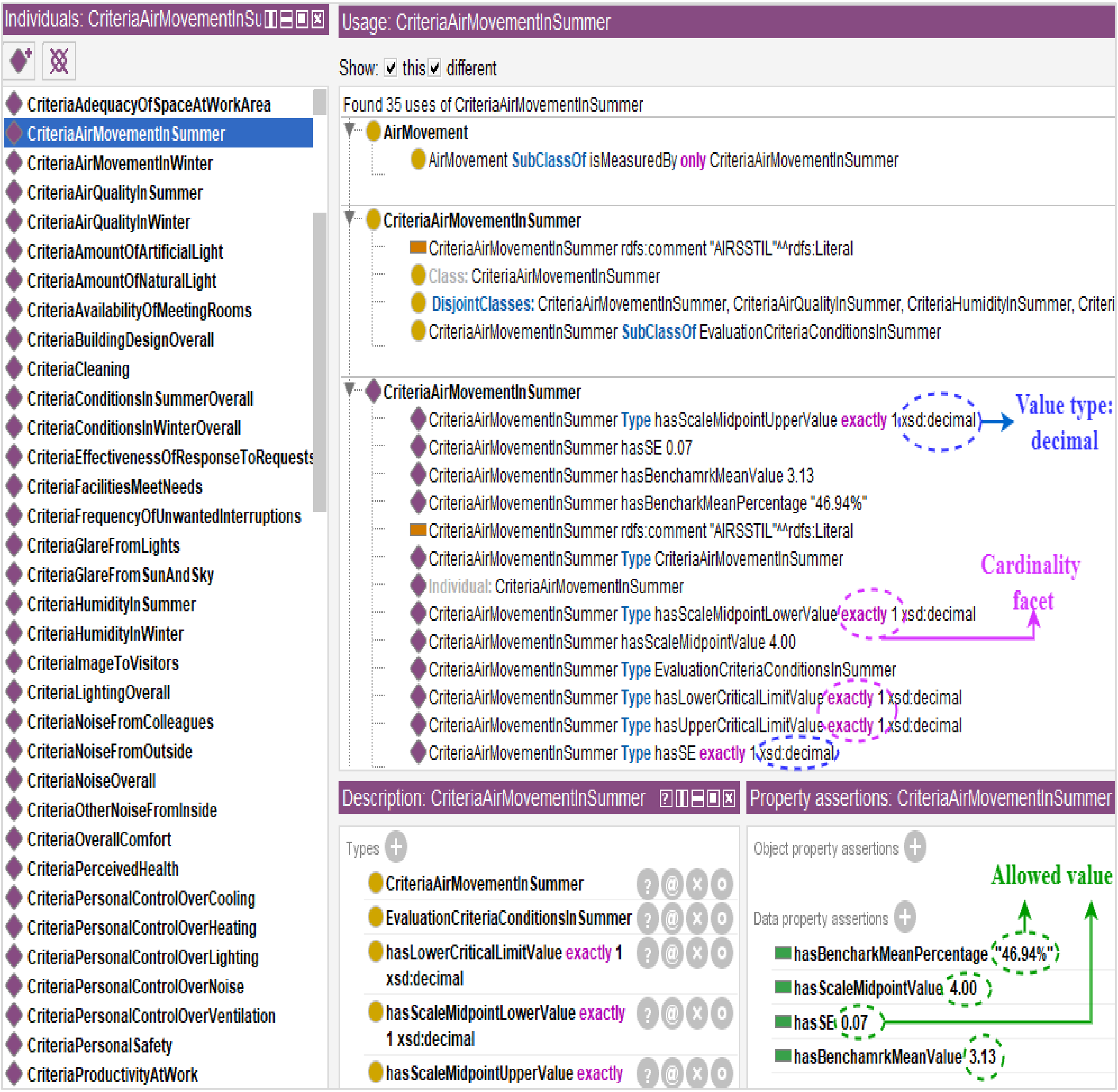


Figure 4-21 Facets of the slots

4.2.7 Create instances

The last step is to create the individual instances of the classes in ontology. The instances of classes can be defined by the following steps: choosing a class first, then creating individual instances for the chosen class, and at the end to define the slot values for the instances. As shown in Figure 4-22, the individual instances are defined for the classes, and the values of instances' object properties and data properties are defined as well. For example, the

'EvaluatedBuilding' class has two instances in this model, *'BuildingEasternGateway'* and *'BuildingMichaelSterling'*. The object properties and the values of data properties are defined, such as the instance of *'BuildingEasternGateway'* has a data property of *'hasMeanresponseAIRMOVEMENT'*, the value of this property is *'3.21'*.

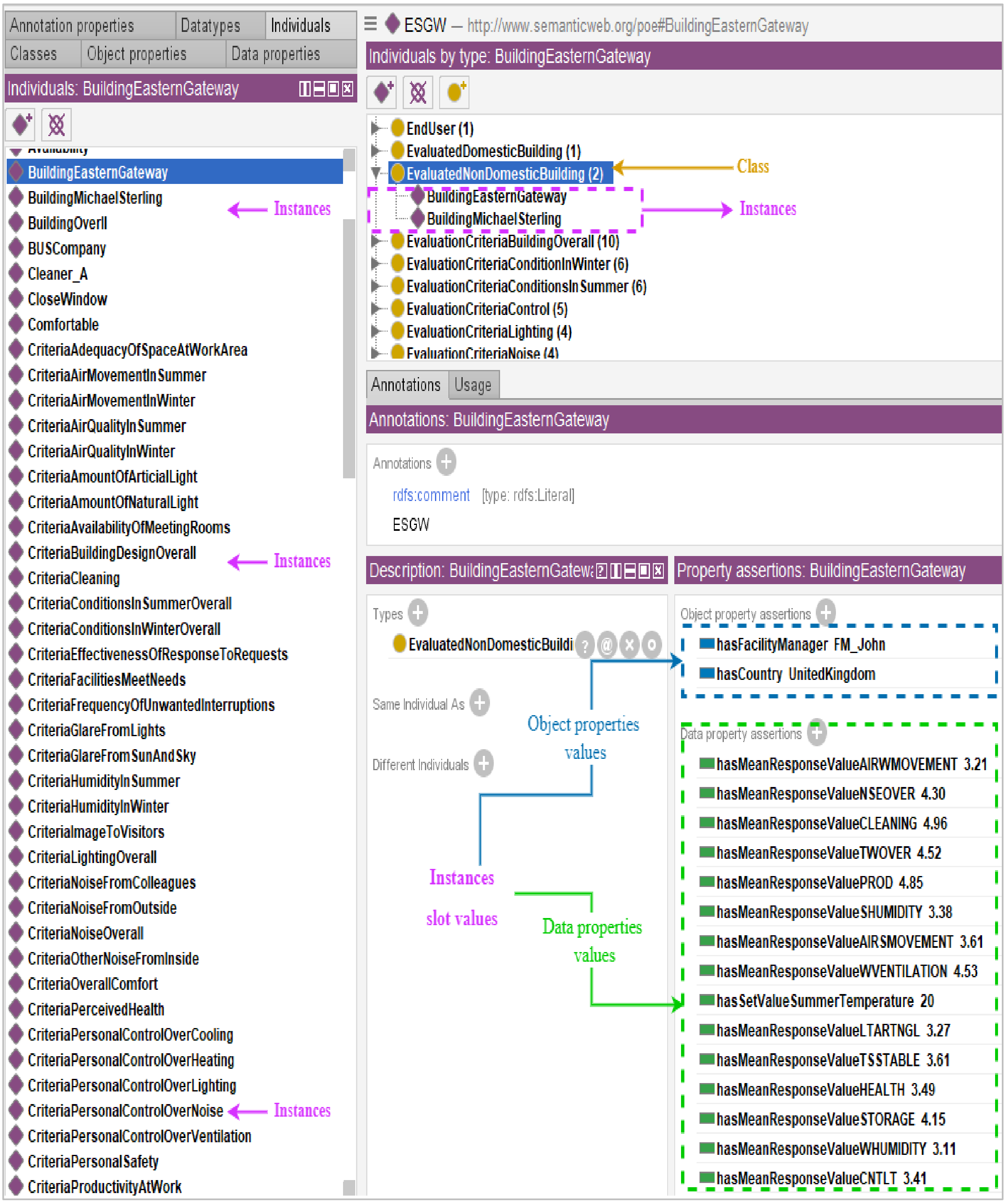


Figure 4-22 Instances of ontology and slots values

So far, the basic knowledge base of POE ontology has been built by following the methodology from Noy and McGuinness (2001), however, this is still a lightweight ontology without developing SWRL rules and SQWRL rules into the ontology model, so the next task is to develop semantic rules into ontology.

4.2.8 Overall relations between entities

So far, the knowledge base of the proposed POE ontology has been established from scratch by following the conventional '7-Steps' ontology development methodology (Noy and McGuinness, 2001). The overall structure between the main classes and properties is shown in Figure 4-23. To achieve the reasoning and query capabilities of this heavyweight ontology, the SWRL rules and SQWRL query rules are developed in the next section

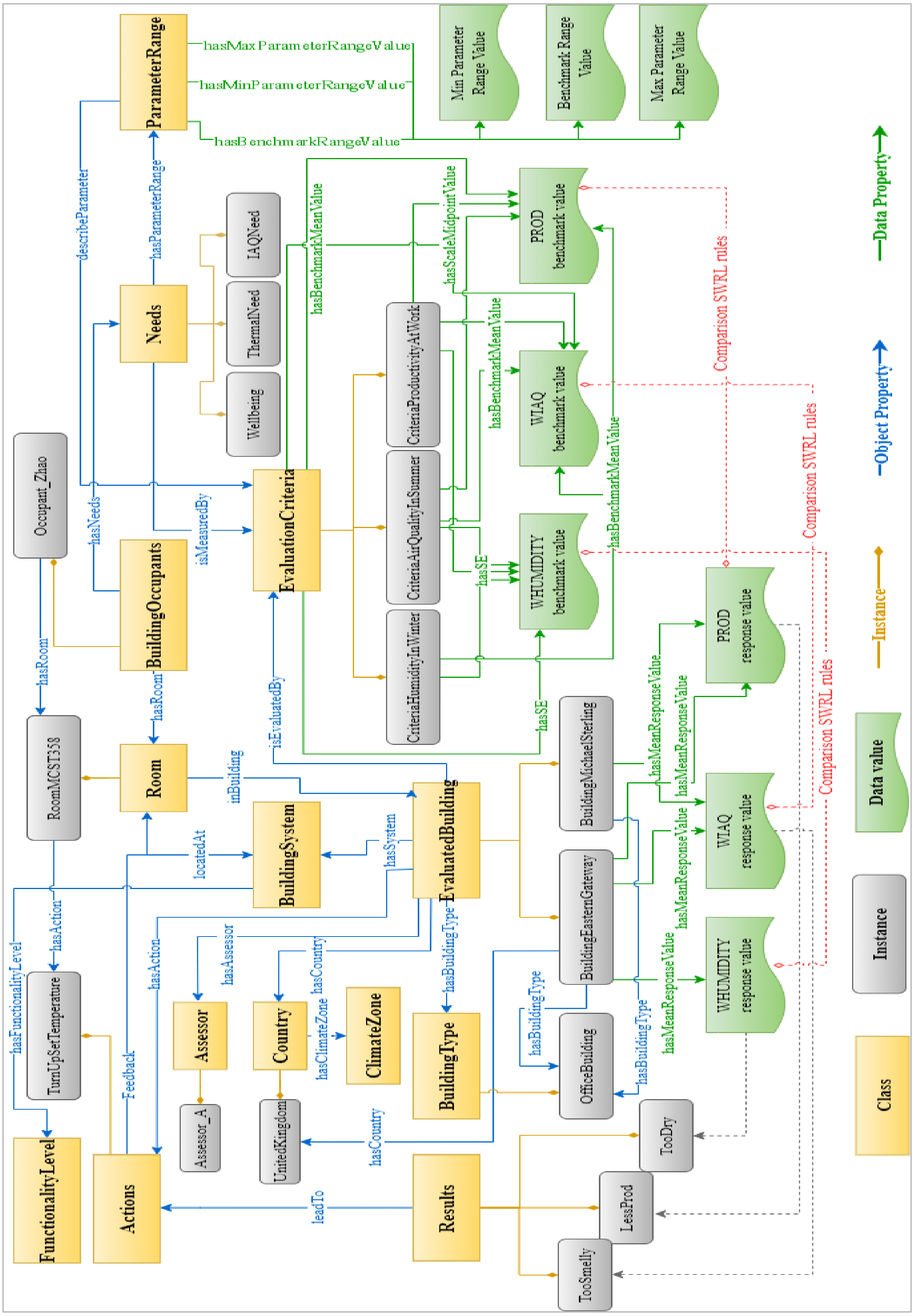


Figure 4-23 Overall relations between ontology entities

4.3 SWRL rules development

Based on the SWRL rules editing form as introduced in Section 2.6.4 of Chapter 2, SWRL rules are defined in an extensible Protégé plug-in, called SWRLTab, which provides a development environment for editing SWRL rules (O'Connor and Das, 2006). As shown in Figure 4-24, under the SWRLTab, there is a set of operation panels for rules editing and reasoning operation. The main rules edit panel consists few fields, the Name field is for naming the edited rules, the Comment field is used to annotate rules, the Status field is used to display red warnings about syntax errors when defining rules, and the rules editing zone is where the rules are edited.

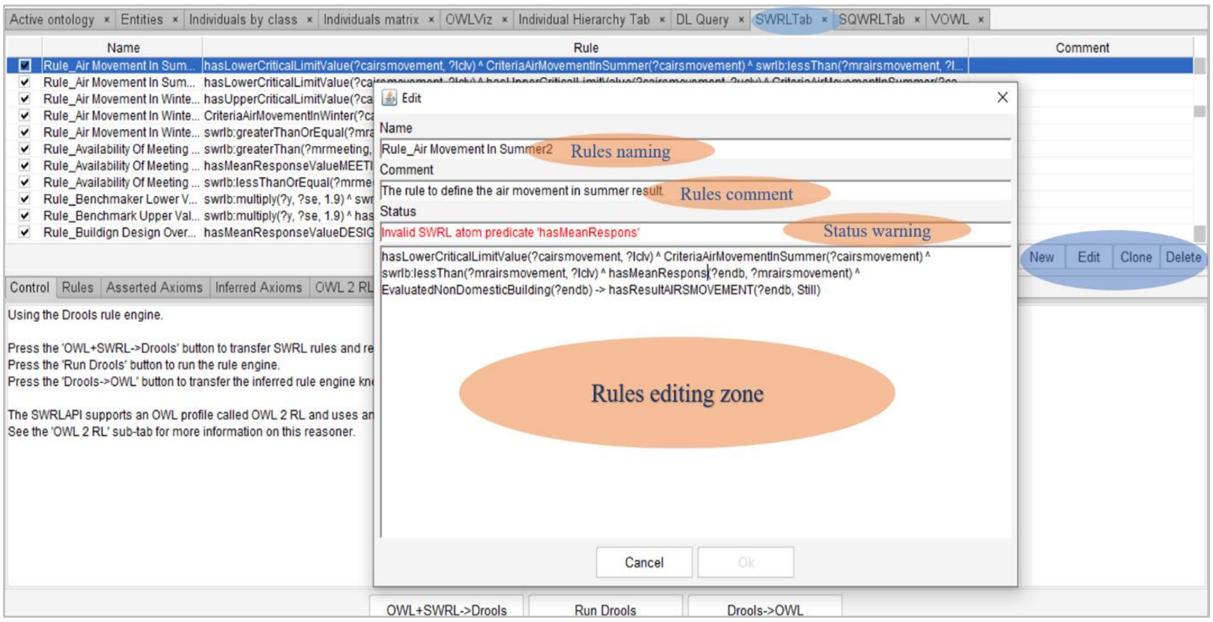


Figure 4-24 SWRLTab in Protégé

4.3.1 The thresholds of POE criteria SWRL rules defining

As discussed in Chapter 3, the POE assessment of building performances on satisfying occupant needs is measured by the evaluation criteria, mostly derived from the BUS Methodology, and the results of building performances assessment are presented at different levels. Therefore, the calculation of threshold values for the various assessment is developed upon two sets of standardized benchmark values from the BUS Methodology benchmarks dataset, namely, the benchmark mean value and the SE (Standard Error). According to the BUS Methodology guide, each evaluation criterion has the lower and upper limit values of scale midpoint mean value, and the lower and upper limit values of benchmark mean value respectively, that is, each evaluation criterion has two sets of lower limit mean values and upper limit mean values. In the BUS Methodology, the calculation of the upper and lower limit values is given a 95% confidence interval (CI), which is the most common used one. The Z value for the 95% confidence interval (CI) of each POE criterion is 1.96.

The calculation equation of the lower limit value and upper limit value is shown below:

$$\textit{Upper /Lower limit mean value} = \bar{x} \pm (1.96 \times SE) \tag{6}$$

where the $\bar{x}$ means the mean value, SE means the Standard Error, 1.96 is the Z value for the 95% confidence interval (CI).

To facilitate the editing of SWRL rules, the above equation can be simply expressed as:

$$\mathbf{X = Z \pm Y} \tag{7}$$

Based on Equations (6) and (7), the calculation rules can be expressed as following *if-then* form:

If the evaluation criteria have the Upper/Lower limit mean value *'X'*,
and *'Z'* is the mean value of scale midpoint mean value or the benchmark mean value, $\bar{x}$
and *'Y'* is the multiplication of 1.96 and Standard Error (SE)
and *'X'* is the summation/subtraction of *'Z'* and *'Y'*

Then the evaluation criteria have the Lower/Upper limit mean value *'X'*

Table 4-11 presents the SWRL rules for defining the benchmark lower value and benchmark upper value of evaluation criteria, and the used SWRL math built-ins include *swrlb:multiply, swrlb:subtract, and swrlb:add.*

Table 4-11 Benchmark mean lower and upper value rules

Rule 1: Benchmark Lower Value
EvaluationCriteria(?ec) ^ hasBenchamrkMeanValue(?ec, ?bmv) ^ swrlb:multiply(?y, ?se, 1.96) ^ hasSE(?ec, ?se)^swrlb:subtract(?x, ?bmv, ?y) -> hasBenchmarkLowerValue(?ec, ?x)
Rule 2: Benchmark Upper Value
EvaluationCriteria(?ec) ^ hasBenchamrkMeanValue(?ec, ?bmv) ^ swrlb:multiply(?y, ?se, 1.96) ^ hasSE(?ec, ?se) ^ swrlb:add(?x, ?bmv, ?y) -> hasBenchmarkUpperValue(?ec, ?x)

Figure 4-25, shows the visualised SWRL rules development process for defining the benchmark lower value and benchmark upper value of evaluation criteria.

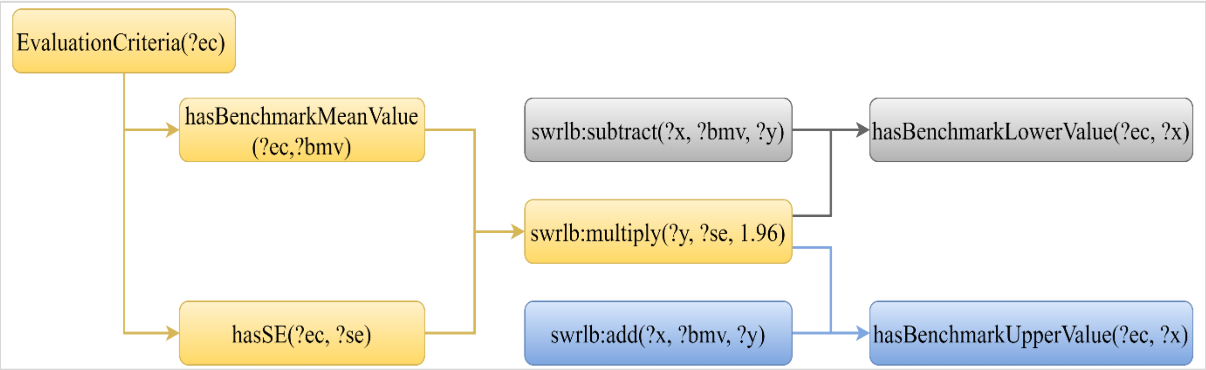


Figure 4-25 Benchmark mean lower and upper rules structure

Based on the same rules defining logic, the SWRL rules for defining the scale midpoint lower value and scale midpoint upper value of evaluation criteria are shown in Table 4-12, and the used SWRL math built-ins include *swrlb:multiply*, *swrlb:subtract, and swrlb:add.*

Table 4-12 Scale midpoint lower and upper value rules

Rule1: Scale Midpoint Lower Value
EvaluationCriteria(?ec) ^ hasScaleMidpointValue(?ec, ?smv) ^ swrlb:multiply(?y, ?se, 1.96) ^ hasSE(?ec, ?se) ^swrlb:subtract(?x, ?smv, ?y) -> hasScaleMidpointLowerValue(?ec, ?x)
Rule2: Scale Midpoint Upper Value
EvaluationCriteria(?ec) ^ hasScaleMidpointValue(?ec, ?smv) ^ swrlb:multiply(?y, ?se, 1.96) ^ hasSE(?ec, ?se) ^ swrlb:add(?x, ?smv, ?y) -> hasScaleMidpointUpperValue(?ec, ?x)

Figure 4-26, shows the visualised SWRL rules development process for defining the scale midpoint lower value and scale midpoint upper value of evaluation criteria.

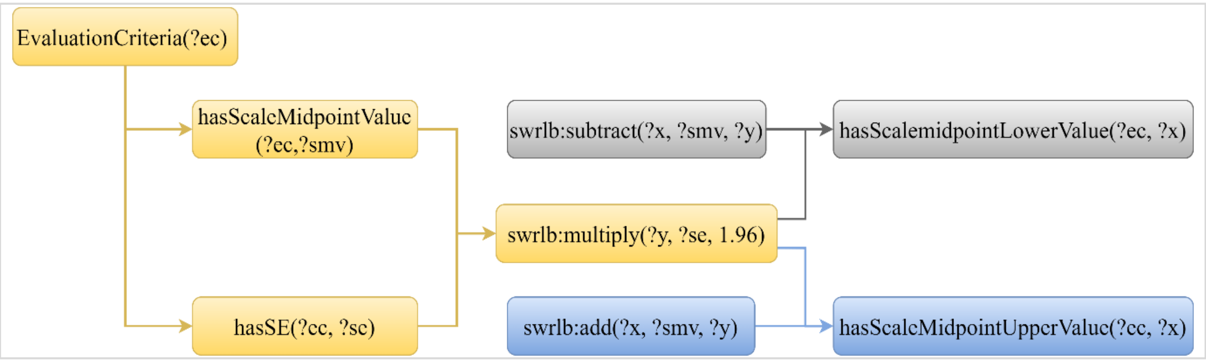


Figure 4-26 The lower and upper scale midpoint value rules structure

By adding the above-defined SWRL rules into the POE OWL ontology, the rules reasoning engine is used to generate new inferred knowledge, and the new knowledge is attached to the attributes of ontology individuals. Based on the new OWL facts knowledge that is generated via previously defined rules, the following SWRL rules are developed for facilitating further rules reasoning. According to the BUS Methodology guide, the lower and upper critical limit value of evaluation criteria is based on the value comparison between the lower and upper critical limit value of scale midpoint value and the benchmark mean value.

Table 4-13 presents the SWRL rules for defining the lower critical limit value of evaluation criteria, the key SWRL built-ins used in rules is *swrlb:lessThan*, which provides the ability to conduct the comparison between values. The rules can be expressed in an *if-then* form as,

If the evaluation criteria have a lower critical limit value,

and scale midpoint lower value,

and the benchmark lower value,

and the benchmark lower value is less than the scale midpoint upper value,

Then the evaluation criteria have a benchmark lower value as the lower critical limit value, and vice versa.

Table 4-13 The lower critical limit value defining rules

Rule1: Lower Critical Limit Value
EvaluationCriteria(?ec) ^ hasBenchmarkLowerValue(?ec, ?bmlv) ^ hasScaleMidpointLowerValue(?ec, ?smlv) ^ swrlb:lessThan(?bmlv, ?smlv) -> hasLowerCriticalLimitValue(?ec, ?bmlv)
Rule2: Lower Critical Limit Value
EvaluationCriteria(?ec) ^ hasBenchmarkLowerValue(?ec, ?bmlv) ^ hasScaleMidpointLowerValue(?ec, ?smlv) ^ swrlb:lessThan(?smlv, ?bmlv) -> hasLowerCriticalLimitValue(?ec, ?smlv)

Figure 4-27, shows the visualised value assigning SWRL rules development process for the lower critical limit value of evaluation criteria.

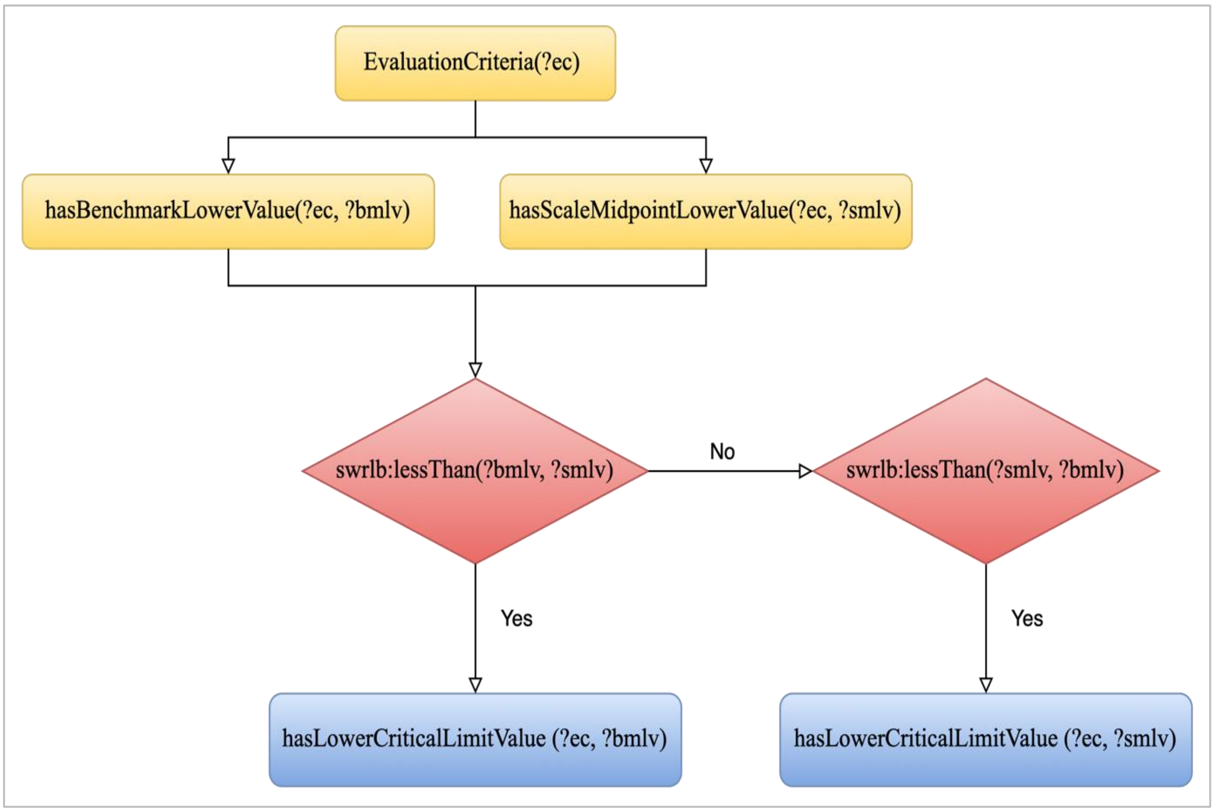


Figure 4-27 The lower critical limit value rules defining structure

Table 4-14 presents the SWRL rules for defining the upper critical limit value of evaluation criteria, the key SWRL built-ins is *swrlb:greaterThan*, which is used to compare the values. The rules can be expressed in an '*if-then*' form as, if the evaluation criteria have the benchmark upper value, scale midpoint upper value and the upper critical limit value, and the benchmark upper value is greater than the scale midpoint upper value, then the evaluation criteria have benchmark upper value as the upper critical limit value.

Table 4-14 The upper critical limit value defining rules

Rule1: Upper Critical Limit Value
EvaluationCriteria(?ec) ^ hasBenchmarkUpperValue(?ec, ?bmuv) ^ hasScaleMidpointUpperValue(?ec, ?smuv) ^ swrlb:greaterThan(?bmuv, ?smuv) -> hasUpperCriticalLimitValue(?ec, ?bmuv)
Rule2: Upper Critical Limit Value
EvaluationCriteria(?ec) ^ hasBenchmarkUpperValue(?ec, ?bmuv) ^ hasScaleMidpointUpperValue(?ec, ?smuv) ^ swrlb:greaterThan(?smuv, ?bmuv) -> hasUpperCriticalLimitValue(?ec, ?smuv)

Figure 4-28 shows the visualised value assigning SWRL rules development process for the scale midpoint lower value and scale midpoint upper value of evaluation criteria.

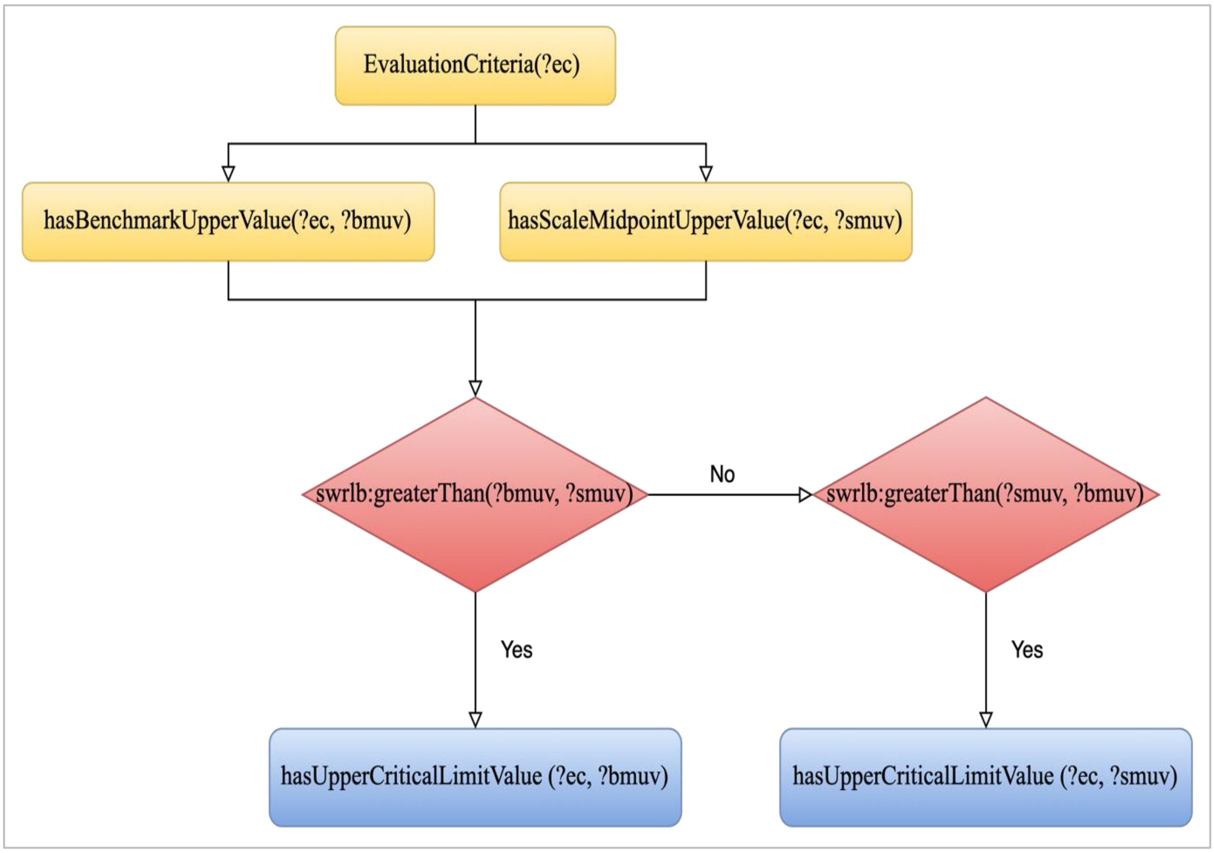


Figure 4-28 The upper critical limit value defining structure

The defined SWRL rules are stored in SWRLtab as part of the knowledge base. As the SWRL provides deductive reasoning capabilities by adding rules to extend OWL-DL, after defining the SWRL rules into the ontology knowledge base, the new facts knowledge are generated when executing the reasoner, then the new facts can be inferred from the existing OWL knowledge base, the new knowledge can be written back to the OWL ontology for further SWRL rules reasoning (O'Connor, Shankar, *et al.*, 2007). So far, the SWRL rules for assigning values to the key data properties of evaluation criteria have been established, the rules reasoning engine is used to infer implicit new facts or axioms from the given explicit facts in the knowledge base. After running the rule engine, the new facts are generated and attached to the related ontology concepts. In this study, the rule engines Pellet and Drools are applied, more details are given in Section 4.5.

4.3.2 POE assessment constraints SWRL rules defining

In Section 5.3.1, SWRL rules that define the thresholds of POE criteria have been established. These rules have laid a foundation for the editing and performing of POE assessment rules in this section. In this section, a set of building POE assessment rules has been developed in OWL ontology, here, take a few assessment indicators as examples to expound on the POE assessment rules definition and development process. As declared in previous sections, the POE assessment results have been clarified into three levels, and different evaluation results are defined corresponding to the different results levels of evaluation criteria.

As shown in Table 4-15, the assessment articles for temperature in summer have been coded into SWRL rules. It can be expressed as: 'For the evaluation criterion of temperature in summer, it has the threshold values of lower limit value and upper limit value, and the evaluated building has an assessed response mean value. If the building assessed response mean value is less than the lower limit value of the evaluation criterion, then the result of temperature in summer is too hot; if this assessed response mean value is greater than the upper limit value, then the result of temperature in summer is too cold, but if the assessed response mean value is greater than or equal to the lower limit value and less than or equal to the upper limit value, then the result of temperature in summer is satisfactory'. This evaluation rule also applies to other criteria.

Table 4-15 Temperature in summer assessment rules

Rule1: Temperature in summer assessment for the result of Too Hot
EvaluatedNonDomesticBuilding(?endb)^hasMeanResponseValueTSHOT(?endb,?mrtshot) ^ isEvaluatedBy(?endb, ?ctshot) ^ CriteriaTemperatureInSummer(?ctshot) ^ hasLowerCriticalLimitValue(?ctshot, ?lclv) ^ swrlb:lessThan(?mrtshot, ?lclv) -> hasResultTSHOT(?endb, TooHot)
Rule2: Temperature in summer assessment for the result of Satisfactory
EvaluatedNonDomesticBuilding(?endb)^hasMeanResponseValueTSHOT(?endb,?mrtshot) ^ CriteriaTemperatureInSummer(?ctshot) ^ hasLowerCriticalLimitValue(?ctshot, ?lclv) ^ hasUpperCriticalLimitValue(?ctshot, ?uclv) ^ isEvaluatedBy(?endb, ?ctshot) ^ swrlb:greaterThanOrEqual(?mrtshot, ?lclv) ^ swrlb:lessThanOrEqual(?mrtshot, ?uclv) -> hasResultTSHOT(?endb, Satisfactory)
Rule3: Temperature in summer assessment for the result of Too Cold

```
EvaluatedNonDomesticBuilding(?endb)^hasMeanResponseValueTSHOT(?endb,?mrtshot)
^ CriteriaTemperatureInSummer(?ctshot) ^  hasUpperCriticalLimitValue(?ctshot, ?uclv) ^
isEvaluatedBy(?endb, ?ctshot) ^ swrlb:greaterThan(?mrtshot, ?uclv)  ->
hasResultTSHOT(?endb, TooCold)
```

where, endb is the *evaluated non-domestic building*, *ctshot* is the *criterion of temperature in summer*, *mrtshot* is the *mean response value of temperature in summer*, *lclv* is the *lower critical limit value*, *uclv* is the *upper critical limit value*, *TSHOT* means the *temperature in summer*.

In this case, the rules can be expressed in an '*if-then*' form as,

If the evaluated building has assessed response mean value on evaluation criterion of temperature in summer,

and evaluation criterion has a lower limit value,

and evaluation criterion has an upper limit value,

and the assessed response mean value is less than the lower limit value,

Then the evaluated building has an evaluation result of too hot;

If the assessed response mean value is greater than the upper limit value,
Then the evaluated building has an evaluation result of too cold;

If the assessed response mean value is greater than or equal to the lower limit value
and less than or equal to the upper limit value,
Then the evaluated building has an evaluation result of satisfactory.

To understand the logical reasoning consequences of the asserted axioms, Figure 4-29 is presented to illustrate the assessment reasoning process.

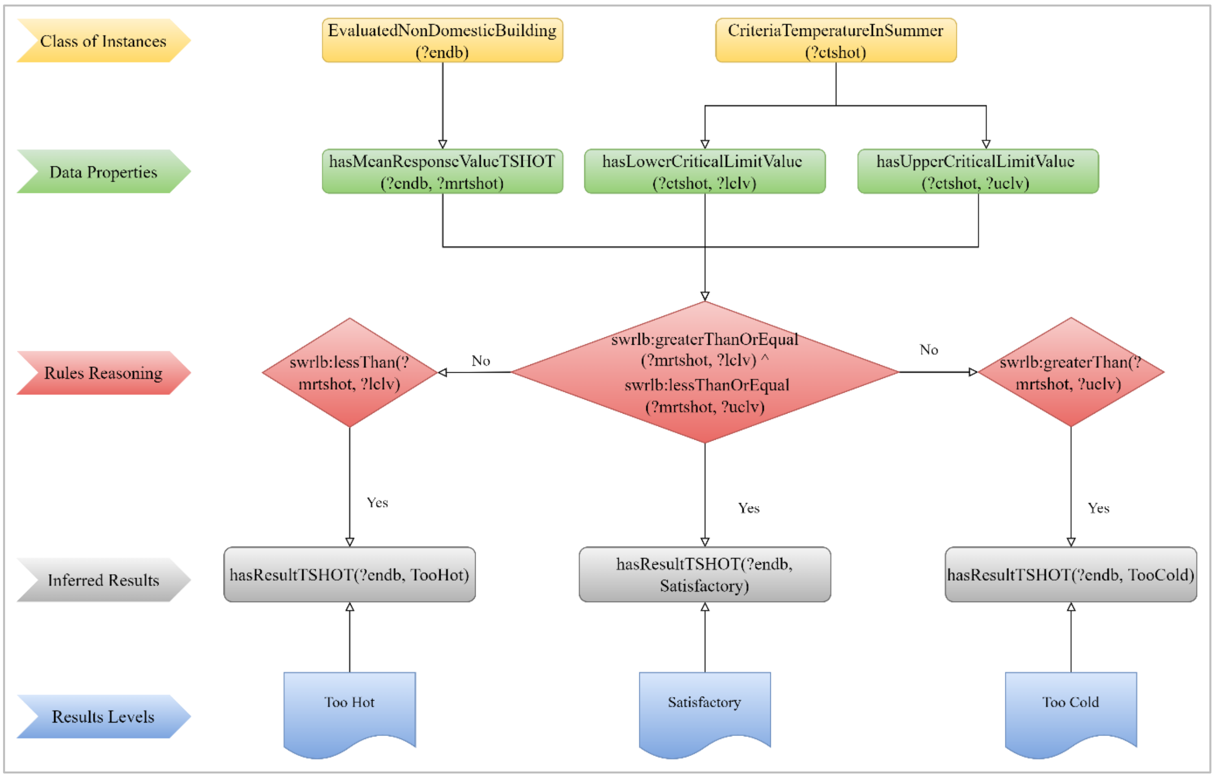


Figure 4-29 Temperature-in-summer assessment processes

Table 4-16 is an example list for several sets of SWRL rules, but not an exhaustive list of all the evaluation criteria assessment SWRL rules, since the other criteria assessment SWRL rules are edited by following the same editing syntax.

Table 4-16 Assessment SWRL rules examples

Rule_Air Movement In Summer
swrlb:greaterThan(?mrairsmovement,?uclv)^hasUpperCriticalLimitValue(?cairsmovemen, ?uclv)^CriteriaAirMovementInSummer(?cairsmovement)^hasMeanResponseValueAIRSMOVEMENT(?endb, ?mrairsmovement) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultAIRSMOVEMENT(?endb, Draughty)
hasLowerCriticalLimitValue(?cairsmovement, ?lclv) ^ CriteriaAirMovementInSummer (?cairsmovement) ^hasMeanResponseValueAIRSMOVEMENT(?endb, ?mrairsmovement) ^ swrlb:lessThan(?mrairsmovement, ?lclv) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultAIRSMOVEMENT(?endb, Still)
EvaluatedNonDomesticBuilding(?endb)^swrlb:lessThanOrEqual(?mrairsmovement, ?uclv) ^ hasLowerCriticalLimitValue(?cairsmovement, ?lclv) ^ hasUpperCriticalLimitValue (?cairsmovement, ?uclv) ^ CriteriaAirMovementInSummer(?cairsmovement) ^

hasMeanResponseValueAIRSMOVEMENT(?endb, ?mrairsmovement) ^ swrlb:greaterThanOrEqual(?mrairsmovement, ?lclv) -> hasResultAIRSMOVEMENT(?endb, Satisfactory)
Rule_Indoor Air Quality Winter
hasUpperCriticalLimitValue(?cwiaq, ?uclv) ^ CriteriaAirQualityInWinter(?cwiaq) ^ hasMeanResponseValueWIAQ(?endb, ?mwiaq) ^ swrlb:greaterThan(?mwiaq, ?uclv) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultWIAQ(?endb, TooSmelly)
EvaluatedNonDomesticBuilding(?endb) ^hasMeanResponseValueWIAQ(?endb,?mwiaq) ^ CriteriaAirQualityInWinter(?cwiaq) ^ hasUpperCriticalLimitValue(?cwiaq, ?uclv) ^ swrlb:lessThanOrEqual(?mwiaq, ?uclv) ^ swrlb:greaterThanOrEqual(?mwiaq, ?lclv) ^ hasLowerCriticalLimitValue(?cwiaq, ?lclv) -> hasResultWIAQ(?endb, Satisfactory)
EvaluatedNonDomesticBuilding(?endb) ^ hasLowerCriticalLimitValue(?csiaq, ?lclv) ^ CriteriaAirQualityInWinter(?cwiaq) ^ hasMeanResponseValueWIAQ(?endb, ?mwiaq) ^ swrlb:lessThan(?mwiaq, ?lclv) -> hasResultWIAQ(?endb, Outstanding)
Rule_Productivity At Work
EvaluatedNonDomesticBuilding(?endb))^hasMeanResponseValuePROD(?endb, ?mrprod) ^hasUpperCriticalLimitValue(?cprod, ?uclv) ^ swrlb:greaterThan(?mrprod, ?uclv) ^ CriteriaProductivityAtWork(?cprod) -> hasResultPROD(?endb, MoreProductivity)
EvaluatedNonDomesticBuilding(?endb) ^ hasMeanResponseValuePROD(?endb, ?mrprod) ^ CriteriaProductivityAtWork(?cprod) ^ hasLowerCriticalLimitValue(?cprod, ?lclv) ^ swrlb:lessThan(?mrprod, ?lclv) -> hasResultPROD(?endb, LessProductivity)
hasUpperCriticalLimitValue(?cprod,?uclv)^ hasLowerCriticalLimitValue(?cprod, ?lclv) ^ hasMeanResponseValuePROD(?endb, ?mrprod) ^ swrlb:lessThanOrEqual(?mrprod, ?uclv) ^ swrlb:greaterThanOrEqual(?mrprod, ?lclv) ^ CriteriaProdAtWork(?cprod) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultPROD(?endb, NoDifference)
Rule_Thermal Comfort In Winter
swrlb:lessThan(?mrtwover,?lclv)^hasMeanResponseValueTWOVER(?endb,?mrtwover) ^ hasLowerCriticalLimitValue(?ctwover,?lclv)^CriteriaThermalComfortInWinter(?ctwover) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultTWOVER(?endb, Uncomfortable)
CriteriaThermalComfortInWinter(?ctwover) ^ EvaluatedNonDomesticBuilding(?endb) ^ swrlb:greaterThan(?mrtwover, ?uclv) ^ hasUpperCriticalLimitValue(?ctwover, ?uclv) ^

hasMeanResponseValueTWOVER(?endb, ?mrtwover) -> hasResultTWOVER(?endb, Outstanding)
CriteriaThermalComfortInWinter(?ctwover) ^ EvaluatedNonDomesticBuilding(?endb) ^ swrlb:lessThanOrEqual(?mrtwover,?uclv)^hasUpperCriticalLimitValue(?ctwover, ?uclv)^ hasLowerCriticalLimitValue(?ctwover,?lclv)^swrlb:greaterThanOrEqual(?mrtwover,?lclv) ^ hasMeanResponseValueTWOVER(?endb, ?mrtwover) -> hasResultTWOVER(?endb, Comfortable)

The concept of relative improvement ratio (RIR) is introduced in this study to show the impact degree of certain building performance on building occupants' satisfaction and indicate how big the performance result gap is between the evaluated buildings and the benchmark buildings. After conducting the POE assessment rules, the results of evaluated buildings are given. Based on these results values, if the results values of evaluated buildings are better than the benchmark values, then this relative improvement ratio will show how much better the evaluated buildings performed than the benchmark buildings; if the assessment results are unsatisfactory, then the relative improvement ratio represents how much potential improvement capability that the evaluated buildings need and also can be used to show the degree of influence on unsatisfactory results with building performance. Based on this relative improvement ratio, building optimization can be prioritized in the ontology model.

The relative improvement ratio is defined by the response mean value of evaluated buildings and the benchmark mean value of evaluation criteria. It is calculated as the absolute value of the subtraction of benchmark mean value and response mean value, then the absolute value divided by response mean value.

$$\textit{Relative Improvement Ratio (rir)} = |Bm - Rm| \,/\, Rm \qquad (8)$$

Where *Bm* means benchmark mean value, *Rm* means the response mean value

Some rules examples are shown in Table 4-17. The used built-ins are swrlb:abs, swrlb:subtract and swrlb:divide.

Table 4-17 Relative improvement ratio rules

Rule_RIR IAQ In Summer
EvaluatedNonDomesticBuilding(?endb) ^ hasBenchamrkMeanValue(?csiaq, ?bmsiaq) ^ CriteriaAirQualityInSummer(?csiaq) ^ hasMeanResponseValueSIAQ(?endb,?mrsiaq) ^ swrlb:subtract(?rgsiaq,?bmsiaq,?mrsiaq) ^ swrlb:divide(?rirsiaq, ?argsiaq, ?mrsiaq) swrlb:abs(?argsiaq, ?rgsiaq) ^ -> hasRIRSIAQ(?endb, ?rirsiaq)
Rule_RIR In Perceived Health
EvaluatedNonDomesticBuilding(?endb) ^ CriteriaPerceivedHealth(?chealth) ^ hasMeanResponseValueHEALTH(?endb, ?mrhealth) ^ swrlb:abs(?arghealth, ?rghealth) ^ hasBenchamrkMeanValue(?chealth, ?bmhealth) ^ swrlb:subtract (?rghealth, ?bmhealth, ?mrhealth) ^ swrlb:divide(?rirhealth, ?arghealth, ?mrhealth) -> hasRIRHEALTH(?endb,?rirhealth)
Rule_RIR Temperature In Winter
EvaluatedNonDomesticBuilding(?endb) ^ CriteriaTemperatureInWinter(?ctwhot) ^ hasMeanResponseValueTWHOT(?endb, ?mrtwhot) ^ swrlb:abs(?argtwhot, ?rgtwhot) ^ hasBenchamrkMeanValue(?ctwhot,?bmtwhot)^swrlb:divide(?rirtwhot,?argtwhot,?mrtwhot) ^ swrlb:subtract(?rgtwhot, ?bmtwhot, ?mrtwhot) -> hasRIRTWHOT(?endb,?rirtwhot)
Rule_RIR In Personal Control Heating
EvaluatedNonDomesticBuilding(?endb)^hasMeanResponseValueCNTHT(?endb,?mrcntht) ^ CriteriaPersonalControlOverHeating(?ccntht) ^swrlb:divide(?rircntht,?argcntht,?mrcntht) ^ hasBenchamrkMeanValue(?ccntht, ?bmcntht) ^ swrlb:abs(?argcntht, ?rgcntht) ^ swrlb:subtract(?rgcntht, ?bmcntht, ?mrcntht) -> hasRIRCNTHT(?endb, ?rircntht)

After defining POE rules, the SWRL rules of the actions corresponding to evaluation results are developed. In response to different evaluation results, the building may need to take different measures to adjust the operation mode of buildings to meet building occupants' satisfaction. Table 4-18 presents some action rules, for example, for the humidity in winter, if the result is too dry, then two suggestions are provided to the building facility managers or the building occupants: add humidifiers or introduce plants to offices. If the temperature in the winter survey result is too hot, that probably means the heating systems setting value is too high, so the corresponding suggested action is to turn down the heating setting temperature value. If the survey assessment result of indoor air quality in summer is too smelly, the suggested actions are to increase the frequency of window opening time or give building occupants more control on window controlling.

Table 4-18 Actions SWRL rules

Rule_Action Humidity In Winter
EvaluatedNonDomesticBuilding(?endb) ^ hasResultWHUMIDITY(?endb, TooDry) -> hasActionWinterHumidity(?endb, AddHumidifier) ^ hasActionWinterHumidity(?endb, IntroducePlantsToOffice)
Rule_Action IAQ TooSmelly In Summer
EvaluatedNonDomesticBuilding(?endb) ^ hasResultSIAQ(?endb, TooSmelly) -> hasActionSummerIAQ(?endb, InceraseFrequencyOfWindowOpening)
Rule_Action Temperature TooCold In Summer
hasResultTSHOT(?endb, TooCold) ^ EvaluatedNonDomesticBuilding(?endb) -> hasActionSummerTemperature(?endb, TurnUpACSettingTemperature)
Rule_Action TooHot In Winter
hasResultTWHOT(?endb, TooHot) ^ EvaluatedNonDomesticBuilding(?endb) -> hasActionWinterTemperature(?endb, TurnDownHeatingSettingTemperature)

So far, the SWRL rules of POE ontology have been established, these rules are developed based on the OWL knowledge base, which provides ontology with the inference function that allows the new inferred facts to be generated after the execution of SWRL rules, and then new facts can be returned to the knowledge base as the new facts of the OWL ontology. More SWRL rules can be found in the attached appendix.

Based on the defined SWRL rules, the SQWRL query rules are developed to retrieve information or knowledge needed by users from the OWL ontology. The development of query rules is presented in the following section.

4.4 SQWRL query rules development

The development of SQWRL query rules follows the pattern specification shown in Equation ***(5)*** of Chapter 2. As introduced in Chapter 2, the queries are operated in combination with SWRL rules to retrieve the knowledge which is inferred by SWRL rules. For instance, in this case, a set of SWRL rules are coded in the POE ontology knowledge base to define the lower and upper critical limit values of evaluation criteria. Based on these rules, the SQWRL query can be written to retrieve the corresponding value for evaluation criteria in POE ontology. Take an example of the lower and upper limit value of the evaluation criteria SQWRL query from the beginning of SWRL rules editing to the end of reasoning.

At first, the SWRL rules of the benchmark mean value and scale midpoint values associated with the lower and upper limit value of evaluation criteria are given in the OWL knowledge base, as shown in Table 4-19.

Table 4-19 Lower & Upper Limit Value of Evaluation Criteria

Rule_Benchmark Lower Value
EvaluationCriteria(?ec)^hasBenchamrkMeanValue(?ec,?bmv)^ swrlb:subtract(?x,?bmv,?y) ^ swrlb:multiply(?y, ?se, 1.96) ^ hasSE(?ec, ?se) -> hasBenchmarkLowerValue(?ec, ?x)
Rule_Benchmark Upper Value
EvaluationCriteria(?ec)^hasBenchamrkMeanValue(?ec, ?bmv) ^ swrlb:add(?x, ?bmv, ?y) ^ swrlb:multiply(?y, ?se, 1.96) ^ hasSE(?ec, ?se) -> hasBenchmarkUpperValue(?ec, ?x)
Rule_Sacle Midpoint Lower Value
EvaluationCriteria(?ec)^hasScaleMidpointValue(?ec,?smv) ^ swrlb:multiply(?y, ?se, 1.96) ^ swrlb:subtract(?x, ?smv, ?y) ^ hasSE(?ec, ?se) -> hasScaleMidpointLowerValue(?ec, ?x)
Rule_Scale Midpoint Upper Value
EvaluationCriteria(?ec) ^hasScaleMidpointValue(?ec, ?smv) ^swrlb:multiply(?y, ?se, 1.96) ^ swrlb:add(?x, ?smv, ?y) ^ hasSE(?ec, ?se) -> hasScaleMidpointUpperValue(?ec, ?x)

Second, the rules for numerical comparisons defined above are shown in Table 4-20.

Table 4-20 Rules for numerical comparisons

Rule_Lower Critical Limit Value 1
EvaluationCriteria(?ec)^hasScaleMidpointLowerValue(?ec,?smlv)^swrlb:lessThan(?bmlv, ?smlv)^hasBenchmarkLowerValue(?ec,?bmlv)^->hasLowerCriticalLimitValue(?ec,?bmlv)
Rule_Lower Critical Limit Value 2
EvaluationCriteria(?ec)^hasScaleMidpointLowerValue(?ec,?smlv)^swrlb:lessThan(?smlv, ?bmlv)^hasBenchmarkLowerValue(?ec,?bmlv) -> hasLowerCriticalLimitValue(?ec, ?smlv)
Rule_Upper Critical Limit Value1

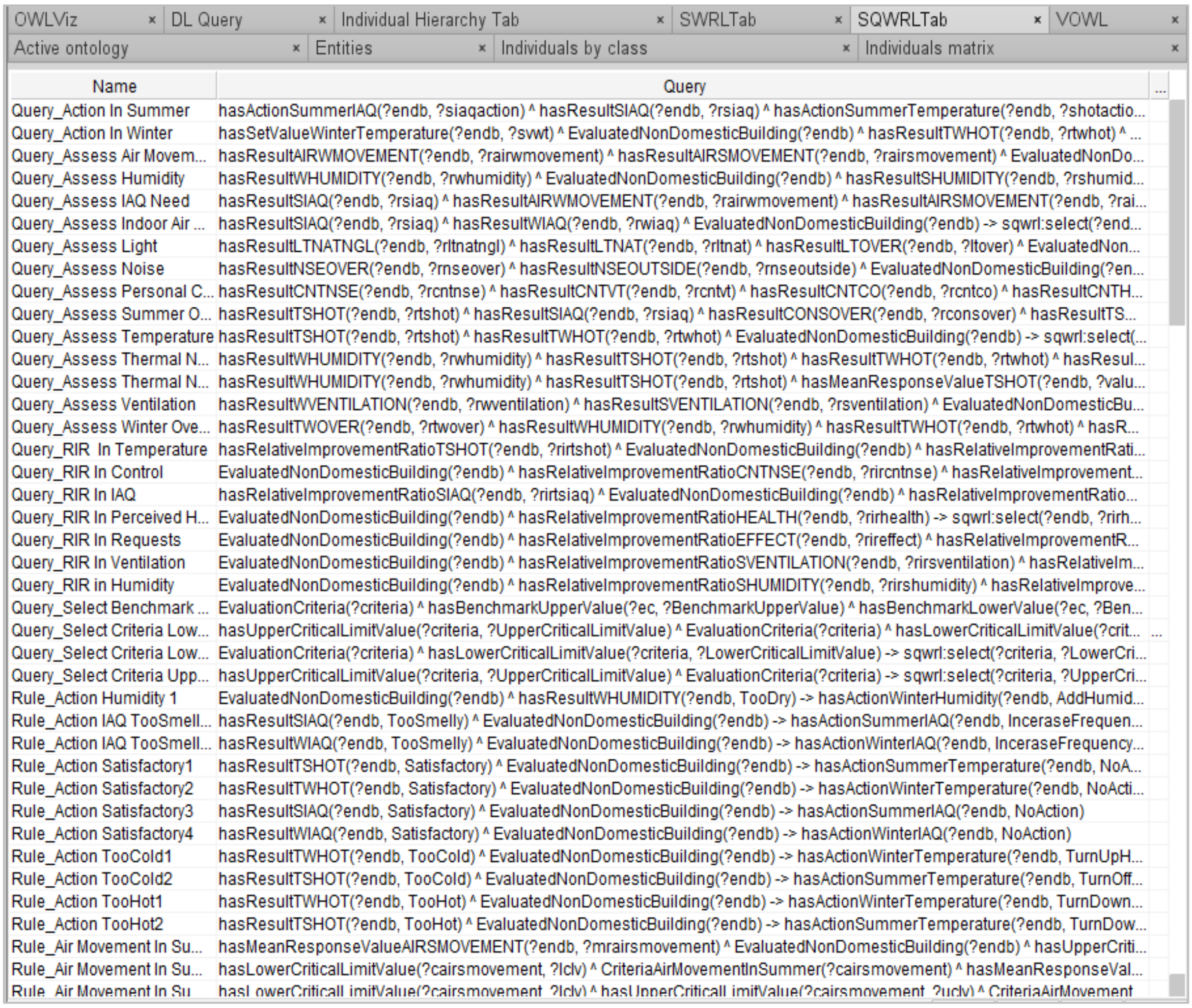
EvaluationCriteria(?ec) ^ hasScaleMidpointUpperValue(?ec, ?smuv) ^ hasBenchmarkUpperValue(?ec, ?bmuv) ^ swrlb:greaterThan(?bmuv, ?smuv) -> hasUpperCriticalLimitValue(?ec, ?bmuv)
Rule_Upper Critical limit Value2
EvaluationCriteria(?ec) ^ hasScaleMidpointUpperValue(?ec, ?smuv) ^ hasBenchmarkUpperValue(?ec, ?bmuv) ^ swrlb:greaterThan(?smuv, ?bmuv) -> hasUpperCriticalLimitValue(?ec, ?smuv)

And then, the SQWRL rule for querying the critical lower and upper limit value of evaluation criteria is defined in Table 4-21.

Table 4-21 Query Select Criterial Lower & Upper Limit Value of Evaluation Criteria

Query_Select Criterial Lower & Upper Limit Value
EvaluationCriteria(?criteria) ^ hasUpperCriticalLimitValue(?criteria, ?UpperCriticalLimitValue) ^ hasLowerCriticalLimitValue(?criteria, ?LowerCriticalLimitValue) -> sqwrl:select(?criteria, ?LowerCriticalLimitValue, ?UpperCriticalLimitValue)

By following the same defining manner as explained above, a group of SQWRL rules are developed into the POE ontology. Figure 4-30 shows part of the developed SWRL rules and SQWRL rules for the proposed POE ontology. The implementation of these rules is illustrated in the case study chapter, and the more detailed rules are provided in the appendix at the end of this thesis.



Name	Query
Query_Action In Summer	hasActionSummerIAQ(?endb, ?siaqaction) ^ hasResultSIAQ(?endb, ?rsiaq) ^ hasActionSummerTemperature(?endb, ?shotactio...
Query_Action In Winter	hasSetValueWinterTemperature(?endb, ?svwt) ^ EvaluatedNonDomesticBuilding(?endb) ^ hasResultTWHOT(?endb, ?rtwhot) ^ ...
Query_Assess Air Movem...	hasResultAIRWMOVEMENT(?endb, ?rairwmovement) ^ hasResultAIRSMOVEMENT(?endb, ?rairsmovement) ^ EvaluatedNonDo...
Query_Assess Humidity	hasResultWHUMIDITY(?endb, ?rwhumidity) ^ EvaluatedNonDomesticBuilding(?endb) ^ hasResultSHUMIDITY(?endb, ?rshumid...
Query_Assess IAQ Need	hasResultSIAQ(?endb, ?rsiaq) ^ hasResultAIRWMOVEMENT(?endb, ?rairwmovement) ^ hasResultAIRSMOVEMENT(?endb, ?rai...
Query_Assess Indoor Air ...	hasResultSIAQ(?endb, ?rsiaq) ^ hasResultWIAQ(?endb, ?rwiaq) ^ EvaluatedNonDomesticBuilding(?endb) -> sqwrl:select(?end...
Query_Assess Light	hasResultLTNATNGL(?endb, ?rltnatngl) ^ hasResultLTNAT(?endb, ?rltnat) ^ hasResultLTOVER(?endb, ?ltover) ^ EvaluatedNon...
Query_Assess Noise	hasResultNSEOVER(?endb, ?rnseover) ^ hasResultNSEOUTSIDE(?endb, ?rnseoutside) ^ EvaluatedNonDomesticBuilding(?en...
Query_Assess Personal C...	hasResultCNTNSE(?endb, ?rcntnse) ^ hasResultCNTVT(?endb, ?rcntvt) ^ hasResultCNTCO(?endb, ?rcntco) ^ hasResultCNTH...
Query_Assess Summer O...	hasResultTSHOT(?endb, ?rtshot) ^ hasResultSIAQ(?endb, ?rsiaq) ^ hasResultCONSOVER(?endb, ?rconsover) ^ hasResultTS...
Query_Assess Temperature	hasResultTSHOT(?endb, ?rtshot) ^ hasResultTWHOT(?endb, ?rtwhot) ^ EvaluatedNonDomesticBuilding(?endb) -> sqwrl:select(...
Query_Assess Thermal N...	hasResultWHUMIDITY(?endb, ?rwhumidity) ^ hasResultTSHOT(?endb, ?rtshot) ^ hasResultTWHOT(?endb, ?rtwhot) ^ hasResul...
Query_Assess Thermal N...	hasResultWHUMIDITY(?endb, ?rwhumidity) ^ hasResultTSHOT(?endb, ?rtshot) ^ hasMeanResponseValueTSHOT(?endb, ?valu...
Query_Assess Ventilation	hasResultWVENTILATION(?endb, ?rwventilation) ^ hasResultSVENTILATION(?endb, ?rsventilation) ^ EvaluatedNonDomesticBu...
Query_Assess Winter Ove...	hasResultTWOVER(?endb, ?rtwover) ^ hasResultWHUMIDITY(?endb, ?rwhumidity) ^ hasResultTWHOT(?endb, ?rtwhot) ^ hasR...
Query_RIR In Temperature	hasRelativeImprovementRatioTSHOT(?endb, ?rirtshot) ^ EvaluatedNonDomesticBuilding(?endb) ^ hasRelativeImprovementRati...
Query_RIR In Control	EvaluatedNonDomesticBuilding(?endb) ^ hasRelativeImprovementRatioCNTNSE(?endb, ?rircntnse) ^ hasRelativeImprovement...
Query_RIR In IAQ	hasRelativeImprovementRatioSIAQ(?endb, ?rirtsiaq) ^ EvaluatedNonDomesticBuilding(?endb) ^ hasRelativeImprovementRatio...
Query_RIR In Perceived H...	EvaluatedNonDomesticBuilding(?endb) ^ hasRelativeImprovementRatioHEALTH(?endb, ?rirhealth) -> sqwrl:select(?endb, ?rirh...
Query_RIR In Requests	EvaluatedNonDomesticBuilding(?endb) ^ hasRelativeImprovementRatioEFFECT(?endb, ?rireffect) ^ hasRelativeImprovementR...
Query_RIR In Ventilation	EvaluatedNonDomesticBuilding(?endb) ^ hasRelativeImprovementRatioSVENTILATION(?endb, ?rirsventilation) ^ hasRelativeIm...
Query_RIR in Humidity	EvaluatedNonDomesticBuilding(?endb) ^ hasRelativeImprovementRatioSHUMIDITY(?endb, ?rirshumidity) ^ hasRelativeImprove...
Query_Select Benchmark ...	EvaluationCriteria(?criteria) ^ hasBenchmarkUpperValue(?ec, ?BenchmarkUpperValue) ^ hasBenchmarkLowerValue(?ec, ?Ben...
Query_Select Criteria Low...	hasUpperCriticalLimitValue(?criteria, ?UpperCriticalLimitValue) ^ EvaluationCriteria(?criteria) ^ hasLowerCriticalLimitValue(?crit...
Query_Select Criteria Low...	EvaluationCriteria(?criteria) ^ hasLowerCriticalLimitValue(?criteria, ?LowerCriticalLimitValue) -> sqwrl:select(?criteria, ?LowerCri...
Query_Select Criteria Upp...	hasUpperCriticalLimitValue(?criteria, ?UpperCriticalLimitValue) ^ EvaluationCriteria(?criteria) -> sqwrl:select(?criteria, ?UpperCri...
Rule_Action Humidity 1	EvaluatedNonDomesticBuilding(?endb) ^ hasResultWHUMIDITY(?endb, TooDry) -> hasActionWinterHumidity(?endb, AddHumid...
Rule_Action IAQ TooSmell...	hasResultSIAQ(?endb, TooSmelly) ^ EvaluatedNonDomesticBuilding(?endb) -> hasActionSummerIAQ(?endb, InceraseFrequen...
Rule_Action IAQ TooSmell...	hasResultWIAQ(?endb, TooSmelly) ^ EvaluatedNonDomesticBuilding(?endb) -> hasActionWinterIAQ(?endb, InceraseFrequency...
Rule_Action Satisfactory1	hasResultTSHOT(?endb, Satisfactory) ^ EvaluatedNonDomesticBuilding(?endb) -> hasActionSummerTemperature(?endb, NoA...
Rule_Action Satisfactory2	hasResultTWHOT(?endb, Satisfactory) ^ EvaluatedNonDomesticBuilding(?endb) -> hasActionWinterTemperature(?endb, NoActi...
Rule_Action Satisfactory3	hasResultSIAQ(?endb, Satisfactory) ^ EvaluatedNonDomesticBuilding(?endb) -> hasActionSummerIAQ(?endb, NoAction)
Rule_Action Satisfactory4	hasResultWIAQ(?endb, Satisfactory) ^ EvaluatedNonDomesticBuilding(?endb) -> hasActionWinterIAQ(?endb, NoAction)
Rule_Action TooCold1	hasResultTWHOT(?endb, TooCold) ^ EvaluatedNonDomesticBuilding(?endb) -> hasActionWinterTemperature(?endb, TurnUpH...
Rule_Action TooCold2	hasResultTSHOT(?endb, TooCold) ^ EvaluatedNonDomesticBuilding(?endb) -> hasActionSummerTemperature(?endb, TurnOff...
Rule_Action TooHot1	hasResultTWHOT(?endb, TooHot) ^ EvaluatedNonDomesticBuilding(?endb) -> hasActionWinterTemperature(?endb, TurnDown...
Rule_Action TooHot2	hasResultTSHOT(?endb, TooHot) ^ EvaluatedNonDomesticBuilding(?endb) -> hasActionSummerTemperature(?endb, TurnDow...
Rule_Air Movement In Su...	hasMeanResponseValueAIRSMOVEMENT(?endb, ?mrairsmovement) ^ EvaluatedNonDomesticBuilding(?endb) ^ hasUpperCriti...
Rule_Air Movement In Su...	hasLowerCriticalLimitValue(?cairsmovement, ?lclv) ^ CriteriaAirMovementInSummer(?cairsmovement) ^ hasMeanResponseVal...
Rule_Air Movement In Su	hasLowerCriticalLimitValue(?cairsmovement, ?lclv) ^ hasUpperCriticalLimitValue(?cairsmovement, ?uclv) ^ CriteriaAirMovement

Figure 4-30 SWRL rules and SQWRL queries in POE ontology

4.5 Reasoning and query implementation

So far, the SWRL rules for inferring new facts and the SQWRL rules for querying needed information in the POE ontology have been established. The rule engine is used to infer implicit new facts or axioms from the given explicit facts in the knowledge base. After running the rule engine, the new facts are generated and attached to the related ontology concepts.

4.5.1 Ontology validation and reasoning

After building the knowledge base of POE ontology, to ensure the consistency and semantic accuracy of developed ontology, it is a good practice to verify the ontology and check if there are any errors. The ontology validation is processed by using the reasoners, for instance, the Pellet and Drools. Pellet is an open-source Java-based OWL DL reasoner and is mainly used for ontology validation and can be used at any time in the process of constructing the ontology.

It provides the functionalities of consistency checking, classification hierarchy computing, information inferring, and so on.

Pellet is a plug-in incorporated in Protégé 5.5.0, under the Reasoner tab, select Pellet, then click Start reasoner. After running the reasoner, the reasoning log is generated under the Configure, if there is any error, it will be highlighted in red. As shown in Figure 4-31, this POE ontology does not contain any contradictory facts, it also gives the reasoning time which is 112 ms in this case, the reasoning process is very quick.

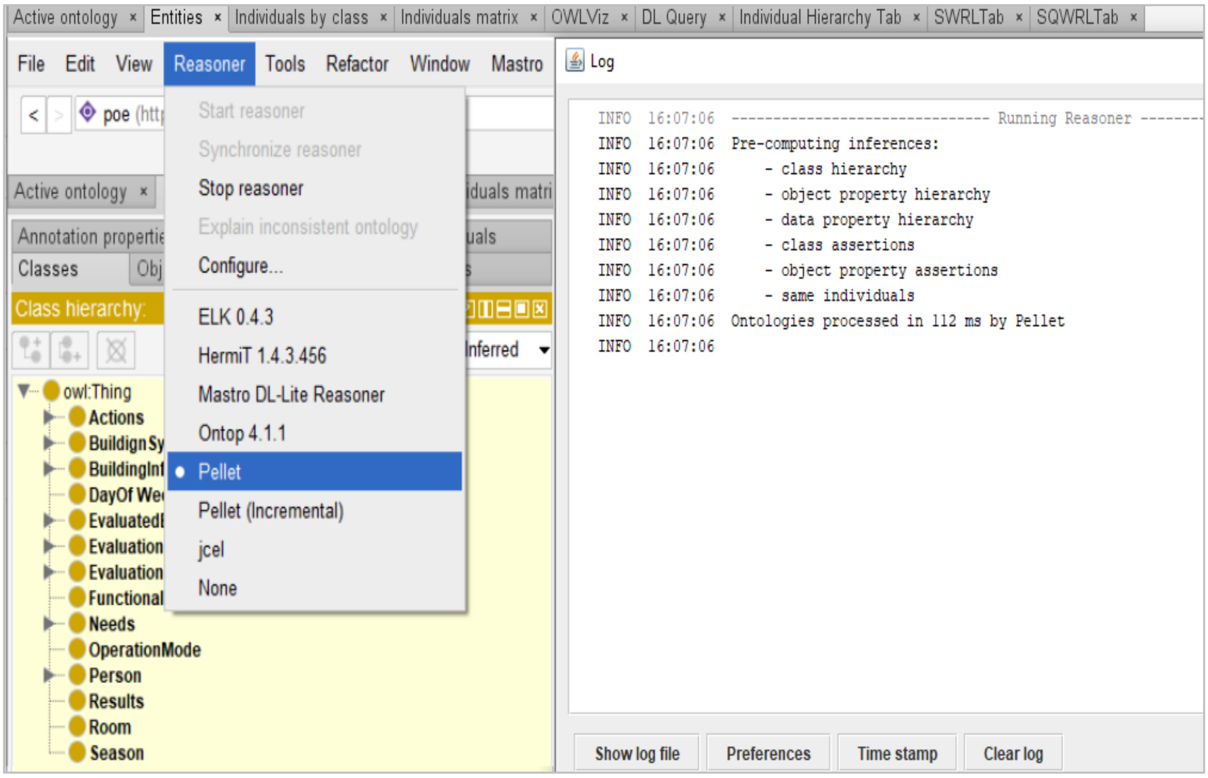


Figure 4-31 Ontology validation

In addition to ontology consistency checking, the Pellet also can be used to infer implicit new facts or axioms of ontology from the given explicit facts in the knowledge base, and any inferred axioms caused by the SWRL rules are highlighted in yellow, but not be transferred back to the current knowledge base. An implementation example of this functionality is given in the case study chapter.

The Drools engine is a plugin to the SWRLAPI (O'Connor and Das, 2012) that is used to assert new knowledge into the ontology as axioms. As shown in Figure 4-32, the Drools engine has three buttons to conduct the reasoning and manage new inferred information transfer and inference processes. The new axioms inferred by Drools engine are written back to the OWL ontology knowledge base after pressing the *'Drools->OWL'* button. Different from the Pellet reasoner, the Drools engine translates the inferred axioms back to the OWL knowledge, which to the ontology editor looks the same as other user-defined axioms (DeBellis, 2020).

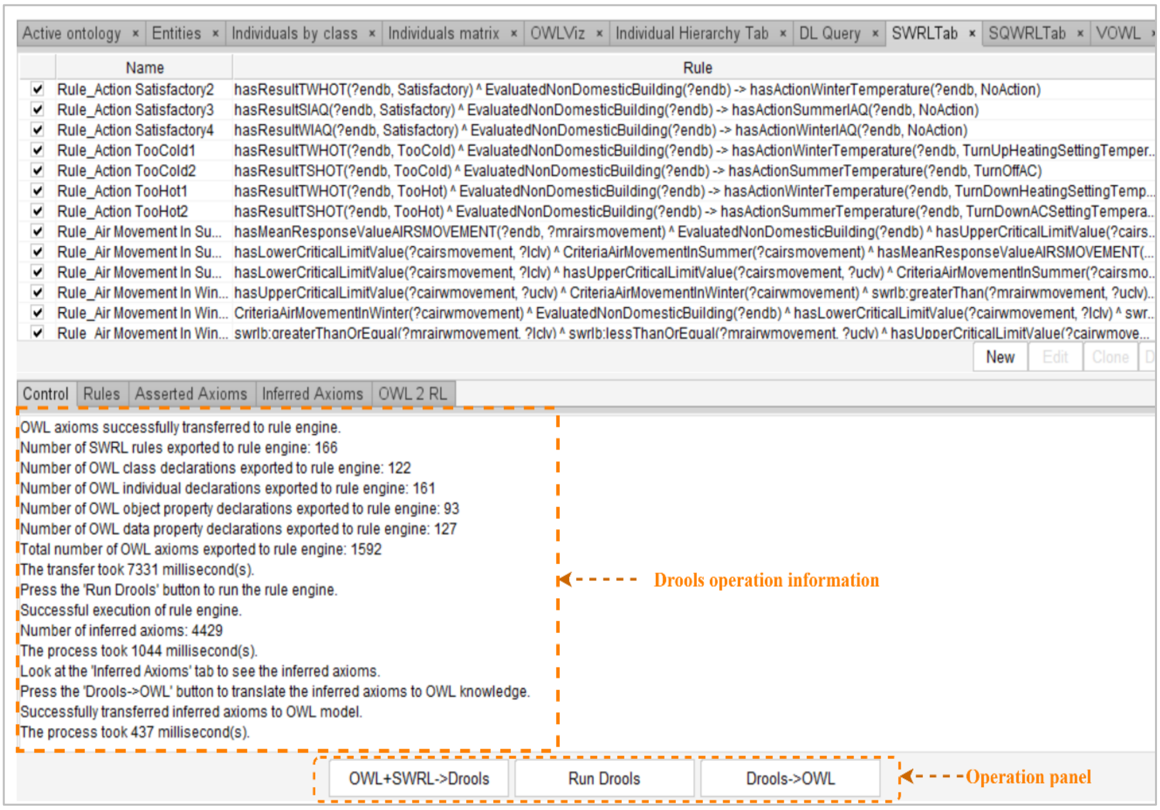


Figure 4-32 Drools rule engine

4.5.2 Query implementation

SQWRL queries are operated in combination with SWRL rules to retrieve the knowledge which is inferred by SWRL rules. The SQWRL queries have no access to the information it accumulates from within a rule, so the results from the SQWRL cannot be written back to the OWL ontology and do not perform any ontology modifications (O'Connor, 2016).

The execution of SQWRL query rules is as shown in Figure 4-33: Step 1 Select the query rules; Step 2 Run the rules; Step 3 Generate query results. After executing the query, as shown Figure 4-34, the retrieved results are listed in a new sub-tab, but the query results will not be written back to the OWL ontology.

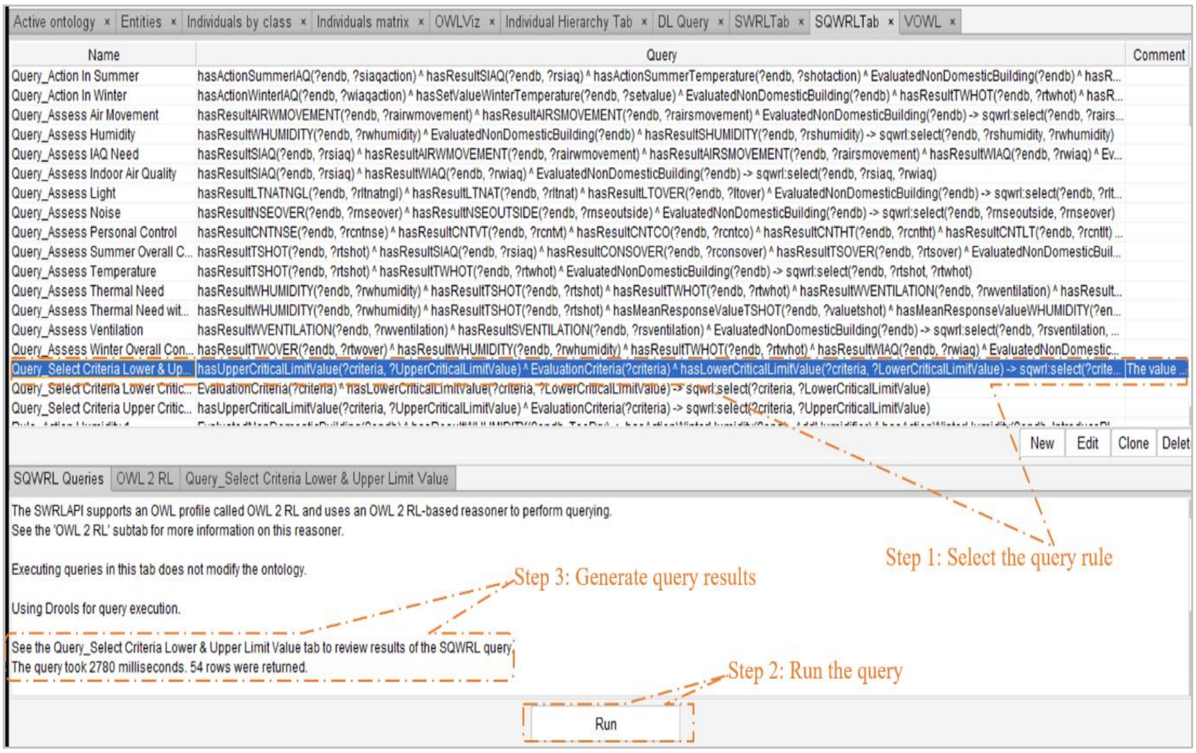


Figure 4-33 SQWRLTab rules query operation panel

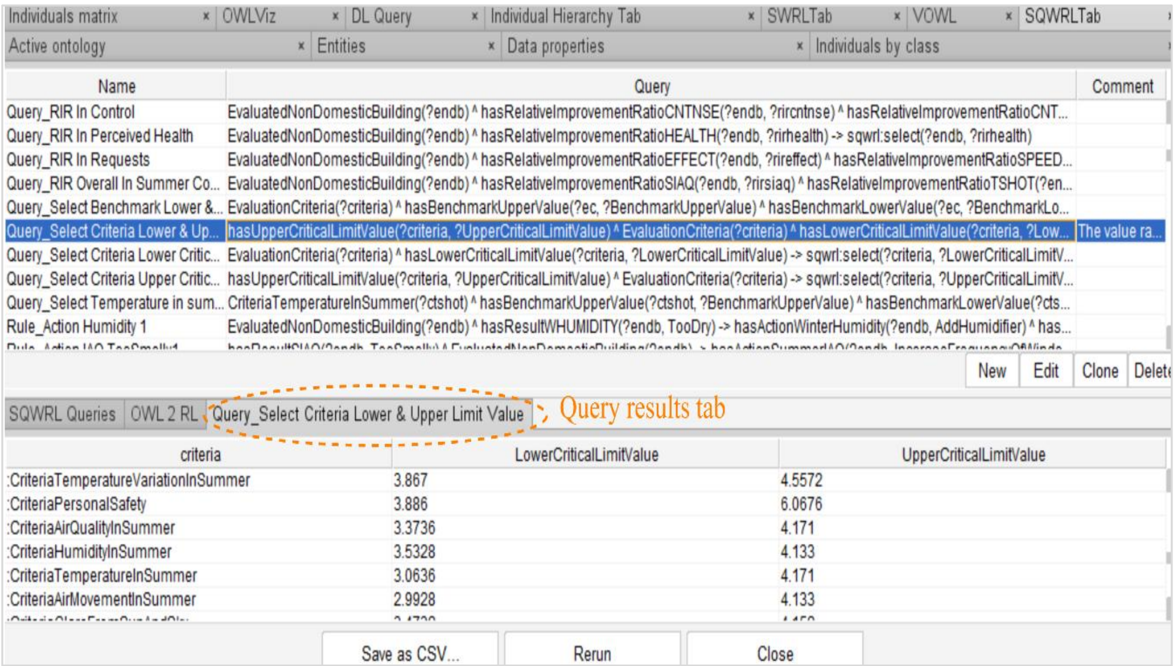


Figure 4-34 SQWRL query rules results example

4.6 Summary

This chapter illustrates the design and development of the proposed POE ontology. The adopted development methodologies are introduced in the previous chapter, and the development processes of this proposed POE ontology include knowledge base development, rules development, and ontology validation.

To develop the POE ontology knowledge base, the ‘7-Steps’ methodology from (Noy and McGuinness (2001) is adopted in this study. In addition, as a new ontology developer, the guides from Horridge (2011) and DeBellis (2021) on building OWL ontologies by using Protégé, have been followed to construct the proposed POE ontology from scratch in ontology editor Protégé. These practical guides provide detailed ontology construction illustrations by demonstrating how to build a pizza ontology, which is easy to understand for new developers. The outcome of this process is the knowledge base, and important post-occupancy evaluation related concepts are classified into the ontology class hierarchies. And a list of object properties and data properties is defined to represent the relationships and attributes between concepts in POE ontology.

The rules development process includes SWRL rules development and SQWRL Query rules development. The ontology editor Protége provides a set of plug-ins for defining, running and querying rules, and a set of SWRL built-ins are applied to support rules development. The POE assessment and query rules are developed under the SWRLTab in protégé, and the SQWRLTab can be used to query the user needed rules.

The ontology validation process is to validate the proposed ontology. The reasoner of Pellet is used to check the ontology and infer implicit new axioms of ontology from the given explicit facts in the knowledge base. The Drools rule engine can be used here to assert new knowledge into the ontology as axioms.

A field study has been conducted based upon the Building Use Study (BUS) methodology to validate the proposed ontology-based post-occupancy evaluation framework in the next chapter.

Chapter 5 Case study for framework validation

Chapter 4 explicitly discusses the development of the proposed ontology-based post-occupancy evaluation (POE) framework which is built on a POE knowledge base and a set of assessment SWRL rules and SQWRL rules. The Semantic Web technology has been adopted to develop a heavyweight ontology with the focus on building post-performance assessment through building occupants' satisfaction assessment on building parameters. In the POE ontology, the building occupants’ needs for building performance are generalized and classified, and the corresponded building performance assessment knowledge is formalized. In this chapter, a field case study is carried out based upon the Building Use Studies (BUS) Methodology to illustrate the validity and feasibility of this ontology framework. The required data for this evaluation is extracted from the field BUS survey, combined with SWRL rules from the ontology and exported to the reasoning engine.

Section 5.1 describes the case study in detail, including the licensed Building Use Studies (BUS) Methodology and the assessed sample buildings; the data collection and analysis and integration with ontology are presented in Section 5.2; Section 5.3 presents the ontology application scenarios and framework validation; Section 5.4 is the summary of this chapter.

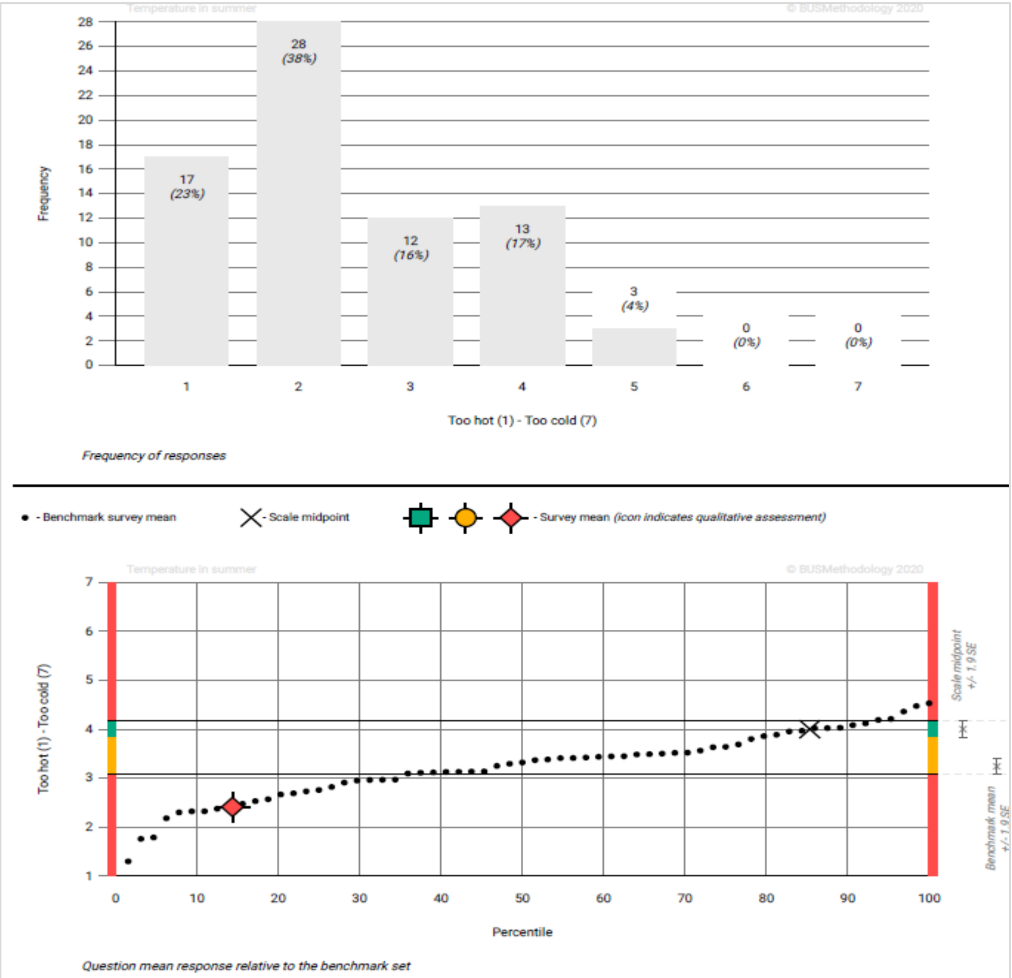
5.1 Case study description

This proposed framework can be used to evaluate different types of buildings, e.g. domestic and non-domestic buildings. Different types of buildings have different benchmark value datasets, which can be separately defined in the ontology model. As non-domestic buildings have more complex building systems and cover more comprehensive building performance evaluation criteria than domestic buildings, therefore, the developed method and framework can also be applied to domestic buildings. In this research, two non-domestic buildings are investigated to prove the effectiveness of the developed methods and the implementability of the proposed framework.

The licensed BUS Methodology questionnaire has been applied to carry out a survey assessment in two non-domestic buildings, the Eastern Gateway (ESGW) building and Michael Sterling (MCST) building, they are selected from a university in the United Kingdom. These two buildings are functionally similar, and the Eastern Gateway building has achieved a

BREEAM “Excellent” rating with heating provided by a wood pelleting boiler and much of the building using natural ventilation (Kawneer UK Limited, 2014).

The collected assessment data is analysed by comparison with similar buildings’ benchmark values from the database of BUS Methodology. The current non-domestic building benchmark database is established based on more than 850 buildings from around the world, and it has established a consistent dataset of quality resulting from over 70,000 participants' responses. By benchmarking the evaluated building data against a large database of results for similar buildings, the corresponding suggestions are created and decisions informed to improve occupant experience and optimise building performance (BUS METHODOLOGY, 2017).



	Mean	Percentile	SE	Mean - 1.9SE	Mean + 1.9SE
Question mean response	2.41	14.37	-	-	-
Benchmark mean	3.24	46.73	0.09	3.08	3.40
Scale midpoint	4.00	85.34	-	3.84	4.16

Figure 5-1 Result presentation in BUS Methodology

According to the BUS Methodology, the building performance assessment results are usually presented in the form of statistical tables, graphs and scatter plots, as shown in Figure 5-1. A slider graphic with Red, Amber, Green (RAG) markers is supplied for ease of interpretation. The green rectangle represents the mean values of assessed buildings are significantly better or higher than both the benchmark values; the amber oval means there is not much difference between the mean values of assessed buildings and the benchmark values; the red diamond means survey mean values are worse or lower than the benchmark value.

5.2 Data analysis and integration with ontology

To increase the response rate, this study adopts the paper-based questionnaire method, 150 paper-based questionnaires were handed out in each building, and a total of 78 and 88 valid questionnaires are collected back respectively from these two buildings, and the response rate is higher than 50%. The collected qualitative and quantitative data were analysed against similar buildings in the established dataset. By analyzing the collected non-benchmarked and benchmarked survey responses, this section illustrates the analysis results of this study.

5.2.1 Analysis of non-benchmarked survey responses

Table 5-1 shows non-benchmarked survey results in the aspects of age, gender, window seat, work duration in the building each day and every week, the number of people in the shared office or work area, etc. These two buildings are both office buildings and are functionally similar. Most of the assessment parameters have slight differences in results between these two buildings. However, there is a significant difference in the factor of the space shared with the people in offices, this factor is related to the noise factor and might be related to air quality in the room, etc. In building ESGW, there are 9% of people shared with 5-8 others, and up to 51% of occupants shared office with more than 8 people; however, in building MCST, these two proportions are 45% and 27% separately. The work duration in buildings each day and every week might be related to health, productivity, and so on. There are significant differences between these two buildings, more than 60% of occupants work more than 8 hours in the MCST, that number in the ECST is only 46%, and the seating to the window might be related to occupants lighting satisfaction. Surveying the sample background information would help to understand the correlations between these sample parameters with benchmarked survey results and minimize potential errors.

Table 5-1 Analysis of non-benchmarked survey response

Sample characteristics		Eastern Gateway	Michael Sterling
30 years of age or older		74%	75%
Gender	Female	65%	51%
	Male	35%	49%
Occupied in building for a year or more		74%	77%
Number of occupants in office or work area	occupied by yourself	11%	2%
	Shared with 1 person	11%	5%
	Shared with 2-4 person	14%	19%
	Shared with 5-8 person	9%	45%
	Shared with more than 8 person	51%	27%
Hours spent in the building each day	5 or less than 5 hours	1%	6%
	6 hours	11%	9%
	7 hours	39%	22%
	8 hours	39%	46%
	9 or more than 9 hours	7%	14%
Hours spent at desk each day	5 or less than 5 hours	21%	21%
	6 hours	33%	27%
	7 hours	27%	24%
	8 hours	13%	30%
	9 or more than 9 hours	3%	3%
Days spent in the building each week	2	--	1%
	3	6%	10%
	4	16%	16%
	5	77%	66%
	6	--	5%
Next to window		34%	35%

Conditions in building change behaviour	28%	32%
Requests for Changes	21%	35%

After having a general analysis of the background information of building occupants, part of the information has been encoded into the POE ontology model to build the building occupants' information profile, for example, the gender has been coded into the '*hasGender'* data property of building occupants; the age information has been converted to the *'hasAgeGroup'* data property of building occupants; the days spent in the building each week information is edited as the data property of '*hasDaysAtWork*' for building occupants; the desk space of occupants in the work area is related to the data property of '*hasMeanResponseValueSPACEDESK'* in ontology, and so on. The full detailed survey also covers the aspects of the journey time to work, the mode of travel, effect on behaviour changing, etc.

5.2.2 Analysis of benchmarked survey responses

The mean value of responses for each evaluation criterion in the survey is calculated using Excel, part of the core results are shown in Table 5-2.

Table 5-2 Mean value of evaluation criteria

Evaluation Criteria	Sub-Evaluation Criteria	ESGW	MCST
EvaluationCriteria ConditionsInSummer	CriteriaAirMovementInSummer	3.61	4.36
	CriteriaAirQualityInSummer	4.58	4.53
	CriteriaHumidityInSummer	3.38	3.07
	CriteriaTemperatureInSummer	2.77	2.41
	CriteriaTemperatureVariationInSummer	3.61	4.36
	CriteriaVentilationInSummer	5.04	4.88
	CriteriaThermalComfortInSummer	3.46	3.34
	CriteriaConditionsInSummerOverall	3.54	3.56
EvaluationCriteria ConditionsInWinter	CriteriaAirMovementInWinter	3.21	2.89
	CriteriaAirQualityInWinter	3.95	4.03
	CriteriaHumidityInWinter	3.11	3.14
	CriteriaTemperatureInWinter	4.11	4.84

	CriteriaTemperatureVariationInWinter	3.95	3.14
	CriteriaVentilationInWinter	4.53	4.16
	CriteriaThermalComfortInWinter	4.52	4.26
	CriteriaConditionsInWinterOverall	4.25	4.32
EvaluationCriteria PersonalControl	CriteriaPersonalControlOverCooling	2.16	1.74
	CriteriaPersonalControlOverHeating	2.01	1.62
	CriteriaPersonalControlOverLighting	3.41	4.51
	CriteriaPersonalControlOverNoise	1.62	1.78
	CriteriaPersonalControlOverVentilation	2.96	3.32
EvaluationCriteria Lighting	CriteriaAmountOfArtificialLight	4.44	4.25
	CriteriaAmountOfNaturalLight	3.65	3.99
	CriteriaGlareFromLights	3.27	3.00
	CriteriaGlareFromSunAndSky	3.69	3.70
	CriteriaLightingOverall	4.79	5.00
EvaluationCriteria Requests	CriteriaEffectivenessOfResponse	3.40	3.39
	CriteriaSpeedOfResponse	3.35	3.41
EvaluationCriteria Noise	CriteriaFrequencyOfInterruptions	3.75	3.14
	CriteriaNoiseFromColleagues	3.85	3.21
	CriteriaNoiseFromOutside	3.27	4.18
	CriteriaOtherNoiseFromInside	3.72	2.91
	CriteriaNoiseOverall	4.30	4.71
EvaluationCriteria BuildingOverall	CriteriaUsabilityOfFurniture	4.83	5.49
	CriteriaStorageArrangements	4.15	4.95
	CriteriaPersonalSafety	5.64	6.10
	CriteriaImageToVisitors	5.77	5.62
	CriteriaFacilitiesMeetNeeds	4.85	5.44
	CriteriaSpaceUse	4.54	5.12
	CriteriaCleaning	4.96	5.43
	CriteriaAvailabilityOfMeetingRooms	3.88	4.30
	CriteriaAdequacyOfSpaceAtWorkArea	4.59	5.25
	CriteriaProductivityAtWork	4.85	5.83

	CriteriaOverallComfort	4.59	5.09
	CriteriaPerceivedHealth	3.49	3.86
	CriteriaBuildingDesignOverall	4.89	5.08

The surveyed mean values in Table 5-2 are integrated into the ontology model as the asserted values of the data properties of evaluated buildings. For instance, the survey mean value of the building performance on cleaning is 5.43 for the Michael Sterling building, in terms of ontology, as shown in Figure 5-2, the value of 5.43 has been assigned to the *'hasMeanResponseValueCLEANING'* data property of the Michael Sterling building; the value of building performance in the aspect of indoor air quality in summer is 4.58 for Eastern Gateway building and 4.53 for Michael Sterling building, so in the POE OWL ontology, the statement of this property is '*BuildingEasternGateway hasMeanResponseValueSIAQ* 4.58', '*BuildingMichaelSterling hasMeanResponseValueSIAQ* 4.53', etc.

Two other critical values need to be applied to perform evaluation queries, namely the benchmark mean value and the standard error value, which are used to set up the assessment value limit thresholds for each building performance evaluation criteria. In this research, the benchmark mean value and the standard error value are adopted from the licensed BUS Methodology benchmark database. By following the agreement of license, the whole set of benchmark numbers can not reveal in this research (Leaman, 2011), therefore, this study used several sets of individual benchmark numbers that are permitted by the BUS Methodology to verify the POE model. After the case study values and the required benchmark values are integrated into the OWL ontology, the knowledge model of the case study for system validation now is completed.

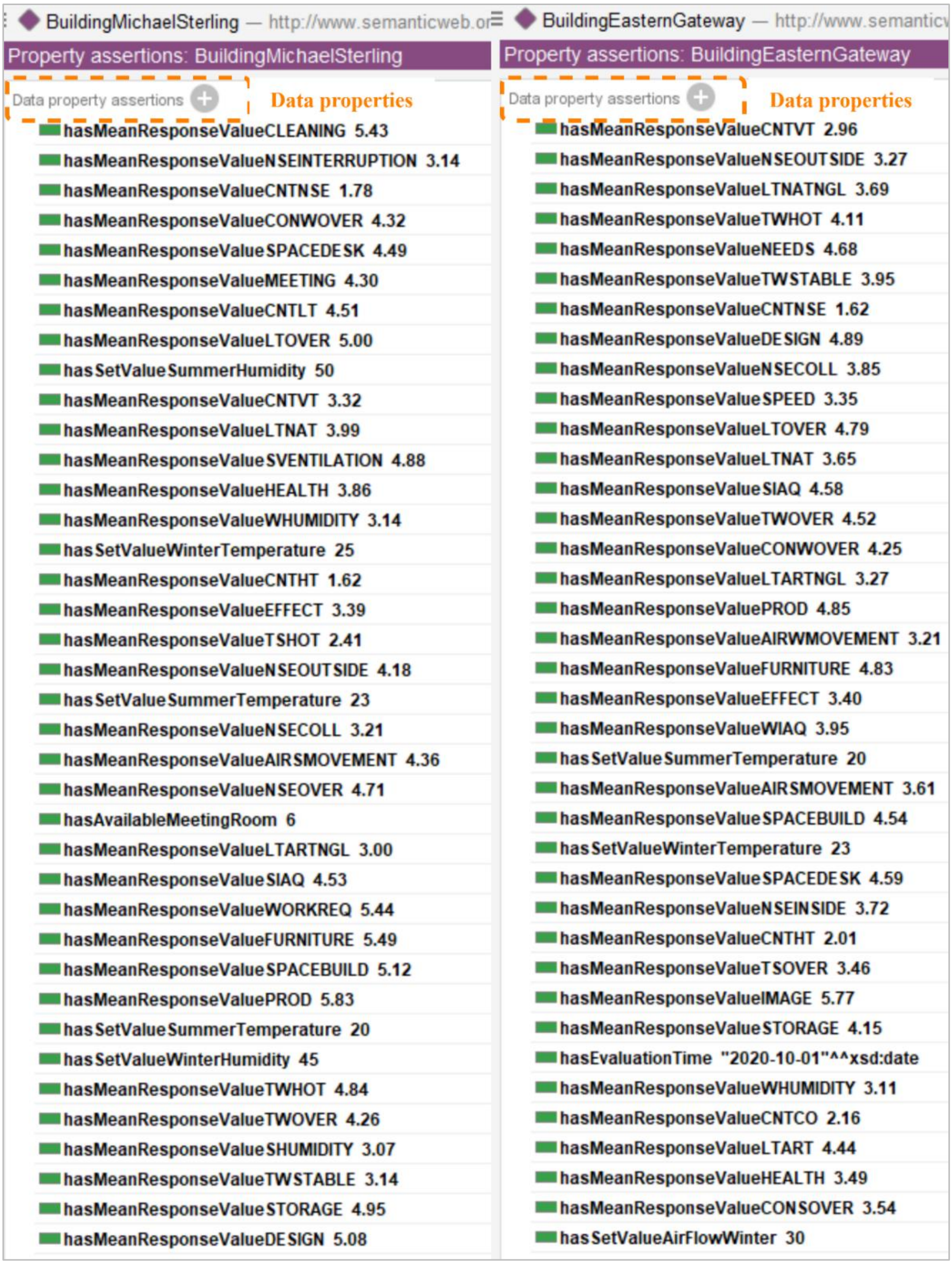


Figure 5-2 The value of data properties of evaluated buildings

5.3 Framework application scenarios

To verify the validity and feasibility of the proposed POE ontology framework, this section presents several application scenarios of this system after integrating the case study knowledge into the OWL ontology. The successful implementation of this system in different scenarios has proved the validity and feasibility of this POE ontology framework and the value of this research.

5.3.1 Knowledge transformation of data in POE OWL ontology

In this study, the critical benchmark mean values and the standard error values of evaluation criteria adopted from the licensed BUS Methodology benchmark database are taken as the initial facts for the datatype properties of evaluation criteria. For example, for the evaluation criteria temperature in summer, its benchmark mean value is 3.24, the SE is 0.09 and the scale midpoint value is 4.00. As shown in Figure 5-2, in the POE ontology knowledge base, these initial facts of the '*CritriaTemperatureInSummer'* instance have been stated as data properties of *hasBenchmarkMeanValue 3.24, hasSE 0.09, hasMidpintValue 4.00.*

The initial facts of instances are manually defined in the knowledge base of POE OWL ontology, and then a group of SWRL rules are defined with a set of SWRL built-ins to construct knowledge retrieval specifications. The implementation of knowledge transformation relies on the three functional buttons in the SWRLTab, as shown in Figure 5-3, based on the predefined SWRL rules, the relevant initial OWL axioms and SWRL rules are transferred to the rule engine after pressing the 'OWL+SWRL->Drools' button; and then after running the rule engine Drools, the new inferred axioms or facts are transferred back into OWL knowledge base by pressing the 'Drools->OWL' button. So far, the knowledge transformation of data is completed and the ontology is modified after executing these steps.

Figure 5-3 gives the details of the knowledge transformation process, such as the number of OWL axioms exported to the rule engine, the number of inferred axioms, the processing time, etc.

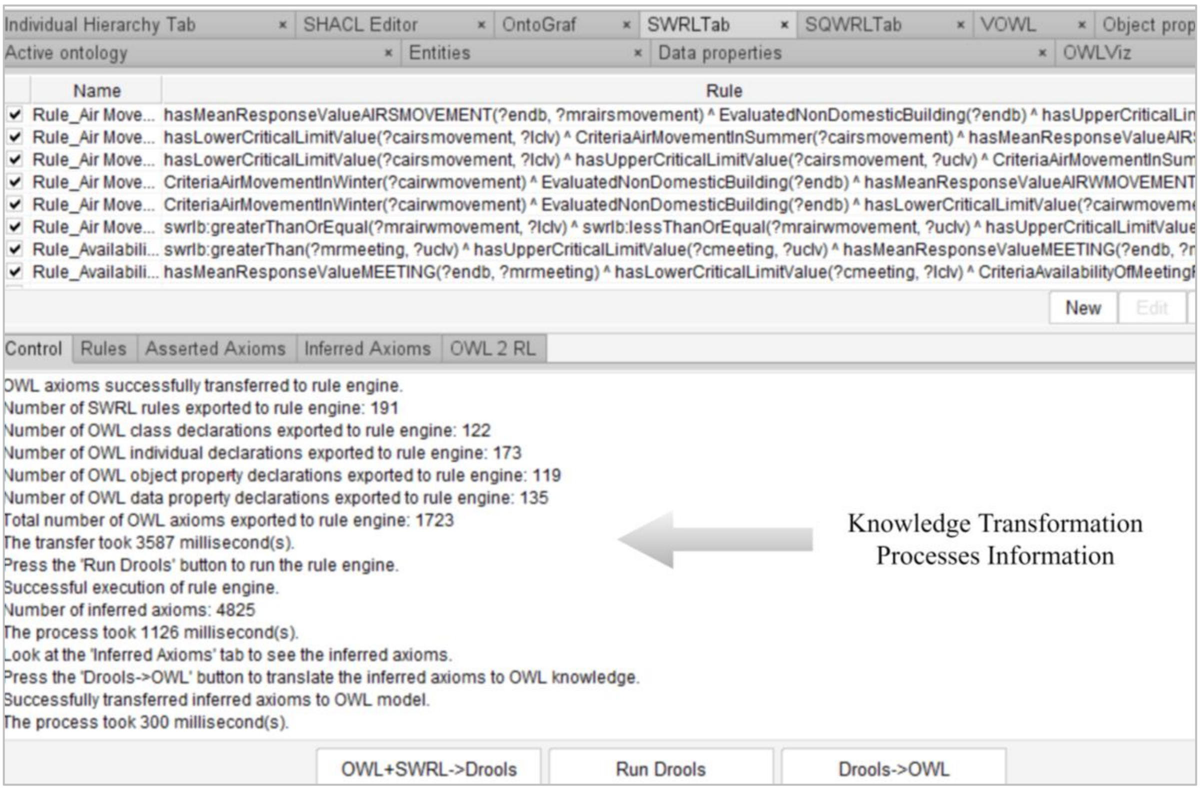


Figure 5-3 Knowledge transformation of data in the OWL ontology

A detailed example of inferring the lower and upper critical limit value of evaluation criteria is given below to demonstrate how the knowledge transformation of data between the initial facts and inferred facts is accomplished in this ontology. The evaluation criteria of temperature in summer has the initial facts of benchmark mean value 3.24, the SE 0.09 and the scale midpoint value 4.00, these initial facts have been pre-defined as data properties of criteria temperature in summer in the POE ontology knowledge base, as shown in Figure 5-4.

First, as introduced in Section 3.3 of Chapter 3, the SWRL rules for defining the lower and upper values of the benchmark mean value and scale midpoint value of evaluation criteria are established in the OWL knowledge base, relevant rules are shown in Table 5-3 and Table 5-4

Table 5-3 Lower & Upper Limit Value of Evaluation Criteria

Rule_Benchmark Lower Value1
EvaluationCriteria(?ec)^hasBenchamrkMeanValue(?ec,?bmv)^swrlb:multiply(?y,?se,1.96) ^ hasSE(?ec,?se) ^ swrlb:subtract(?x,?bmv,?y) -> hasBenchmarkLowerValue(?ec, ?x)
Rule_Benchmark Upper Value2

swrlb:multiply(?y,?se,1.96)^hasBenchamrkMeanValue(?ec,?bmv) ^ swrlb:add(?x,?bmv,?y) ^ hasSE(?ec, ?se) ^ EvaluationCriteria(?ec) -> hasBenchmarkUpperValue(?ec, ?x)
Rule_Sacle Midpoint Lower Value1
EvaluationCriteria(?ec)^swrlb:multiply(?y, ?se,1.96)^hasScaleMidpointValue(?ec, ?smv)^ hasSE(?ec, ?se)^ swrlb:subtract(?x,?smv,?y) -> hasScaleMidpointLowerValue(?ec, ?x)
Rule_Scale Midpoint Upper Value2
swrlb:multiply(?y,?se, 1.96) ^ swrlb:add(?x, ?smv, ?y) ^ hasScaleMidpointValue(?ec, ?smv) ^ hasSE(?ec, ?se) ^ EvaluationCriteria(?ec) -> hasScaleMidpointUpperValue(?ec, ?x)

Then, the SWRL rules for numerical comparisons defined above are shown in Table 5-4, these rules are used to determine the lower and upper critical limit value of evaluation criteria.

Table 5-4 SWRL rules for numerical comparisons

Rule_Lower Critical Limit Value1
EvaluationCriteria(?ec)^hasScaleMidpointLowerValue(?ec,?smlv)^swrlb:lessThan(?bmlv, ?smlv)^hasBenchmarkLowerValue(?ec,?bmlv) -> hasLowerCriticalLimitValue(?ec, ?bmlv)
Rule_Lower Critical Limit Value2
EvaluationCriteria(?ec)^hasScaleMidpointLowerValue(?ec,?smlv)^swrlb:lessThan(?smlv,?bmlv)^ hasBenchmarkLowerValue(?ec, ?bmlv) -> hasLowerCriticalLimitValue(?ec, ?smlv)
Rule_Upper Critical Limit Value1
EvaluationCriteria(?ec) ^ hasScaleMidpointUpperValue(?ec,?smuv) ^ hasBenchmarkUpperValue(?ec,?bmuv) ^ swrlb:greaterThan(?bmuv,?smuv) -> hasUpperCriticalLimitValue(?ec, ?bmuv)
Rule_Upper Critical limit Value2
EvaluationCriteria(?ec) ^ hasBenchmarkUpperValue(?ec,?bmuv) ^ hasScaleMidpointUpperValue(?ec,?smuv) ^ swrlb:greaterThan(?smuv,?bmuv) -> hasUpperCriticalLimitValue(?ec, ?smuv)

After determining all the relevant rules, the inferred new facts knowledge is automatically generated by following the rule engine execution process described previously. In this case, the new inferred facts of example instance are marked in red as shown in Figure 5-4, they are automatically transferred back to the OWL knowledge base as new data properties of the inferred individual.

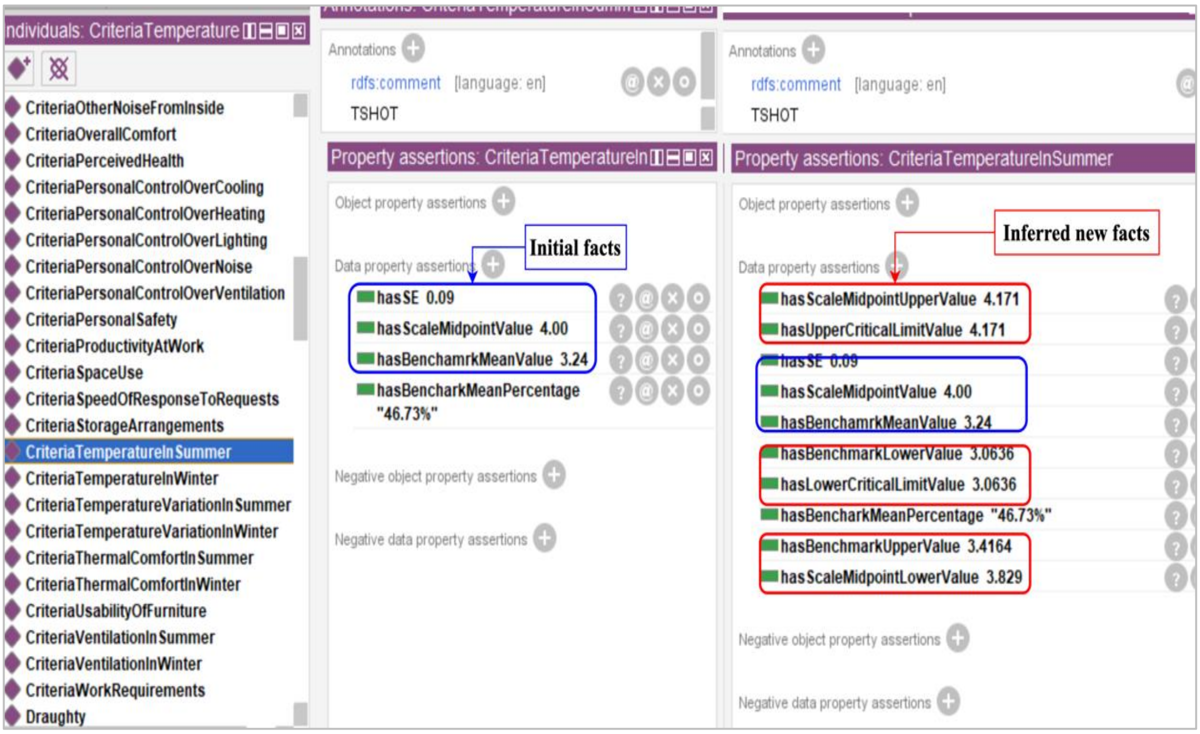


Figure 5-4 Initial facts and inferred new facts of evaluation criteria example

In addition to acquiring new facts by executing SWRL rules, the new facts can also be inferred through executing SQWRL queries in SQWRLTab, in this way, the inferred new knowledge will not be transferred back to the POE OWL ontology knowledge base, and this method does not modify the ontology.

To illustrate the SQWRL query method, the related query rule for inferring the relevant values of evaluation criteria temperature in summer is given in Table 5-5, and the result is generated in a new sub-tab, as shown in Figure 5-5.

Table 5-5 Query rule for benchmark data set of evaluation criteria

Query_Select Temperature in summer benchmark data set	
CriteriaTemperatureInSummer(?ctshot)	^
hasBenchmarkUpperValue(?ctshot,?BenchmarkUpperValue)	^
hasBenchmarkLowerValue(?ctshot,?BenchmarkLowerValue)	^
hasScaleMidpointUpperValue(?ctshot,?ScaleMidpointUpperValue)	^
hasScaleMidpointLowerValue(?ctsho,?ScaleMidpointLowerValue)	^
hasUpperCriticalLimitValue(?ctshot,?UpperCriticalLimitValue)	^
hasLowerCriticalLimitValue(?ctshot,?LowerCriticalLimitValue) ->	sqwrl:select

(?ctshot, ?BenchmarkLowerValue, ?BenchmarkUpperValue, ?ScaleMidpointLowerValue, ?ScaleMidpointUpperValue, ?LowerCriticalLimitValue, ?UpperCriticalLimitValue)

By following the same defining manner as explained above, a group of self-defined SQWRL query rules are established in the POE ontology. The query result can be returned as a single variable or a group of variables, it depends on the types and number of variables defined in the antecedent and consequent of rules. As shown in Figure 5-6, a group of results are generated by executing the SQWRL query rule the one is defined for inferring the lower and upper critical limit values of each evaluation criterion.

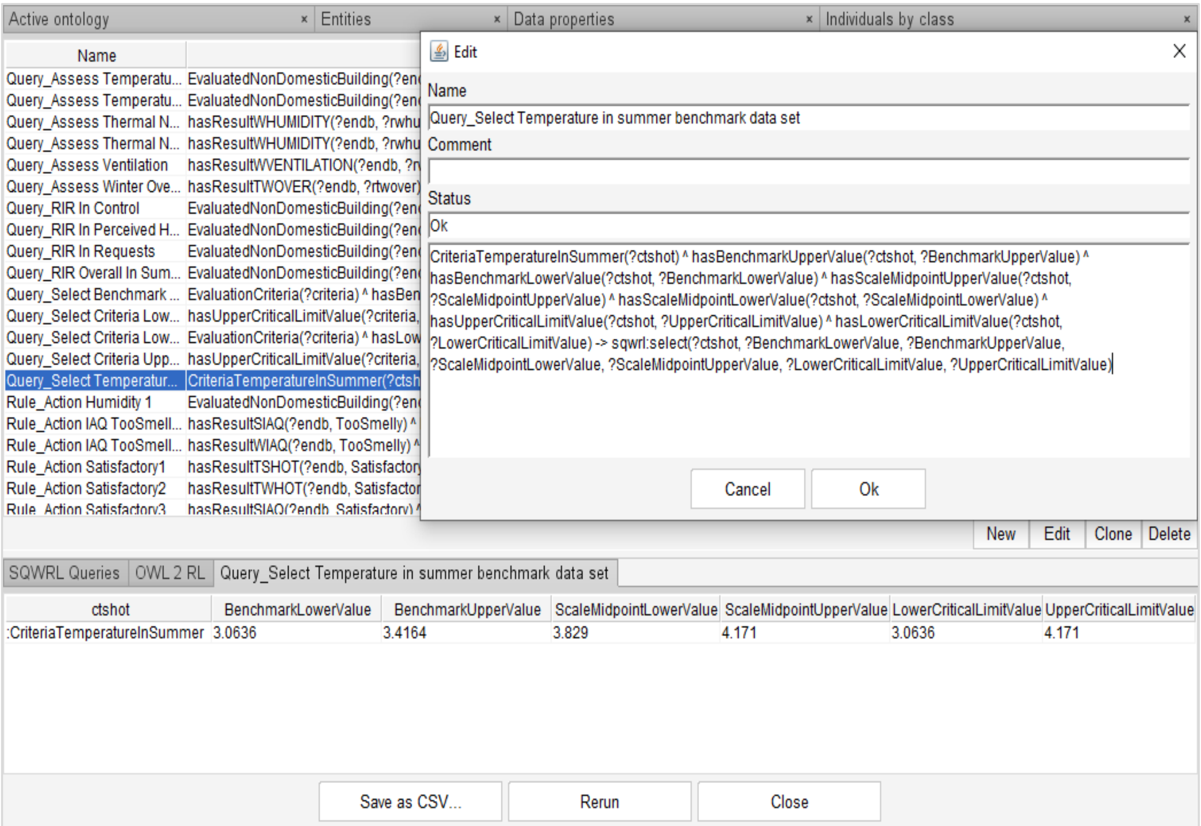


Figure 5-5 New facts inferred by SQWRL query

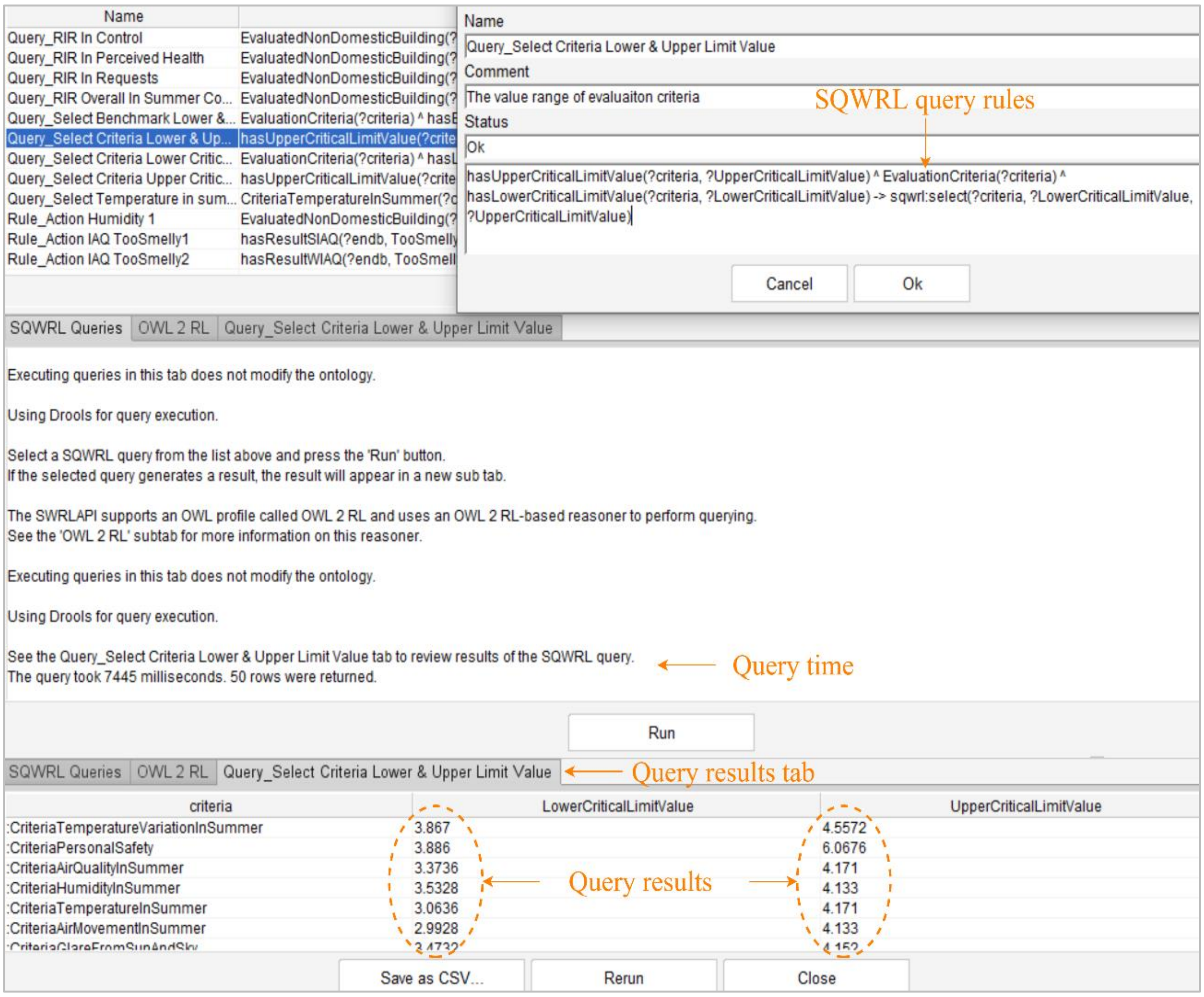


Figure 5-6 Query result set of critical values for evaluation criteria

5.3.2 Perform building assessment in POE OWL ontology

After determining the critical values for each evaluation criteria, a systematic POE of buildings can be carried out in the OWL ontology according to the pre-defined assessment principle SWRL rules as well as the asserted and inferred OWL axioms.

In this case, to demonstrate the feasibility of this application scenario, a building assessment is implemented in this POE ontology model in the aspects of temperature, humidity, air quality, temperature and ventilation, etc. At first, according to the SWRL rules for defining the lower and upper values of the benchmark mean value and scale midpoint value of evaluation criteria defined in Table 5-3, the inferred benchmark values are determined in the OWL knowledge base. To facilitate the analysis, the asserted and inferred benchmark values of evaluation conditions in summer are exported as in Table 5-6.

Table 5-6 The asserted and inferred benchmark values of conditions in summer

Criteria	Value	Mean	SE	Mean - 1.96SE	Mean + 1.96SE
Air Movement in Summer	Benchmark mean	3.13	0.07	2.9928	3.2672
	Scale midpoint	4.00	-	3.8628	4.1372
Air Quality in Summer	Benchmark mean	3.55	0.09	3.3736	3.7264
	Scale midpoint	4.00	-	3.8236	4.1764
Humidity in Summer	Benchmark mean	3.67	0.07	3.5328	3.8072
	Scale midpoint	4.00	-	3.8628	4.1372
Temperature in Summer	Benchmark mean	4.48	0.06	4.3624	4.5976
	Scale midpoint	4.00	-	3.8824	4.1176
Ventilation in Summer	Benchmark mean	4.58	0.09	4.4036	4.7564
	Scale midpoint	4.00	-	3.8236	4.1764
Conditions in Summer Overall	Benchmark mean	4.05	0.11	3.8344	4.2656
	Scale midpoint	4.00	-	3.7844	4.2156

According to the numerical comparison rules defined in Table 5-4 and the asserted and inferred benchmark values in Table 5-6, the lower and upper critical limit values of these criteria instances are determined. These values are stated as the OWL properties of these instances in ontology, the assessment constraints statements are shown in Table 5-7.

Table 5-7 The OWL properties of benchmark set of conditions in summer

Criteria	Thresholds	Result Level	Data Property	Data Value	Data Type
Air Movement in Summer (amis)	1≤ amis <2.9928	Still	hasLowerCriticalLimitValue	2.9928	xsd: decimal
	2.9928≤amis≤4.1376	Satisfactory	-	-	-
	4.1376≤ amis ≤7	TooDraughty	hasUpperCriticalLimitValue	4.1372	xsd: decimal
Air Quality in Summer (aqis)	1≤ aqis <3.3736	Outstanding	hasLowerCriticalLimitValue	3.3736	xsd: decimal
	3.3736≤aqis≤4.1746	Satisfactory	-	-	-
	4.1746< aqis ≤7	Smelly	hasUpperCriticalLimitValue	4.1746	xsd: decimal

Humidity in Summer (his)	1≤ his <3.5328	TooDry	hasLowerCritical LimitValue	3.5328	xsd: decimal
	3.5328≤his≤4.6137	Satisfactory	-	-	-
	4.1372< his ≤7	TooHumid	hasUpperCritical LimitValue	4.1372	xsd: decimal
Temperature in Summer (tis)	1≤ tis <3.8824	TooHot	hasLowerCritical LimitValue	3.8824	xsd:decimal
	3.88≤ tis≤4.5976	Satisfactory	-	-	-
	4.5976< tis ≤7	TooCold	hasUpperCritical LimitValue	4.5976	xsd: decimal
Ventilation in summer (vis)	1≤ vis <3.8236	Outstanding Fresh	hasLowerCritical LimitValue	3.8236	xsd: decimal
	3.8236≤vis≤4.7564	Satisfactory	-	-	-
	4.7564< vis ≤7	Stuffy	hasUpperCritical LimitValue	4.7564	xsd: decimal
Conditions in Summer Overall (ciso)	1≤ ciso <3.7844	Unsatisfactory	hasLowerCritical LimitValue	3.7844	xsd: decimal
	3.7844≤ciso≤4.2651	Satisfactory	-	-	-
	4.2651< ciso ≤7	Outstanding	hasUpperCritical LimitValue	4.2651	xsd: decimal

Following the SWRL syntax, all the POE assessment constraints are transferred into SWRL rules facilitated with a set of comparison built-ins, such as the *swrlb:lessThan, swrlb:greaterThan, swrlb:lessThanOrEqual, swrlb:greaterThanOrEqual*. In this example, take the indoor air quality and temperature in summer as examples, the corresponding assessment constraints SWRL rules are written in Table 5-8.

Table 5-8 POE assessment SWRL rules example

Rule_Indoor Air Quality In Summer1
CriteriaAirQualityInSummer(?csiaq) ^ hasMeanResponseValueSIAQ(?endb, ?msiaq) ^ EvaluatedNonDomesticBuilding(?endb) ^ isEvaluatedBy(?endb, ?csiaq) ^ hasUpperCriticalLimitValue(?csiaq, ?uclv) ^ swrlb:greaterThan(?msiaq, ?uclv) -> hasResultSIAQ(?endb, TooSmelly)
Rule_Indoor Air Quality In Summer2
EvaluatedNonDomesticBuilding(?endb) ^ isEvaluatedBy(?endb, ?csiaq) ^ hasLowerCriticalLimitValue(?csiaq, ?lclv) ^ swrlb:lessThanOrEqual(?msiaq, ?uclv) ^

swrlb:greaterThanOrEqual(?msiaq, ?lclv) ^ CriteriaAirQualityInSummer(?csiaq) ^ hasMeanResponseValueSIAQ(?endb,?msiaq) ^ hasUpperCriticalLimitValue(?csiaq, ?uclv) -> hasResultSIAQ(?endb, Satisfactory)
Rule_Indoor Air Quality In Summer3
CriteriaAirQualityInSummer(?csiaq) ^ hasMeanResponseValueSIAQ(?endb, ?msiaq) ^ hasLowerCriticalLimitValue(?csiaq, ?lclv) ^ swrlb:lessThan(?msiaq, ?lclv) ^ EvaluatedNonDomesticBuilding(?endb) ^ isEvaluatedBy(?endb, ?csiaq) -> hasResultSIAQ(?endb, Outstanding)
Rule_Temperature In Summer1
CriteriaTemperatureInSummer(?ctshot) ^ hasLowerCriticalLimitValue(?ctshot, ?lclv) ^ hasMeanResponseValueTSHOT(?endb, ?mrtshot) ^ swrlb:lessThan(?mrtshot, ?lclv) EvaluatedNonDomesticBuilding(?endb) ^ isEvaluatedBy(?endb, ?ctshot) ^ -> hasResultTSHOT(?endb, TooHot)
Rule_Temperature In Summer2
EvaluatedNonDomesticBuilding(?endb) ^ hasUpperCriticalLimitValue(?ctshot, ?uclv)^ isEvaluatedBy(?endb, ?ctshot) ^ hasMeanResponseValueTSHOT(?endb,?mrtshot) ^ swrlb:greaterThanOrEqual(?mrtshot, ?lclv) ^ CriteriaTemperatureInSummer(?ctshot) ^ hasLowerCriticalLimitValue(?ctshot, ?lclv) ^ swrlb:lessThanOrEqual(?mrtshot, ?uclv) -> hasResultTSHOT(?endb, Satisfactory)
Rule_Temperature In Summer3
EvaluatedNonDomesticBuilding(?endb)^hasMeanResponseValueTSHOT(?endb,?mrtshot) ^ CriteriaTemperatureInSummer(?ctshot) ^ hasUpperCriticalLimitValue(?ctshot,?uclv) ^ isEvaluatedBy(?endb, ?ctshot) ^ swrlb:greaterThan(?mrtshot, ?uclv) -> hasResultTSHOT(?endb, TooCold)

According to the predefined assessment constraints SWRL rules in POE ontology, the corresponding SQWRL queries rules can be self-defined to retrieve user needed information from the OWL knowledge base. As the SQWRL query have no access to the information it accumulates from within a rule, the outputs of SQWRL queries will not be transferred back to the OWL ontology and do not modify the ontology. As introduced in Chapter 2, the SQWRL queries can be performed in combination with SWRL rules to retrieve the new knowledge which is inferred through SWRL rules.

In this case, the SQWRL rule for querying the assessment results of the above-defined conditions in summer is shown in Table 5-9.

Table 5-9 Query rule for conditions in summer assessment results

Query_Assess Conditions in summer overall
EvaluatedNonDomesticBuilding(?endb) ^ hasResultCONSOVER(?endb, ?rconsover) ^ hasResultAIRSMOVEMENT(?endb, ?rsairmove) ^ hasResultTSHOT(?endb, ?rtshot) ^ hasResultSIAQ(?endb, ?rsiaq) ^ hasResultSHUMIDITY(?endb, ?rshumidity) ^ hasResultSVENTILATION(?endb, ?rsventilation) -> sqwrl:select (?endb, ?rshumidity, ?rsairmove, ?rtshot, ?rsiaq, ?rsventilation, ?rconsover)

After executing the query, the outputs are listed in a subtab, as shown in Figure 5-7. The evaluated buildings of Michael Sterling and Eastern Gateway have the same satisfactory results in humidity (TooDry), temperature (TooHot), IAQ (TooSemlly), ventilation (Stuffy) and overall satisfaction (Unsatisfactory), but in the air movement aspect, the former has a result of TooDraughty (unsatisfactory), the latter has a result of satisfactory. This query rule is defined to output the overall satisfaction results for each building performance evaluation indicator, without listing the details of the results.

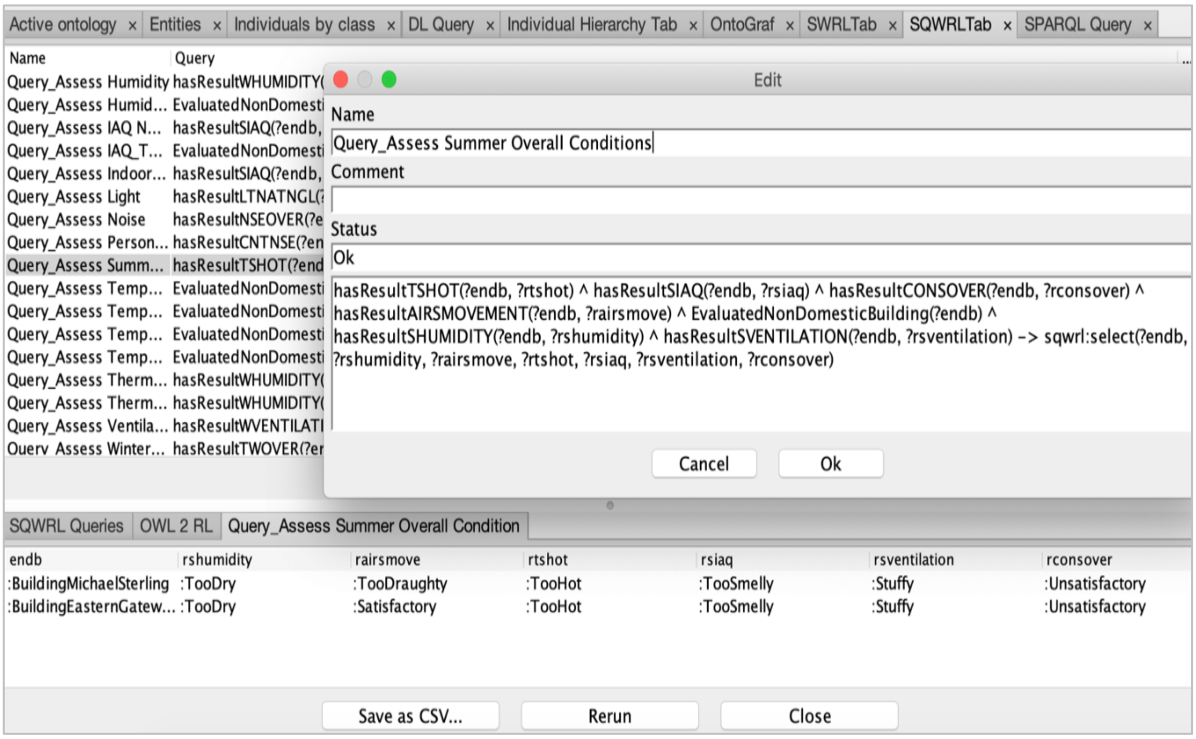


Figure 5-7 Query results for conditions in summer assessment

The aim to carry out a POE assessment for buildings is to effectively identify the building operations that perform well and those that do not, and to provide detailed feedback on buildings' overall performance, even propose the optimised operation improvement advice.

To provide more details on results, the following rule in Table 5-10 is defined to output the results of assessed variables with a set of assessment constraints values. The assessment results of temperature in summer and winter are given with the evaluated buildings' response mean value and the benchmark values of these two criteria. In this example, the temperature in winter has a lower critical limit value (lclvw) of 3.8860 and a higher critical limit value (uclvw) of 4.5976; the evaluated Eastern Gateway building has a response mean value of temperature in winter (mrtwhot) of 4.11, which is in the satisfactory thresholds zone; but the response mean value of temperature in winter (mrtwhot) for Michael Sterling building is 4.84, which is greater than the higher critical limit value (uclvw) of 4.5976, so its result is too cold, as shown in Figure 5-8.

Table 5-10 Combination query rules in temperature assessment

Query_Assess Temperature in summer and winter results
EvaluatedNonDomesticBuilding(?endb)^hasMeanResponseValueTSHOT(?endb,?mrtshot) ^hasResultTSHOT(?endb, ?rtshot) ^ hasMeanResponseValueTWHOT(?endb, ?mrtwhot) ^ hasLowerCriticalLimitValue(?ctshot, ?lclvs) ^hasUpperCriticalLimitValue(?ctshot, ?uclvs) ^ CriteriaTemperatureInSummer(?ctshot) ^ hasLowerCriticalLimitValue(?ctwhot, ?lclvw) ^ hasResultTWHOT(?endb, ?rtwhot) ^ CriteriaTemperatureInWinter(?ctwhot) ^ hasUpperCriticalLimitValue(?ctwhot, ?uclvw) -> sqwrl:select (?endb, ?mrtshot, ?lclvs, ?uclvs, ?rtshot, ?mrtwhot, ?lclvw, ?uclvw, ?rtwhot)

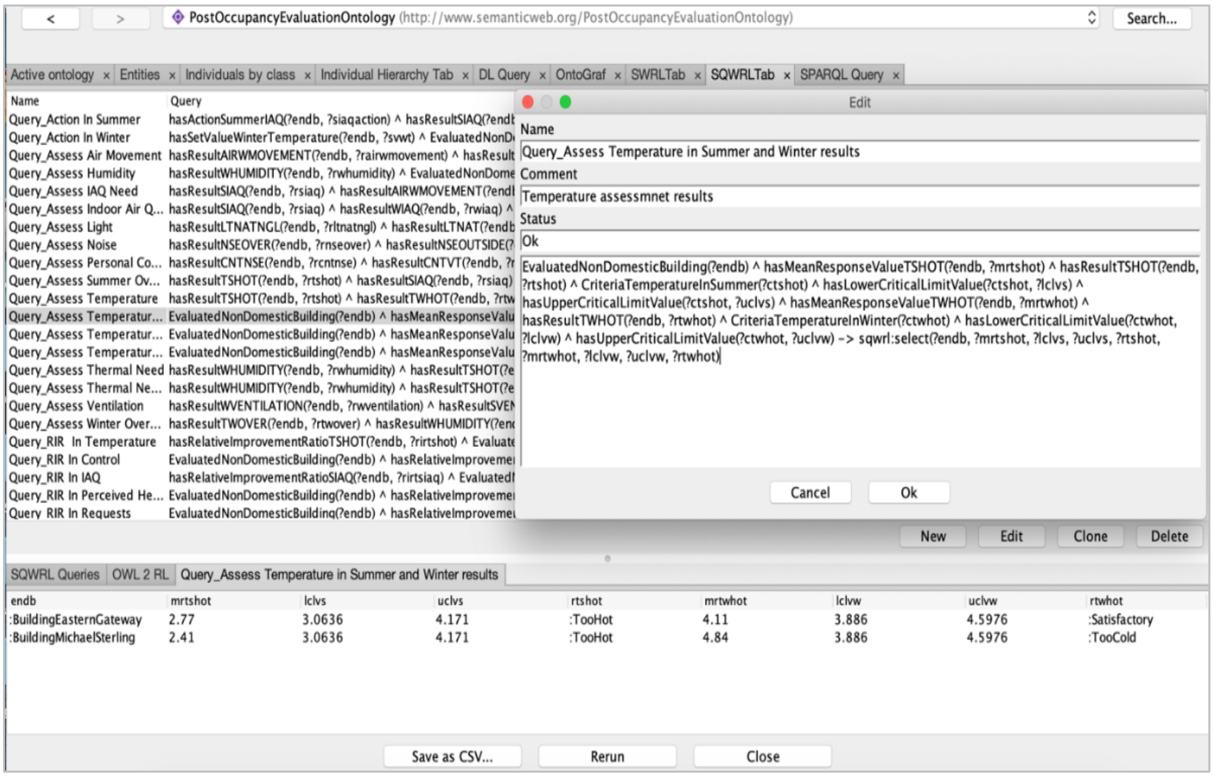


Figure 5-8 Query results combination

Following the same syntax, a comprehensive query rule is defined in Table 5-11 based on the multiple evaluation criteria in the summer listed in Table 5-7. As only a maximum of 11 arguments can be passed to built-ins at once, this rule is written with multiple *sqwrl:select* operators.

Table 5-11 Query rule for summer conditions results with data

Query_Summer conditions results with data
EvaluatedNonDomesticBuilding(?endb)^hasMeanResponseValueTSHOT(?endb,?mrtshot) ^ hasResultTSHOT(?endb,?rtshot) ^ CriteriaTemperatureInSummer(?ctshot) ^ hasLowerCriticalLimitValue(?ctshot,?lclvst) ^hasUpperCriticalLimitValue(?ctshot,?uclvst) ^ hasMeanResponseValueSIAQ(?endb, ?mrsiaq) ^ hasResultSIAQ(?endb, ?rsiaq) ^ CriteriaAirQualityInSummer(?csiaq) ^ hasLowerCriticalLimitValue(?csiaq, ?lclvsiaq) ^ hasUpperCriticalLimitValue(?csiaq,?uclvsiaq) ^ CriteriaHumidityInSummer(?cshum) ^ hasMeanResponseValueSHUMIDITY(?endb,?mrshum) ^ hasResultSHUMIDITY (?endb,?rshum) ^ hasLowerCriticalLimitValue(?cshum,?lclvsh) ^ hasUpperCriticalLimitValue(?cshum, ?uclvsh) ^ CriteriaVentilationInSummer(?csven)^ hasLowerCriticalLimitValue(?csven, ?lclvsven) ^ hasUpperCriticalLimitValue

(?csven, ?uclvsven) ^ hasMeanResponseValueSIAQ(?endb, ?mrsven) ^
hasMeanResponseValueAIRSMOVEMENT(?endb, ?mrsairmove) ^
hasResultSVENTILATION(?endb, ?rsven) ^ CriteriaAirMovementInSummer(?csairmove)
^ hasLowerCriticalLimitValue(?csairmove, ?lclvsairmove) ^ hasUpperCriticalLimitValue
(?csven, ?uclvsairmove) ^ hasResultAIRSMOVEMENT(?endb, ?rsairmove) ->
sqwrl:select(?endb,?mrtshot, ?lclvst, ?uclvst, ?rtshot, ?mrsiaq, ?lclvsiaq, ?uclvsiaq, ?rsiaq)
^ sqwrl:select(?mrshum, ?lclvsh, ?uclvsh, ?rshum, ?mrsven, ?lclvsven, ?uclvsven, ?rsven)
^ sqwrl:select(?mrsairmove, ?lclvsairmove, ?uclvsairmove, ?rsairmove)

The outputs after running this query are shown in Figure 5-9, which contains multiple variables of the lower critical limit value (lclv) and higher critical limit value (uclv) of each evaluation criterion, as well as the mean response values of evaluated buildings.

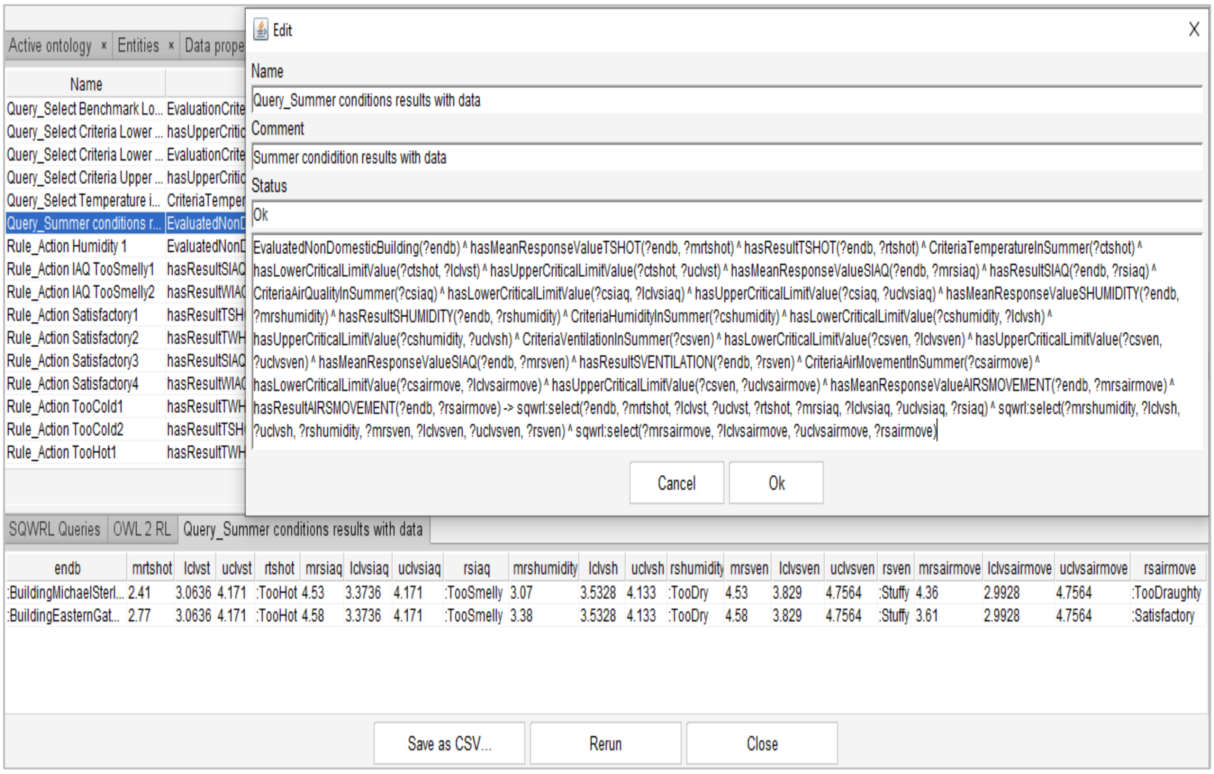


Figure 5-9 Query results of summer conditions with data

The examples demonstrated above illustrate that, based on the combination of SWRL rules and SQWRL queries, the proposed OWL POE ontology can support not only single criterion building assessment with simple queries, but also has the capability to conduct a systematic method for multi-criteria and multi-tasks building assessments.

Based on this step, this ontology can produce a complete evaluation report, in which, both the good building performances that meet building occupants' needs and the poor building performances that do not meet occupants' needs are identified and categorized. The well-performed building performance indicators can be used as the benchmark for future operations, however, for the poor building performances, the corresponding optimizations should be provided for future maintenance and operations.

5.3.3 Application for ranking building performance optimization indicator

Further building operation optimization concepts are encoded into the POE ontology based on the evaluation query results from the previous step. As introduced in the previous section, after executing the POE assessment queries, the results of evaluated buildings are generated, however, from the outputs in Figure 5-7, it is hard to tell how different the performance gap is between evaluated buildings and benchmark buildings. To explicitly identify the performance gap, the concept of relative improvement ratio (RIR) is introduced in this example to show the impact degree of certain building performance on building occupants' satisfaction, and also indicate how big the performance gap is between the evaluated buildings and the benchmark buildings. As introduced in Chapter 5, if the assessment results are better than the benchmark data, then this RIR indicator can be used to indicate how better evaluated buildings performed than benchmark buildings in certain areas, the bigger the RIR is the better performance buildings have; if the assessment results are unsatisfactory, then this RIR indicator is used to represent how much improvement the evaluated buildings need in terms of poor performances, at the same time, it also shows the impact degree of poor performances on the overall satisfaction of building occupants, the bigger the RIR is the poorer performance buildings have. The RIR indicator can be taken as the reference for building optimization priority.

The editing principle of SWRL rules for the RIR indicator is introduced in Chapter 4. In this case, take the summer conditions of IAQ, temperature, humidity, ventilation, air movement as examples to illustrate the definition of RIR SWRL rules, as shown in Table 5-12. The SWRL rules of the RIR for the rest of the evaluation criteria are defined in the same schema.

Table 5-12 SWRL rules of RIR for summer conditions

Rule_RIR IAQ in Summer
EvaluatedNonDomesticBuilding(?endb) ^ hasMeanResponseValueSIAQ(?endb, ?mrsiaq) ^ swrlb:subtract(?rgsiaq, ?bmsiaq, ?mrsiaq) ^ swrlb:divide(?rirsiaq, ?argsiaq, ?mrsiaq) ^ CriteriaAirQualityInSummer(?csiaq) ^ hasBenchamrkMeanValue(?csiaq, ?bmsiaq) ^ swrlb:abs(?argsiaq, ?rgsiaq) -> hasRIRSIAQ(?endb, ?rirsiaq)
Rule_RIR Temperature in Summer
EvaluatedNonDomesticBuilding(?endb)^hasMeanResponseValueTSHOT(?endb,?mrtshot) ^hasBenchamrkMeanValue(?ctshot,?bmtshot)^swrlb:subtract(?rgtshot,?bmtshot,?mrtshot)^ swrlb:abs(?argtshot,?rgtshot) ^ CriteriaTemperatureInSummer(?ctshot) ^ swrlb:divide (?rirtshot, ?argtshot, ?mrtshot) -> hasRIRTSHOT(?endb, ?rirtshot)
Rule_RIR Humidity in Summer
EvaluatedNonDomesticBuilding(?endb) ^ hasBenchamrkMeanValue(?cshum, ?bmshum) ^ CriteriaHumidityInSummer(?cshum) ^ swrlb:subtract(?rgshum,?bmshumidity, ?mrshum) ^ hasMeanResponseValueSHUMIDITY(?endb, ?mrshum) ^ swrlb:abs(?argshum,?rgshum) ^swrlb:divide(?rirshum,?argshum,?mrshum) ->hasRIRSHUMIDITY(?endb, ?rirshumidity)
Rule_RIR Ventilation in Summer
EvaluatedNonDomesticBuilding(?endb)^hasMeanResponseValueSVENTILATION(?endb, ?mrsventilation) ^ swrlb:subtract(?rgsventilation, ?bmsventilation, ?mrsventilation) ^ swrlb:abs(?argsventilation, ?rgsventilation)^CriteriaVentilationInSummer(?csventilation) ^ hasBenchamrkMeanValue(?csventilation, ?bmsventilation)^swrlb:divide(?rirsventilation, ?argsventilation,?mrsventilation)->hasRIRSVENTILATION(?endb, ?rirsventilation)
Rule_RIR Air Movement in Summer
EvaluatedNonDomesticBuilding(?endb) ^ hasMeanResponseValueAIRSMOVEMENT (?endb, ?mrsairmove)^swrlb:subtract(?rgsairmove, ?bmsairmove, ?mrsairmove) ^ swrlb:abs (?argsairmove, ?rgsairmove) ^ hasBenchamrkMeanValue(?csairmove, ?bmsairmove) ^ CriteriaAirMovementInSummer(?csairmove)^swrlb:divide(?rirsairmove, ?argsairmove, ?mrsairmove) -> hasRelativeImprovementRatioAIRSMOVEMENT(?endb, ?rirsairmove)

The composite SQWRL query rule for querying the RIR indicator of summer conditions is edited as in Table 5-13. The users can define the number of variables to be queried according to their query needs.

Table 5-13 Query rule for RIR of summer conditions

Query_RIR Overall In Summer Conditions
EvaluatedNonDomesticBuilding(?endb)^hasRIRSVENTILATION(?endb, ?rirsventilation) ^hasRIRTSHOT(?endb, ?rirtshot)^hasRIRSIAQ(?endb,?rirsiaq)^hasResultTSHOT(?endb, ?rtshot) ^ hasRIRSHUMIDITY(?endb, ?rirshumidity) ^ hasResultSIAQ(?endb, ?rsiaq) ^ hasRIRAIRSMOVEMENT(?endb, ?rirsairmove) ^ hasResultSHUMIDITY(?endb,?rshum) ^hasResultAIRSMOVEMENT(?endb,?rsairmove)^hasResultSVENTILATION(?endb, ?rsventilation)->sqwrl:select(?endb, ?rtshot, ?rirtshot, ?rsiaq,?rirsiaq, ?rshum,?rirshum, ?rsventilation, ?rirsventilation,?rsairmove,?rirsairmove)

Figure 5-10 shows the query results after performing the RIR query rule in Table 5-13. For the Michael Sterling building, the temperature factor has the highest RIR value of 0.34, followed by the air movement of 0.28, the IAQ of 0.22, the humidity of 0.20, and the ventilation of 0.06, this means that in summer conditions, the temperature has the worst impact on building occupants satisfaction, while ventilation has the least impact. Also, it indicates that in summer, this building has the worst performance in temperature aspect. However, for the Eastern Gateway building, the indoor air quality has the worst performance on building occupant satisfaction.

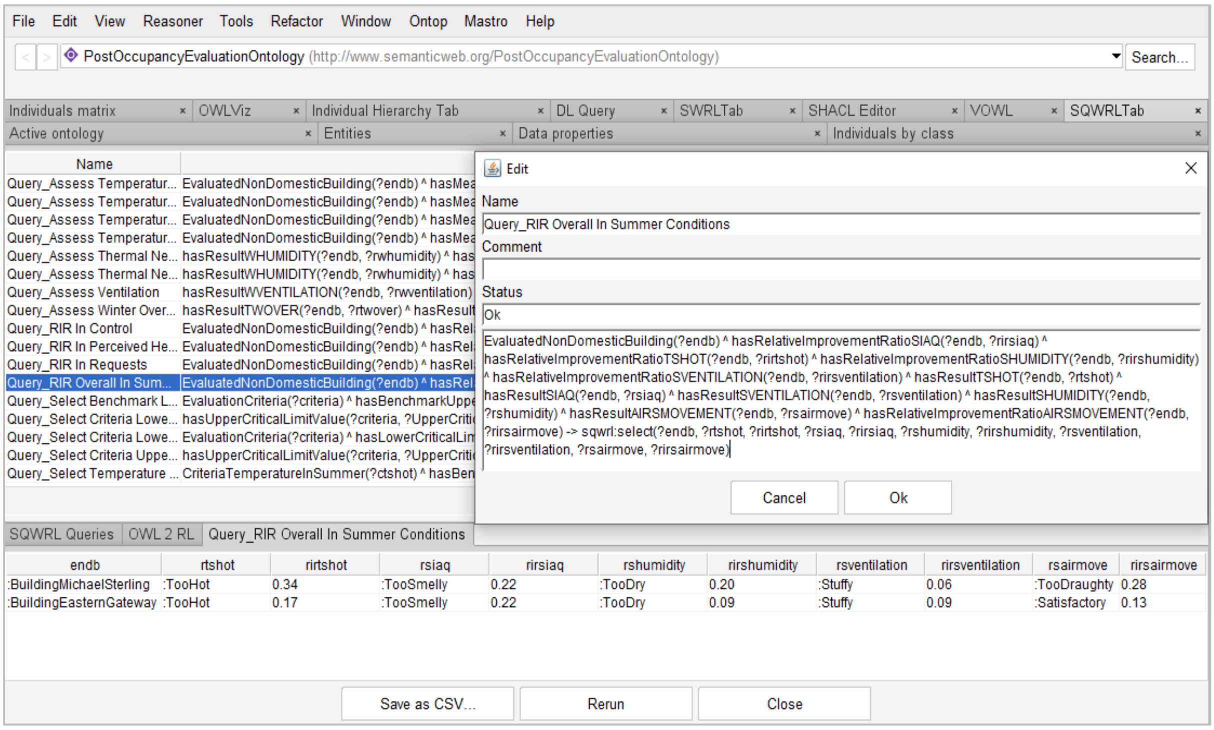


Figure 5-10 Query results for RIR of summer conditions

Based on all the OWL axioms asserted in the proposed POE ontology knowledge base, the self-defined POE constraints-based SWRL rules and the specified SQWRL query rules for POE assessments, a complete case study evaluation is carried out in this ontology. In this case, the outputs of the overall building RIR query are shown in Figure 5-11.

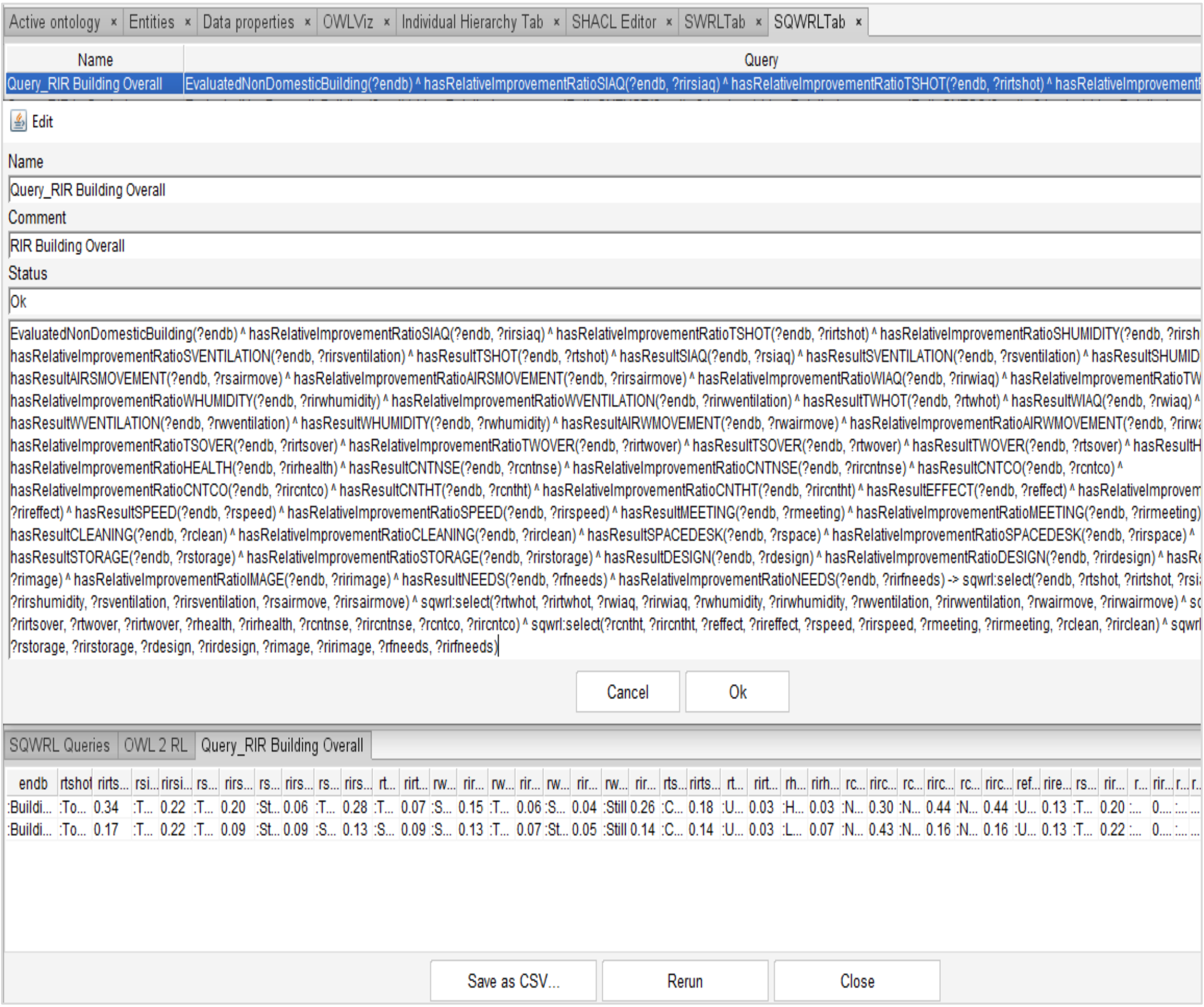


Figure 5-11 RIR query results for overall building performance indicators

For convenience, the outputs of the above query can be exported in the CSV format from the SQWRLTab, as shown in Table 5-14.

Table 5-14 RIR query results for overall building performance indicators

Indicators	**BuildingEasternGateway**		**BuildingMichaelSterling**	
	Result	RIR	Result	RIR
rtshot	TooHot	0.17	TooHot	0.34

rsiaq	TooSmelly	0.22	TooSmelly	0.22
rshumidity	TooDry	0.09	TooDry	0.20
rsventilation	Stuffy	0.09	Stuffy	0.06
rsairmove	Satisfactory	0.13	TooDraughty	0.28
rtsover	Uncomfortable	0.14	Uncomfortable	0.18
rtwhot	Satisfactory	0.09	TooCold	0.07
rwiaq	Satisfactory	0.13	Satisfactory	0.15
rwhumidity	TooDry	0.07	TooDry	0.06
rwventilation	Stuffy	0.05	Satisfactory	0.04
rwairmove	Still	0.14	Still	0.26
rtwover	Comfortable	0.03	Comfortable	0.03
rhealth	LessHealthy	0.07	Healthy	0.03
rcntnse	NoControl	0.13	NoControl	0.23
rcntco	NoControl	0.16	NoControl	0.44
rcntht	NoControl	0.16	NoControl	0.44
reffect	Unsatisfactory	0.13	Unsatisfactory	0.13
rspeed	TooSlow	0.22	TooSlow	0.20
rmeeting	Satisfactory	0.28	Satisfactory	0.15
rclean	Satisfactory	0.08	Satisfactory	0.01
rspace	Satisfactory	0.02	TooMuch	0.14
rstorage	Satisfactory	0.06	Outstanding	0.12
rdesign	Satisfactory	0.04	Satisfactory	0.01
rimage	Good	0.01	Good	0.01
rfneeds	Satisfactory	0.10	Satisfactory	0.05
rprod	NoDifference	0.05	MoreProductivity	0.13
rltart	Satisfactory	0.01	Satisfactory	0.04
rltnat	TooLittle	0.03	Satisfactory	0.06

where, the prefix '*r*' means 'the results of', so, *rsiaq/rwiaq* means the result of IAQ in summer/winter, *rtshot/stwhot* for temperature in summer/winter, *rshumidity*/*rwhumidity* for humidity in summer/winter, *rsairmove*/*rwairmove* for air movement in summer/winter, *rtsover/rtwover* for thermal comfort in summer/winter, *rsventilation/rwventilation* for ventilation in Summer/winter, *rhealth* for perceived health, *rcntnse* for control over noise, *rcntco/rcntht* for control over cooling/heating, *rfneeds* for facility meet needs, *reffect*/*rspeed* for response effectiveness/speed for request, *rmeeting for* the availability of meeting rooms, *rclean* for cleaning in buildings, *rspace* for space at work or desk, *rstorage* for storage arrangements, *rdesign* for building overall design, *rimage* for image to visitors, *rprod* for productivity at work, *rltart/rltnat for* amount of artificial light/natural light.

After performing the query, the RIR results of each building performance indicator are given, depending on the value of these RIR indicators, the importance of building optimization is identified and ranked, and corresponding action suggestions can be put forward. For the building performance with unsatisfactory results, the bigger the RIR value is, the poorer the performance is, that is, the larger the performance gap is between the evaluated building and benchmark buildings. In this case study, for building Eastern Gateway, it has poor building performances in the aspects of indoor air quality (0.22), speed to response requests (0.22), the temperature in summer (0.17) and the control over cooling (0.16) or heating (0.16), thermal comfort in summer (0.14) and winter air movement (0.14), etc.; but for building Michael Sterling, it has poor performances in the aspects of control ability over cooling (0.44) or heating (0.44), the temperature in summer (0.34), air movement in summer (0.28), indoor air quality in summer (0.27), air movement in summer (0.26), speed to response requests (0.2) and summer humidity (0.2), etc. The results from Table 5-14 also indicate that the overall performance of the Michael Sterling building in summer conditions is worse than the Eastern Gateway building. Therefore, this RIR indicator can not only be used to present the performance gap between the evaluated buildings and benchmark buildings, but also be used to indicate the building performance gap between the peer example buildings,

In addition, based upon the results, the corresponding suggestion actions can be proposed and written into SWRL rules. In this example, the pre-defined SWRL rules for the actions in winter are shown in Table 5-15.

Table 5-15 SWRL rules for actions

Rule_Action Humidity TooDry

EvaluatedNonDomesticBuilding(?endb) ^ hasResultWHUMIDITY(?endb, TooDry) -> hasActionWinterHumidity(?endb, AddHumidifier) ^ hasActionWinterHumidity(?endb, TurnUpHeatingSettingTemperature)
Rule_Action WIAQ Satisfactory
hasResultWIAQ(?endb, Satisfactory) ^ EvaluatedNonDomesticBuilding(?endb) -> hasActionWinterIAQ(?endb, NoAction)
Rule_Action TWHOT TooHot
hasResultTWHOT(?endb, TooHot) ^ EvaluatedNonDomesticBuilding(?endb) -> hasActionWinterTemperature(?endb, TurnDownHeatingSettingTemperature)

The query rules in Table 5-16 are used to output the information in the aspects of the heating system set value of temperature in winter (svwt), the evaluation result of winter temperature, the suggested actions for evaluated buildings, and the systems setting value for humidity in winter (svwh), the result of assessed winter humidity, as well as the suggested actions for improving the building occupants' comfort on the humidity in winter.

Table 5-16 Query rules for actions

Query_Actions In Winter
EvaluatedNonDomesticBuilding(?endb) ^ hasSetValueWinterTemperature(?endb, ?svwt) ^ hasResultTWHOT(?endb, ?rtwhot) ^ hasResultWHUMIDITY(?endb, ?rhumidity) ^ hasSetValueWinterHumidity(?endb,?svwh)^hasActionWinterTemperature(?endb, ?whotaction) ^ hasActionWinterHumidity(?endb, ?whumidityaction) -> sqwrl:select(?endb, ?svwt, ?rtwhot, ?whotaction, ?svwh, ?rhumidity, ?whumidityaction)

After executing the query in Table 5-16, the outputs are shown in Figure 5-12. For example, if the evaluated building has a satisfactory result with room temperature in winter at 20 degrees, then there is no need to take any actions for this building. But if the assessment result is too cold with the setting temperature at 18 degrees, then the facility managers need to take the action of adjusting the heating system setting temperature, such as turning up the heating system setting value. The result of humidity in winter in both two sample buildings is too dry, so the occupants or facility managers are suggested to add extra humidifiers or introduce plants to offices to improve the occupants' comfort with the humidity in winter. The suggested actions can be self-defined in this model.

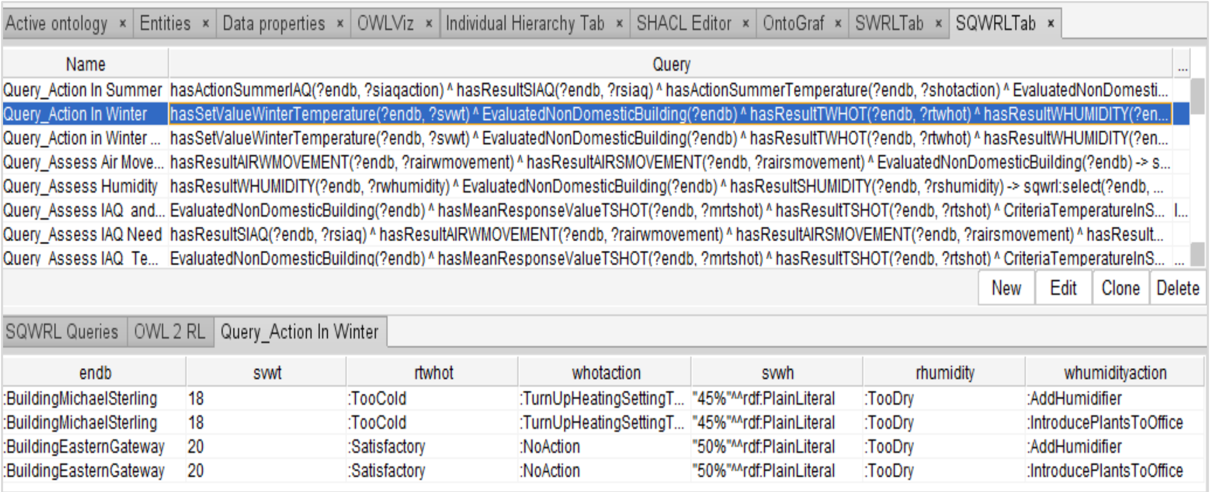


Figure 5-12 Query result for suggested actions

In addition, the Table 5-17 query is used to provide the system controller information, the location, the ID, etc. Through this query, facility managers can easily locate operation devices or the system control room. The output result of the above query is given in Figure 5-13.

Table 5-17 Building system information

Query_Building System Information
EvaluatedNonDomesticBuilding(?endb)^hasSetValueWinterTemperature(?endb,?svwt) ^ hasResultTWHOT(?endb,?rtwhot)^hasSystemHeating(?endb,?system)^locatedAt(?system, ?location)^hasID(?system,?id)->sqwrl:select(?endb, ?svwt, ?rtwhot, ?system, ?id, ?location)

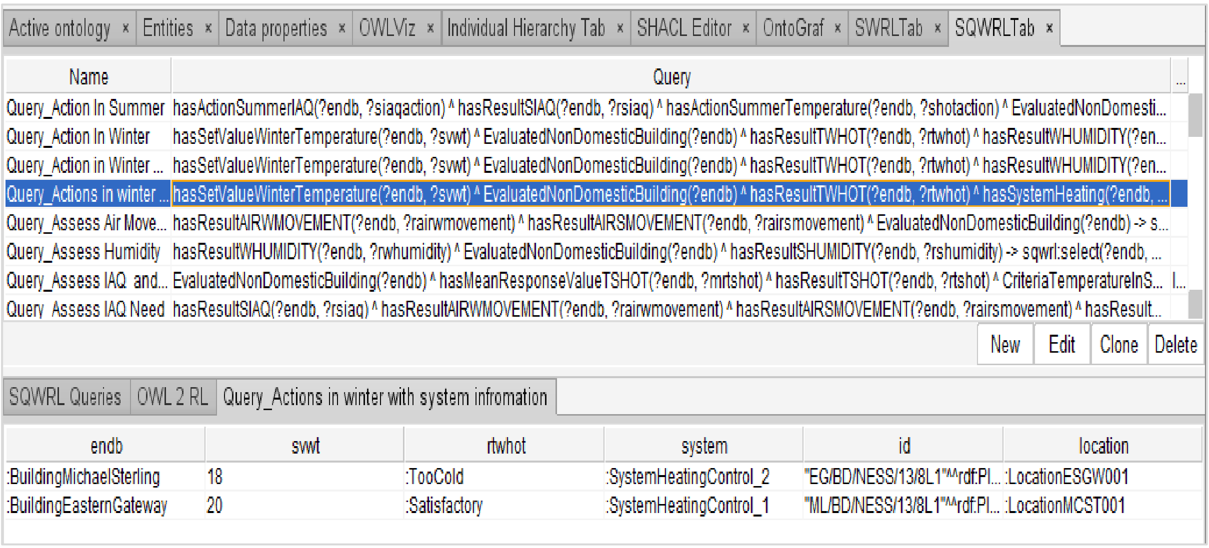


Figure 5-13 Building system information query results

This POE ontology can not only be used to conduct a comprehensive multi-objective and multi-criteria post-occupancy evaluation with a focus on occupant satisfaction with building performances, but also be used to systematically describe building information.

Figure 5-14 provides an overview of how the application feasibilities of this framework can be achieved in different scenarios. There are three layers of the ontology system, the SWRL rules layer, the SQWRL query layer, and the query results layer. The SWRL rules layer defines the internal constraints rules of the POE assessment; the SQWRL rules layer is a user-needs driven query rule defining layer, where the users can define the query rules according to the information they wanted to query in this model; the results layer provides the user-driven query results, which can return single query result for simple query task, or return integrated query results for multiple query tasks. The successful implementation of this model in different scenarios has proved the validity and feasibility of this POE ontology framework and the value of this research.

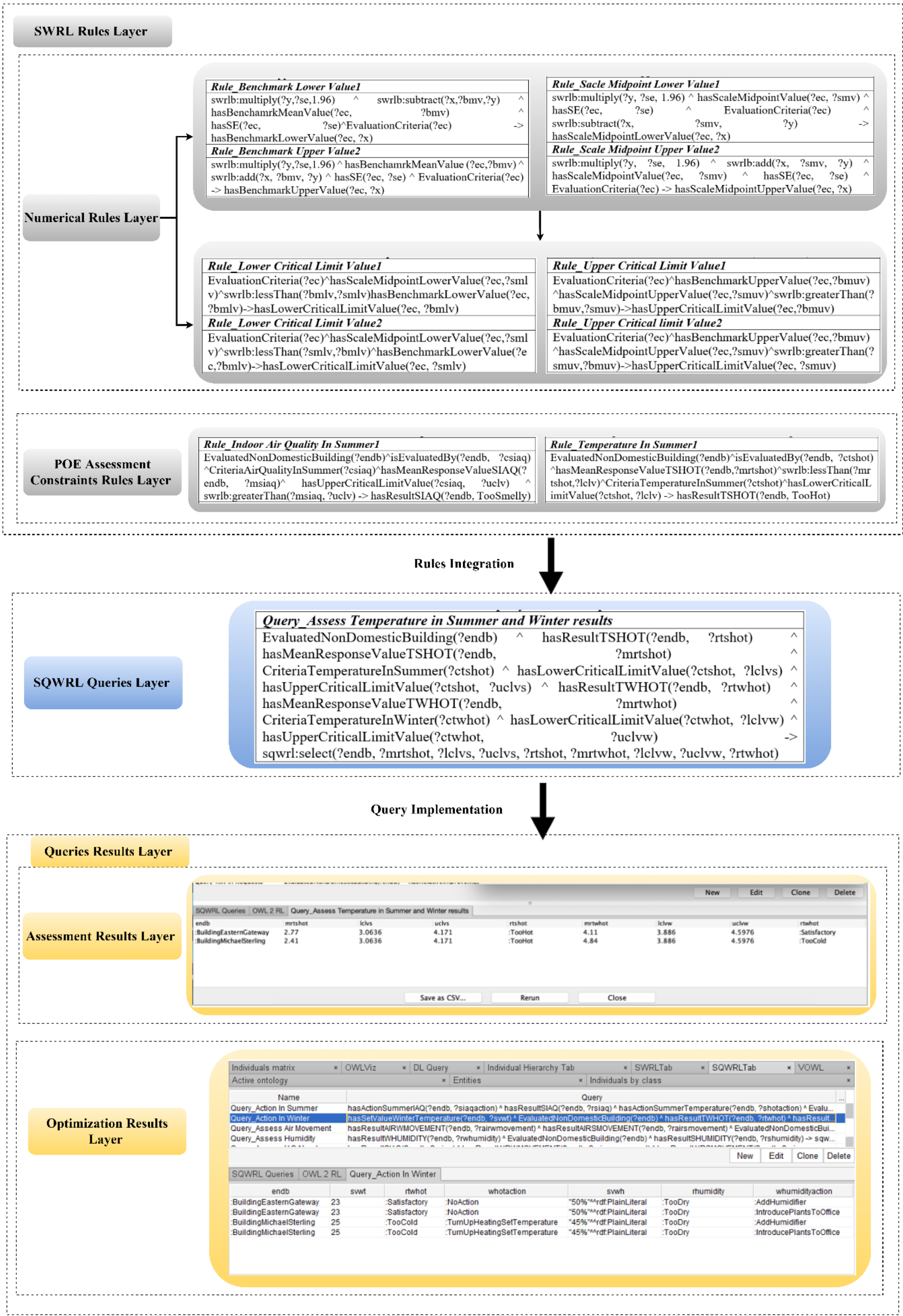


Figure 5-14 A multi-objective and multi-criteria holistic POE assessment process

5.4 Summary

In this chapter, based on the combination of pre-defined POE conditions SWRL rules and the users' needs-driven SQWRL queries, a case study of POE assessment is carried out in the POE ontology framework to validate the feasibility and effectiveness of the system. At first, the background information of case study sample buildings and the introduction of BUS Methodology are given. As illustrated in Figure 5-14, the SWRL rules for calculating the assessment benchmark thresholds are defined first in the SWRL rules layer, as well as the POE assessment conditions rules. In the SQWRL query layer, based upon the predefined SWRL rules and a holistic POE assessment with the focus of occupant satisfaction on building performances method that includes multiple objectives and multiple assessment criteria are built by using SQWRL queries, to illustrate the assessment results with identifying the building performance gap between evaluated buildings and benchmark buildings. In addition, according to the output results after the previous evaluation queries implementation, the satisfaction levels of building occupants' needs for all aspects of building performances are generated, and the importance of building performance optimization is identified and ranked based on the relative improvement ratio (RIR) value. Lastly, an example of application scenarios in operation optimization is given.

Except for the examples illustrated in this research, users can define additional SWRL rules and SQWRL query rules in the proposed ontology depending on their usage requirements and application scenarios, to enrich the OWL ontology. The successful implementation of this system in different application scenarios shows that the proposed POEOntology has the capability to conduct the multi-objective and multi-criteria post-occupancy evaluation of buildings.

Chapter 6 Conclusions and future work

This chapter presents a review and summary of the aim and contributions of this research. The limitations of this work are discussed, as well as the future work.

6.1 Research problems and aim review

Starting with a brief introduction of the research background, research problems are investigated in the first part of this work, followed by research motivations and contributions to knowledge. The development of human-centred sustainable building has become a global trend, which aims to provide building occupants with an energy-efficient and healthy built environment. However, for many reasons, buildings fail to meet the design requirements and building occupants' needs. As the AEC industry is a knowledge-intensive field, the failure of effective knowledge management and information exchange is recognized as one of the main causes of poor building performance. To overcome the failures, the Semantic Web technology has been introduced in many domains of the AEC industry, to facilitate effective knowledge management and information exchange.

After a broad critical literature review of ontology application in different fields of the AEC industry, this study identified that there is a lack of a comprehensive knowledge formalization system in the occupants-participated post-occupancy evaluation (POE) domain. Moreover, the ontology research in building performance evaluation mainly focuses on energy-efficiency-related fields, but not enough research on occupant satisfaction related fields. Most of the existing building evaluation related ontologies are lightweight ontologies that mainly focus on building a structured system to represent the specific domain knowledge or information, without developing formal axioms and constraints to provide higher expressivity and functionality.

Based on the identified problems, this study developed a heavyweight ontology dedicated to post-occupancy evaluation (POE) with a focus on occupant satisfaction evaluation in terms of building performance, with the ultimate aim of optimizing building operation and improving occupant use experience quality and well-being. The outcomes of this research also provide benchmark guidelines for future building management systems maintenance.

6.2 Research findings

A state-of-the-art literature review is carried out on the aspects of post-occupancy evaluation (POE) and ontology development.

- **Findings from literature review of post-occupancy evaluation (POE)**

Over the decades, the rise of emerging technologies and building evaluation schemes have produced a large amount of knowledge and information, the scattered and fragmented knowledge has made it time-consuming and error-prone to acquire explicit knowledge and information in the building management field.

Moreover, there is no one-size-fits-all POE, the POE assessment targets vary from building to building. Therefore, the current AEC industry is facing the challenges of choosing the appropriate assessment indicators and techniques among the vast and scattered knowledge, that has hindered the POE development.

- **Findings from literature review of Semantic Web and ontology**

The limitations of the current Web are identified, including the Web content is not machine-accessible and it provides an overwhelmed of heterogeneous knowledge sources in different formats. As the key technology of the Semantic Web, ontology technology provides a vocabulary of terms and relations with which to model the domain and plays a key role in analysing, modelling and processing the domain knowledge. In addition, it represents knowledge in a machine-readable format and achieves the interaction not only between different machines but also between machines and people.

Because of its capabilities of achieving knowledge formalization, sharing, reusing, reasoning and retrieval, ontology has been widely applied in knowledge management engineering, artificial intelligence, computer science, etc. The application ranges of ontology technology in the AEC industry include compliance checking domain, safety and risk management, building system management, building performance evaluation, human behaviours analysis in buildings, etc. The ontology research in the AEC industry mainly focuses on energy-efficiency-related fields, but not enough research on occupant satisfaction related fields. There is not an occupant satisfaction-centric POE ontology yet, which is the starting point of this research. In addition, the existing ontologies related to analyzing occupant behaviours or activities in buildings are mainly lightweight ontologies that focus on building a structured representation of the domain-specific knowledge or information, without delving into rule-based reasoning and query functions to provide higher expressivity.

6.3 Methodology review

In Chapter 3, Figure 3-1 presents an overview of the developed methodologies and the corresponding development processes in this research.

At first, a state-of-the-art literature review on POE and ontology is given, based on the findings from the literature review, the research aim of developing a post-occupancy evaluation ontology framework with the focus on building occupants' satisfaction evaluation on building performances is proposed.

Then, the heavyweight POE OWL ontology has been established to capture the fragmented knowledge of building assessment in the POE domain. The '7-Steps' methodology from Noy and McGuinness (2001) is used to develop the POE ontology knowledge base. In addition, the practical guides from Horridge (2011) and DeBellis (2021) for building OWL ontologies by using ontology editor Protégé have been followed, to build the proposed POE ontology from scratch within the software Protégé 5.5.0 and a set of plug-ins. The outcome of this process is a formalized knowledge base, the building occupants' needs for building performance are generalized and classified, and a list of object properties, data properties and instances are defined to represent the relationships and attributes between concepts in POE ontology.

Based on the asserted axioms in the OWL knowledge base developed above and the assessment criteria constraints rules in BUS methodology, the SWRL rules for determining the evaluation constraints conditions in POE assessment are defined. This has enabled the inference function of ontology to allow the new inferred facts to be generated after the execution of SWRL rules in the rule engine, and then new facts can be returned to the knowledge base as the new facts of the OWL items, this has determined the assessment parameter thresholds for building performance. After determining the POE constraints SWRL rules, the user-need-driven SQWRL query rules are defined according to the information users wanted to query in the ontology.

A complete heavyweight POE ontology has been established. To validate the proposed ontology, the Pellet reasoner is used to carry out a consistency checking and classification of ontology, and to infer implicit new axioms of ontology from the given explicit facts in the knowledge base. Also, the Drools rule engine can be used here to infer new knowledge and transfer the inferred rule engine knowledge to the OWL knowledge, and write back to the ontology model.

As structured in Figure 3-6, the POE ontology framework development in detail, including the POE knowledge acquisition and formalization, SWRL rules and SQWRL query definition, as well as the used technologies and tools, etc.

In the end, based on the combination of pre-defined POE constraints conditions SWRL rules and the user-need-driven SQWRL queries, a case study of POE assessment is carried out in the POEOntology framework to illustrate the feasibility and effectiveness of the ontology framework in different application scenarios. The successful implementation of this system in different application scenarios shows that the proposed POE ontology has the capability to conduct the multi-objective and multi-criteria post-occupancy evaluation of buildings. And this ontology model also enables effective POE-related knowledge retrieving and sharing, and promotes its implementation in the POE domain.

Except for the examples illustrated in this research, users can define additional SWRL rules and SQWRL query rules in the proposed ontology depending on their usage requirements and application scenarios, to enrich the OWL ontology. The successful implementation of this model in different scenarios has proved the validity and feasibility of this POE ontology framework and the value of this research.

6.4 Contribution to the knowledge

This study has conducted a research on the development of an ontology-based post-occupancy evaluation framework, the contributions to knowledge can be summarized as follows:

- This research conducts a state-of-the-art review on the development of post-occupancy evaluation and Semantic Web ontologies, especially the ontology application in the AEC industry, and provides valuable findings.

- On the one hand, this research reuses and extends existing ontologies to systematically present building occupant’s needs inside buildings; on the other hand, the developed ontology in this research is also reusable to provide semantic resources for future relevant research in the building construction industry that achieves knowledge reusing and sharing.
- Developed an enhanced occupants-participated POE assessment knowledge formalization system. In the POE ontology, the building occupants’ needs for building performance are generalized and classified, and the corresponded building performance assessment knowledge is formalized. The proposed heavyweight ontology model realizes the structural representation, sharing, and reuse of fragmented knowledge and heterogeneous data in the building assessment domain.
- This research develops a set of SWRL rules and SQWRL query rules in the proposed ontology framework, which helps to conduct a Semantic Web-based automatic multi-objective and multi-criteria POE assessment of buildings. And this ontology model also enables effective POE-related knowledge retrieving and sharing, and promotes its integration within the building life cycle assessment.
- A ‘one-size fits all POE’ does not exist, therefore this proposed POE framework is tailored to specific building applications with the emphasis on occupant satisfaction in the built environment. A user-centric comprehensive POE framework is developed by reusing the existing building occupants' related ontologies and schemes into this POE to realize the interoperability and reusing of different schemes.
- This research also presents a real use case study that demonstrates how the proposed ontology can be used to infer implicit assessment knowledge to facilitate building performance assessment in use, with the ultimate aim of optimizing building operation guidelines and improving occupants’ use experience quality and well-being.

6.5 Research limitations and future work

This research undoubtedly is limited by the scope of research and time constraints. To some extent, this research has been affected by the current pandemic.

In this research, the identified research limitations and the corresponding suggested future work are stated as follows:

- This research sampled two buildings with over 150 participants, the sample size is deep but narrow. Future research would broaden the sample base. The building occupants research conducted in this research, only sampled one point time, future research will benefit from understanding changes over time, i.e. the time series to report and analyse variances.
- The current pandemic has fundamentally changed people's needs and behaviours inside buildings, future research could consider understanding how the pandemic has changed people's needs and perceptions.
- This research case study was conducted in non-domestic office buildings, future research could consider other building types, for example, residential buildings, industrial buildings, etc. And future work also should consider developing different benchmark datasets for specific building types.
- This research is developed for building performance assessments with a focus on building occupants' satisfaction with subjective measures. Future research could expand the scope of this study and consider covering the assessment in real-time building consumption and physical measurement data on building performances in use.
- As analysed in the research, there are many existing ontologies application examples in the AEC industry, future research should consider the extensibility and interoperability between the proposed ontology and other ontological systems.
- This developed ontology has reused some needs concepts from existing ontologies, and also developed some new needs concepts, but it is not exhaustive and perfect. Future work could consider more detailed requirements.

REFERENCES

Abadi, A., Ben-Azza, H. and Sekkat, S. (2018) ‘Improving integrated product design using SWRL rules expression and ontology-based reasoning’, *Procedia Computer Science*. Elsevier B.V., 127, pp. 416–425. doi: 10.1016/j.procs.2018.01.139.

Abanda, F. H. and Tah, J. H. M. (2008) ‘Towards developing a sustainable building technology ontology’, in *In: Dainty, A (Ed) Procs 24th Annual ARCOM Conference*. Cardiff, UK, pp. 1–3.

Abdelghany, A. S., Darwish, N. R. and Hefni, H. A. (2019) ‘An agile methodology for ontology development’, *International Journal of Intelligent Engineering and Systems*, 12(2), pp. 170–181. doi: 10.22266/IJIES2019.0430.17.

Abdou, A. and Dghaimat, M. Al (2016) ‘Post Occupancy Evaluation of Educational Buildings: A Case Study of a Private School in the UAE’, in *4th Annual International Conference on Architecture and Civil Engineering (ACE 2016)*, pp. 93–100. doi: 10.5176/2301-394X_ACE16.102.

Alatrish, E. (2013) ‘Comparison Some of Ontology Editors’, *Management Information Systems*, 8(2), pp. 018–024.

Alborz, N. and Berardi, U. (2015) ‘A post occupancy evaluation framework for LEED certified U.S. higher education residence halls’, *Procedia Engineering*. Elsevier B.V., 118, pp. 19–27. doi: 10.1016/j.proeng.2015.08.399.

Alborzfard, N. and Berardi, U. (2013) ‘Selection of Indicators for Post Occupancy Evaluation (POE) of Sustainable Dormitories’, in *SB13 Dubai-067*, pp. 1–8. Available at: http://www.irbnet.de/daten/iconda/CIB_DC26865.pdf (Accessed: 6 September 2018).

Andryani, R., Negara, E. S. and Darma, U. B. (2015) ‘Survey on Development Method of Ontology’, in *The 4th ICIBA 2015, International Conference*, pp. 20–21.

Antoniou, G. and Harmelen, F. van (2008) *A Semantic Web Prime*. 2nd edn, *The MIT Press*. 2nd edn. The MIT Press. Available at: https://www.cambridge.org/core/product/identifier/S0269888909990117/type/journal_article.

Arslan, M., Cruz, C. and Ginhac, D. (2019) ‘Understanding Occupant Behaviors in Dynamic Environments using OBiDE framework’, *Building and Environment*. Elsevier, 166(September),

p. 106412. doi: 10.1016/j.buildenv.2019.106412.

Awadh, O. (2017) ‘Sustainability and green building rating systems: LEED, BREEAM, GSAS and Estidama critical analysis’, *Journal of Building Engineering*. Elsevier Ltd, 11(August 2016), pp. 25–29. doi: 10.1016/j.jobe.2017.03.010.

Balaji, B. *et al.* (2016) ‘Brick: Towards a unified metadata schema for buildings’, in *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments*, pp. 41–50.

Balaji, B. *et al.* (2018) ‘Brick: Metadata schema for portable smart building applications’, *Applied Energy*. Elsevier, 226(February), pp. 1273–1292. doi: 10.1016/j.apenergy.2018.02.091.

Bautista-Zambrana, M. R. (2015) ‘Methodologies to Build Ontologies for Terminological Purposes’, *Procedia - Social and Behavioral Sciences*. Elsevier B.V., 173, pp. 264–269. doi: 10.1016/j.sbspro.2015.02.063.

Beach, T. H. *et al.* (2015) ‘A rule-based semantic approach for automated regulatory compliance in the construction sector’, *Expert Systems with Applications*, 42(12), pp. 5219–5231. doi: 10.1016/j.eswa.2015.02.029.

Beetz, J., van Leeuwen, J. and De Vries, B. (2009) ‘IfcOWL: A case of transforming EXPRESS schemas into ontologies’, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, 23(1), pp. 89–101. doi: 10.1017/S0890060409000122.

Behm, M. and Pearce, A. (2017) *Occupational Safety and Health in Green Buildings: LEED PtD Pilot Credit Analysis*.

Bernardi, E. *et al.* (2017) ‘An Analysis of the Most Adopted Rating Systems for Assessing the Environmental Impact of Buildings’, *Sustainability*, 9(7), p. 1226. doi: 10.3390/su9071226.

Berners-Lee, Ti., Hendler, J. and Lassila, O. (2001) ‘The semantic web: a brain for humankind’, *Scientific American*, 284(5), pp. 34–43. doi: 10.1109/MIS.2001.920595.

Bikakis, N. *et al.* (2013) *The XML and semantic web worlds: Technologies, interoperability and integration: A survey of the state of the art*, *Studies in Computational Intelligence*. doi: 10.1007/978-3-642-28977-4_12.

Bilgin, G., Dikmen, I. and Birgonul, M. T. (2018) ‘An ontology-based approach for delay analysis in construction', *KSCE Journal of Civil Engineering*, 22(2), pp. 384–398. doi: 10.1007/s12205-017-0651-5.

Bonino, D. and Corno, F. (2008) 'DogOnt - Ontology Modeling for Intelligent Domotic Environments', in Sheth, A. et al. (eds) *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science), pp. 790–803. doi: 10.1007/978-3-540-88564-1_51.

Bordass, B. *et al.* (2001a) 'Assessing building performance in use 2: Technical performance of the Probe buildings', *Building Research and Information*, 29(2), pp. 103–113. doi: 10.1080/09613210010008027.

Bordass, B. *et al.* (2001b) 'Assessing building performance in use 3: Energy performance of the Probe buildings', *Building Research and Information*, 29(2), pp. 114–128. doi: 10.1080/09613210010008036.

Bordass, B. and Leaman, A. (2015) 'Building performance evaluation in the UK: So many false dawns', *Architecture Beyond Criticism: Expert judgement and performance evaluation*, pp. 160–170.

Bordass, B., Leaman, A. and Ruyssevelt, P. (2001) 'Assessing building performance in use 5: Conclusions and implications', *Building Research and Information*, 29(2), pp. 144–157. doi: 10.1080/09613210010008054.

Bosak, J. and Bray, T. (1999) 'XML and the second-generation Web', *Scientific American*, 280(5), pp. 89–93. doi: 10.1038/scientificamerican0599-89.

Bouzidi, K. R. *et al.* (2011) 'An Ontological Approach for Modeling Technical Standards for Compliance Checking', in *The Fifth International Conference on Web Reasoning and Rule Systems*, pp. 244–249. doi: 10.1007/978-3-642-23580-1_19.

Brandt, S. C. *et al.* (2008) 'An ontology-based approach to knowledge management in design processes', *Computers and Chemical Engineering*, 32(1–2), pp. 320–342. doi: 10.1016/j.compchemeng.2007.04.013.

Bravo, M., Reyes, L. F. H. and Reyes Ortiz, J. A. (2019) 'Methodology for ontology design and construction', *Contaduria y Administracion*, 64(4), pp. 1–24. doi: 10.22201/FCA.24488410E.2020.2368.

BRE Gloabl (2018) *BREEAM UK New Construction 2018 Consultation Draft*. Available at: http://files.bregroup.com/breeam/consultations/SD5078_DRAFT-UK_nondom_NC_2018-manual.pdf.

BRE Global Ltd (2018) *Breeam UK New construction: Non-domestic Buildings (United Kingdom) Technical manual SD5078:BREEAM UK New Construction 2018, BRE Global Ltd*. United Kingdom. doi: 10.1007/s13398-014-0173-7.2.

Brioso, X. *et al.* (2015) 'Using post-occupancy evaluation of housing projects to generate value for municipal governments', *Alexandria Engineering Journal*. Faculty of Engineering, Alexandria University, pp. 1–12. doi: 10.1016/j.aej.2017.01.015.

Brown, Z. *et al.* (2010) 'Evaluating user experience in green buildings in relation to workplace culture and context', *Facilities*, 28(3/4), pp. 225–238. doi: 10.1108/EL-01-2017-0019.

BSI (2015) 'BS 8536-1:2015 Briefing for design and construction - Part 1 : Code of practice for facilities management (Buildings infrastructure)', *BSI Standards Publication*. Published by BSI Standards Limited, pp. 1–96.

BUS METHODOLOGY (2017) *BUS Methodology*. Available at: https://busmethodology.org.uk/index.html (Accessed: 1 October 2021).

Carthey, J. (2006) 'Post Occupancy Evaluation: Development of a Standardised Methodology for Australian Health Projects', *International Journal of Construction Management*, 6(1), pp. 57–74. doi: 10.1080/15623599.2006.10773082.

Catenazzi, N., Sommaruga, L. and Mazza, R. (2009) 'User-Friendly Ontology Editing and Visualization Tools: The OWLeasyViz Approach', in *2009 13th International Conference Information Visualisation*. IEEE, pp. 283–288. doi: 10.1109/IV.2009.34.

Chávez-Feria, S. *et al.* (2020) 'From obXML to the OP ontology: Developing a semantic model for occupancy profile', *CEUR Workshop Proceedings*, 2636(820621), pp. 9–22.

Chávez-Feria, S., Poveda-Villalón, M. and García-Castro, R. (2020) *Occupancy Profile ontology*. Available at: https://bimerr.iot.linkeddata.es/def/occupancy-profile/.

Chen, X. *et al.* (2013) 'An ontology learning system for customer needs representation in product development', *International Journal of Advanced Manufacturing Technology*, 67(1–4), pp. 441–453. doi: 10.1007/s00170-012-4496-2.

Chen, X., Chen, C. and Leong Kah, F. (2011) 'Automated ontology-based customer needs translation and representation', *Emergency Management and Management Sciences (ICEMMS), 2011 2nd IEEE International Conference on*, pp. 907–910. doi: 10.1109/ICEMMS.2011.6015830.

Cohen, Robert *et al.* (2001) 'Assessing building performance in use 1 : the Probe process Assessing building performance in use 1 : the Probe process', *Building Research & Information*, 29(2), pp. 85–102. doi: 10.1080/09613210010008018.

Cohen, R. *et al.* (2001) 'Assessing building performance in use 1: The Probe process', *Building Research and Information*, 29(2), pp. 85–102. doi: 10.1080/09613210010008018.

Compton, M. *et al.* (2012) 'The SSN ontology of the W3C semantic sensor network incubator group', *Journal of Web Semantics*. Elsevier B.V., 17, pp. 25–32. doi: 10.1016/j.websem.2012.05.003.

Corcho, O. *et al.* (2005) 'Building Legal Ontologies with METHONTOLOGY and WebODE', in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer, Berlin, Heidelberg, pp. 142–157. doi: 10.1007/978-3-540-32253-5_9.

Corcho, O., Fernández-López, M. and Gómez-Pérez, A. (2003) 'Methodologies, tools and languages for building ontologies. Where is their meeting point?', *Data and Knowledge Engineering*, 46(1), pp. 41–64. doi: 10.1016/S0169-023X(02)00195-7.

Corry, E. *et al.* (2014) 'Using semantic web technologies to access soft AEC data', *Advanced Engineering Informatics*, 28(4), pp. 370–380. doi: 10.1016/j.aei.2014.05.002.

Corry, E. *et al.* (2015) 'A performance assessment ontology for the environmental and energy management of buildings', *Automation in Construction*. Elsevier B.V., 57, pp. 249–259. doi: 10.1016/j.autcon.2015.05.002.

Daniele, L., den Hartog, F. and Roes, J. (2015) 'Created in Close Interaction with the Industry: The Smart Appliances REFerence (SAREF) Ontology', *Lecture Notes in Business Information Processing*, 225(June 2016), pp. 100–112. doi: 10.1007/978-3-319-21545-7_9.

Darko, A. and Chan, A. P. C. (2016) 'Critical analysis of green building research trend in construction journals', *Habitat International*. Elsevier Ltd, 57, pp. 53–63. doi: 10.1016/j.habitatint.2016.07.001.

DARKWA, I. (2006) ‘Post-Occupancy Evaluation of State-Subsidised Housing Units in Kayamandi, Stellenbosch’, (March), pp. 1–160.

Darlington, M. J. and Culley, S. J. (2008) ‘Investigating ontology development for engineering design support’, *Advanced Engineering Informatics*, 22(1), pp. 112–134. doi: 10.1016/j.aei.2007.04.001.

David Jiboye, A. (2012) ‘Post-occupancy evaluation of residential satisfaction in Lagos, Nigeria: Feedback for residential improvement’, *Frontiers of Architectural Research*. Elsevier, 1(3), pp. 236–243. doi: 10.1016/j.foar.2012.08.001.

DeBellis, M. (2020) *Drools vs. Pellet for SWRL rules*. Available at: https://www.michaeldebellis.com/post/drools-vs-pellet-for-swrl-rules (Accessed: 30 September 2021).

DeBellis, M. (2021) *A Practical Guide to Building OWL Ontologies Using Protégé 5.5 and Plugins Edition 3.0*.

Degha, H. E. *et al.* (2018) ‘Onto-SB: Human Profile Ontology for Energy Efficiency in Smart Building’, in *Proceedings - PAIS 2018: International Conference on Pattern Analysis and Intelligent Systems*. IEEE, pp. 1–8. doi: 10.1109/PAIS.2018.8598509.

Delzendeh, E. *et al.* (2017) ‘The impact of occupants’ behaviours on building energy analysis: A research review’, *Renewable and Sustainable Energy Reviews*. Elsevier Ltd, 80(May), pp. 1061–1071. doi: 10.1016/j.rser.2017.05.264.

Dibley, M. *et al.* (2012) ‘An ontology framework for intelligent sensor-based building monitoring’, *Automation in Construction*. Elsevier B.V., 28, pp. 1–14. doi: 10.1016/j.autcon.2012.05.018.

Ding, L. Y. *et al.* (2016) ‘Construction risk knowledge management in BIM using ontology and semantic web technology’, *Safety Science*, 87, pp. 202–213. doi: 10.1016/j.ssci.2016.04.008.

Ding, Z. *et al.* (2018) ‘Green building evaluation system implementation’, *Building and Environment*. Elsevier, 133(February), pp. 32–40. doi: 10.1016/j.buildenv.2018.02.012.

Disterheft, A. *et al.* (2015) ‘Sustainable universities - A study of critical success factors for participatory approaches’, *Journal of Cleaner Production*. Elsevier Ltd, 106, pp. 11–21. doi: 10.1016/j.jclepro.2014.01.030.

Doan, D. T. *et al.* (2017) 'A critical comparison of green building rating systems', *Building and Environment*. Elsevier Ltd, 123(July 2018), pp. 243–260. doi: 10.1016/j.buildenv.2017.07.007.

van Dronkelaar, C. *et al.* (2016) 'A Review of the Regulatory Energy Performance Gap and Its Underlying Causes in Non-domestic Buildings', *Frontiers in Mechanical Engineering*, 1(January), pp. 1–14. doi: 10.3389/fmech.2015.00017.

DuCharme, B. (2013) *Learning SPARQL : Querying and Updating with SPARQL 1.1*. Second. O'Reilly Media, Inc.

Dundar, M. (2016) 'POST-OCCUPANCY EVALUATIONS IN ARCHITECTURAL PROGRAMMING PROCESS : A CASE STUDY OF', in *Archi-Cultural Interactions through the Silk Road 4th International Conference, Mukogawa Women's Univ.*, pp. 138–141.

Durosaiye, I. O., Hadjri, K. and Liyanage, C. L. (2019) 'A critique of post-occupancy evaluation in the UK', *Journal of Housing and the Built Environment*. Springer Netherlands, 34(1), pp. 345–352. doi: 10.1007/s10901-019-09646-2.

Edirisinghe, R. and Woo, J. (2021) 'BIM-based performance monitoring for smart building management', *Facilities*, 39(1–2), pp. 19–35. doi: 10.1108/F-11-2019-0120.

ETSI (2019a) *Extension to SAREF ;*

ETSI (2019b) *Extension to SAREF ; Part 4 : Smart Cities Domain*. Available at: https://www.etsi.org/deliver/etsi_ts/103400_103499/10341004/01.01.01_60/ts_10341004v010101p.pdf.

ETSI (2019c) *Extension to SAREF ; Part 5 : Industry and Manufacturing Domains*.

ETSI (2020) *SAREF Development Framework and Workflow, Streamlining the Development of SAREF and its Extensions*. doi: 10.3233/ssw200034.

Fekry, A. A., Zafarany, A. M. El and Shamseldin, A. K. M. (2014) 'Develop an environmental assessment technique for human comfort requirements in buildings', *HBRC Journal*. Housing and Building National Research Center, 10(1), pp. 1–9. doi: 10.1016/j.hbrcj.2013.05.013.

Fernández-López, M. *et al.* (1999) 'Building a chemical ontology using methontology and the ontology design environment', *IEEE Intelligent Systems and Their Applications*, 14(1), pp. 37–46. doi: 10.1109/5254.747904.

Fernández-López, M. (1999) ‘Overview Of Methodologies For Building Ontologies’, in *Proceedings of the IJCAI99 Workshop on Ontologies and ProblemSolving Methods Lessons Learned and Future Trends CEUR Publications*, pp. 1–13. doi: 10.1.1.39.6002.

Fernández-López, M. and Gómez-Pérez, A. (2002) ‘Overview and analysis of methodologies for building ontologies’, *Knowledge Engineering Review*, 17(2), pp. 129–156. doi: 10.1017/S0269888902000462.

Fernández-López, M., Gómez-Pérez, A. and Juristo, N. (1997) ‘METHONTOLOGY: From Ontological Art Towards Ontological Engineering’, in *Proceedings of the Ontological Engineering AAAI-97 Spring Symposium Series*, pp. 33–40.

Fronczek-Munter, A. (2013) ‘Evaluation methods for hospital facilities’, *International Journal of Facilities Management*, 12, pp. 215–226. Available at: http://orbit.dtu.dk/files/59511230/Evaluation_methods_for_hospital_facilities.pdf.

Fürst, F. and Trichet, F. (2006) ‘Heavyweight Ontology Engineering’, in *OTM Confederated International Conferences "On the Move to Meaningful Internet Systems“*, pp. 38–39. doi: 10.1007/11915034_18.

Gangolells, M. and Casals, M. (2012) ‘An ontology-based approach for on-site integrated environmental and health and safety management’, *Revista Ingeniería de Construcción*, 27(3), pp. 103–127.

Gecen, A. N. (2017) *Ontology-Based Approach to Design and Analysis of Sustainable Manufacturing Systems and its Implementation and Application Prespective*.

Gene Ontology Consortium (2017) *Protege-tutorial Documentation*. Available at: https://go-protege-tutorial.readthedocs.io/en/latest/index.html.

Gilani, S., Quinn, C. and McArthur, J. J. (2020) ‘A review of ontologies within the domain of smart and ongoing commissioning’, *Building and Environment*. Elsevier Ltd, 182(July), p. 107099. doi: 10.1016/j.buildenv.2020.107099.

Gómez-Pérez, A., Fernández-López, M. and Corcho, O. (2004) *Ontological Engeenering with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. Springer London. doi: 10.4018/978-1-59904-045-5.ch003.

González, E. *et al.* (2021) ‘An approach based on the ifcOWL ontology to support indoor navigation’, *Egyptian Informatics Journal*. Faculty of Computers and Artificial Intelligence, Cairo University, 22(1), pp. 1–13. doi: 10.1016/j.eij.2020.02.008.

Gou, Z., Prasad, D. and Siu-Yu Lau, S. (2013) ‘Are green buildings more satisfactory and comfortable?’, *Habitat International*, 39, pp. 156–161. doi: 10.1016/j.habitatint.2012.12.007.

Grigoris, A. and Frank, van H. (2004) *Semantic Web Primer, Linked Data Tools*. Available at: http://www.linkeddatatools.com/semantic-web-basics Page.

Gruber, T. R. (1993) ‘A translation approach to portable ontology specifications’, *Knowledge Acquisition*, 5(2), pp. 199–220.

Gruber, T. R. (1995) ‘Toward principles for the design of onotologies used for knowledge sharing’, *Human-Computer Studies*, 43, pp. 907–928.

Grüninger, M. and Fox, M. S. (1995) ‘Methodology for the Design and Evaluation of Ontologies’, in *International Joint Conference on Artificial Inteligence (IJCAI95), Workshop on Basic Ontological Issues in Knowledge Sharing*, pp. 1–10. Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.44.8723.

Guinther, L., Carll-White, A. and Real, K. (2014) ‘One size does not fit all: A diagnostic post-occupancy evaluation model for an emergency department’, *Health Environments Research and Design Journal*, 7(3), pp. 15–37. doi: 10.1177/193758671400700303.

Haller, A. *et al.* (2017) *Semantic Sensor Network Ontology, Open Geospatial Consortium (OGC) & World Wide Web Consortium (W3C)*. Available at: https://www.w3.org/TR/vocab-ssn/ (Accessed: 8 October 2021).

Haron, N. and Khairudin, F. (2012) ‘The application of Quality Function Deployment (QFD) in the design phase of Industrialized Building System (IBS) apartment’, *European International Journal of Science and Technology*, 1(3), pp. 56–66. Available at: http://www.cekinfo.org.uk/images/frontImages/gallery/Vol1No3December2012/6.pdf.

Harris, S. and Seaborne, A. (2013) *SPARQL 1.1 Query Language W3C Recommendation, W3C*. Available at: https://www.w3.org/TR/sparql11-query/ (Accessed: 13 October 2021).

Hassin, M. and Azlani, S. N. H. B. (2018) ‘Post-Occupancy Evaluation for Green Building in Kuala Lumpur’, *International Journal of Academic Research in Business and Social Sciences*, 8(8). doi: 10.6007/ijarbss/v8-i8/4942.

Hay, R. *et al.* (2018) ‘Post-occupancy evaluation in architecture: experiences and perspectives from UK practice', *Building Research and Information*, 46(6), pp. 698–710. doi: 10.1080/09613218.2017.1314692.

Hermawati, A. *et al.* (2015) 'Framework model for post-occupancy evaluation of school facilities', *Structural Survey*, 33(4/5), pp. 332–336. doi: 10.1108/EL-01-2017-0019.

Hiromoto, J. (2015) *Architect & Design Sustainable Design Leaders Post Occupancy Evaluation Survey Report*.

Hitzler, P., Krötzsch, M. and Rudolph, S. (2010) *Foundations of Semantic Web Technologies*. Edited by J. Impagliazzo and V. McGettrick. Boca Raton: Chapman & Hall/CRC Press.

Hong, T., D'Oca, S., Turner, W. J. N., *et al.* (2015) 'An ontology to represent energy-related occupant behavior in buildings. Part I: Introduction to the DNAs framework', *Building and Environment*. Elsevier Ltd, 92, pp. 764–777. doi: 10.1016/j.buildenv.2015.02.019.

Hong, T., D'Oca, S., Taylor-Lange, S. C., *et al.* (2015) 'An ontology to represent energy-related occupant behavior in buildings. Part II: Implementation of the DNAS framework using an XML schema', *Building and Environment*. Elsevier Ltd, 94(P1), pp. 196–205. doi: 10.1016/j.buildenv.2015.08.006.

Hong, T. *et al.* (2018) 'Occupant behavior models: A critical review of implementation and representation approaches in building performance simulation programs', *Building Simulation*, 11(1), pp. 1–14. doi: 10.1007/s12273-017-0396-6.

Horridge, M. (2011) *A Practical Guide To Building OWL Ontologies Using Protégè 4 and CO-ODE Tools Edition 1.3*. Available at: http://owl.cs.manchester.ac.uk/tutorials/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_3.pdf.

Horrocks, I. *et al.* (2004) *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*, *W3C*. Available at: https://www.w3.org/Submission/SWRL/ (Accessed: 12 October 2021).

Hou, S. (2015) *An ontology-based holistic approach for multi-objective sustainable structural design*. Available at: http://ezproxy.leedsbeckett.ac.uk/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edsble&AN=edsble.685495&site=eds-live&scope=site.

Hou, S., Li, H. and Rezgui, Y. (2015) 'Ontology-based approach for structural design considering low embodied energy and carbon', *Energy and Buildings*. Elsevier B.V., 102(2015), pp. 75–90. doi: 10.1016/j.enbuild.2015.04.051.

Howsawi, A. and Zhang, J. (2017) 'An Ontology to Support the Move towards Sustainable Construction in Saudi Arabia', in *ASCE International Workshop on Computing in Civil Engineering 2017*, pp. 296–303. doi: 10.1061/9780784480847.037.

Hu, Y. Z. *et al.* (2013) 'The regulation constraint modeling and semantic inferring in construction quality checking', *Structural Survey*, 31(5), pp. 387–400. doi: 10.1108/SS-10-2012-0033.

ICE (2021) *Performance gap between building design and operation*. Available at: https://www.designingbuildings.co.uk/wiki/Performance_gap_between_building_design_and_operation (Accessed: 26 November 2021).

Ivanovic, D. (2021) *Vitro*, *GitHub*. Available at: https://github.com/vivo-project/Vitro (Accessed: 15 October 2021).

IWBL (2016) *THE WELL BUILDING STANDARD*.

Jacob, E. K. (2005) 'Ontologies and the Semantic Web', *Bulletin of the American Society for Information Science and Technology*, 29(4), pp. 19–22. doi: 10.1002/bult.283.

Janowicz, K. *et al.* (2019) 'SOSA: A lightweight ontology for sensors, observations, samples, and actuators', *Journal of Web Semantics*. Elsevier B.V., 56, pp. 1–10. doi: 10.1016/j.websem.2018.06.003.

Jiang, S., Wang, N. and Wu, J. (2018) 'Combining BIM and Ontology to Facilitate Intelligent Green Building Evaluation', *Journal of Computing in Civil Engineering*, 32(5), pp. 1–15. doi: 10.1061/(ASCE)CP.1943-5487.0000786.

Jin, R. *et al.* (2019) 'Integrating BIM with building performance analysis in project life-cycle', *Automation in Construction*, 106(October). doi: 10.1016/j.autcon.2019.102861.

Jones, D., Bench-Capon, T. and Visser, P. (2007) 'Methodologies for Ontology Development', in *Proc. IT&KNOWS Conference of the 15th IFIP World Computer Congress*.

Kalyanpur, A. *et al.* (2006) 'Swoop: A Web Ontology Editing Browser', *Web Semantics*, 4(2), pp. 144–153. doi: 10.1016/j.websem.2005.10.001.

Kapoor, B. and Sharma, S. (2010) 'A Comparative Study of Ontology building Tools in Semantic Web Applications', *International journal of Web & Semantic Technology*, 1(3), pp.

1–13. doi: 10.5121/ijwest.2010.1301.

Kawneer UK Limited (2014) *Case Study : Eastern Gateway Building , Brunel University*. Available at: https://www.kawneer.com/kawneer/united_kingdom/en/products/project_detail.asp?project=Brunel_University&bus=united_kingdom&pc=education.

Khair, N. *et al.* (2015) 'Post occupancy evaluation of physical environment in public low-cost housing', *Jurnal Teknologi*, 75(10), pp. 155–162. doi: 10.11113/jt.v75.5284.

Khoshbakht, M., Gou, Z., Lu, Y., *et al.* (2018) 'Are green buildings more satisfactory? A review of global evidence', *Habitat International*, 74, pp. 57–65. doi: 10.1016/j.habitatint.2018.02.005.

Khoshbakht, M., Gou, Z., Xie, X., *et al.* (2018) 'Green Building Occupant Satisfaction: Evidence from the Australian Higher Education Sector', *Sustainability*, 10(8). doi: 10.3390/su10082890.

Kishnani, N. *et al.* (2014) 'Post-Occupancy Evaluation of Green Mark-Rated Buildings in Singapore', *World sustainable buildings 2014, sustainable building: Results are we moving as quickly as we should? Its up to us*, pp. 455–461.

Konys, A. (2018) 'An Ontology-Based Knowledge Modelling for a Sustainability Assessment Domain', *Sustainability (Switzerland)*, 10(2). doi: 10.3390/su10020300.

Kujawski, W. (2013) 'How post occupancy evaluation can benefit owners, designers and occupants of new and existing residential buildings', *Central Europe towards Sustainable Building 2013*, pp. 1–9.

Leaman, A. (2011) *BUS occupant survey method : Details for licensees*.

Leaman, A. and Bordass, B. (2001) 'Assessing building performance in use 4: The Probe occupant surveys and their implications', *Building Research and Information*, 29(2), pp. 129–143. doi: 10.1080/09613210010008045.

Leaman, A., Stevenson, F. and Bordass, B. (2010) 'Building evaluation: Practice and principles', *Building Research and Information*, 38(5), pp. 564–577. doi: 10.1080/09613218.2010.495217.

Li, P., Froese, T. M. and Brager, G. (2018) 'Post-occupancy evaluation: State-of-the-art analysis and state-of-the-practice review', *Building and Environment*. Elsevier, 133(February), pp. 187–202. doi: 10.1016/j.buildenv.2018.02.024.

Li, Y. *et al.* (2021) 'Semiautomatic Generation of Code Ontology Using ifcOWL in Compliance Checking', *Advances in Civil Engineering*. Edited by H. Yao. Hindawi, 2021, pp. 1–18. doi: 10.1155/2021/8861625.

Liu, J. Y., Low, S. P. and He, X. (2012) 'Green practices in the Chinese building industry: drivers and impediments', *Journal of Technology Management in China*, 7(1), pp. 50–63. doi: 10.1108/17468771211207349.

Liu, X., Li, Z. and Jiang, S. (2016) 'Ontology-based representation and reasoning in building construction cost estimation in China', *Future Internet*, 8(3). doi: 10.3390/fi8030039.

Lork, C. *et al.* (2019) 'An Ontology-Based Framework for Building Energy Management with IoT', *Electronics*, 8(5), p. 485. doi: 10.3390/electronics8050485.

Lu, Y. *et al.* (2015) 'Ontology-based knowledge modeling for automated construction safety checking', *Safety Science*. Elsevier Ltd, 79, pp. 11–18. doi: 10.1016/j.ssci.2015.05.008.

Ma, S. and Tian, L. (2014) 'Ontology-based semantic retrieval for mechanical design knowledge', *International Journal of Computer Integrated Manufacturing*. Taylor & Francis, 28(2), pp. 226–238. doi: 10.1080/0951192X.2013.874593.

Ma, Y. and Verbeke, S. (2021) *Smart Readiness Indicator (SRI) ASSESSMENT PACKAGE : PRACTICAL GUIDE SRI CALCULATION FRAMEWORK v 4.1*.

Maedche, A. and Staab, S. (2001) 'Ontology Learning for the Semantic Web', *IEEE Intelligent Systems*, 16(2), pp. 72–79. doi: 10.1109/5254.920602.

Mahdavi, A. *et al.* (2021) 'The role of occupants in buildings' energy performance gap: Myth or reality?', *Sustainability (Switzerland)*, 13(6). doi: 10.3390/su13063146.

Mahdavi, A. and Taheri, M. (2017) 'An ontology for building monitoring', *Journal of Building Performance Simulation*, 10(5–6), pp. 499–508. doi: 10.1080/19401493.2016.1243730.

Mahdavi, A., Taheri, M. and Wolosiuk, D. (2019) 'A Building Performance Indicator Ontology: Structure and Applications', in *16th IBPSA Conference*, pp. 77–82. doi: 10.26868/25222708.2019.210112.

Mastor, S. H. and Ibrahim, N. (2010) 'Post Occupancy Evaluation Practices: A Procedural

Model For A Successful Feedback', in *Proceedings of the CIB 2010 World Congress*, pp. 200–210.

Matoušek, K., Král, L. and Falc, M. (2004) *Apollo CH Manual, The Open University*.

McGuinness, D. L. and Harmelen, F. van (2004) *OWL Web Ontology Language Overview*. Available at: https://www.w3.org/TR/owl-features/ (Accessed: 11 October 2021).

Meir, I. A. *et al.* (2009) 'Post-occupancy evaluation: An inevitable step toward sustainability', *Advances in Building Energy Research*, 3(1), pp. 189–220. doi: 10.3763/aber.2009.0307.

Moreira, D. A. (2012) 'SWRL Rule Editor - A Web Application as Rich as Desktop Business Rule Editors', *Proceedings of the 14th International Conference on Enterprise Information Systems*, (January), pp. 258–263. doi: 10.5220/0003999402580263.

Munir, K. and Sheraz Anjum, M. (2018) 'The use of ontologies for effective knowledge modelling and information retrieval', *Applied Computing and Informatics*. King Saud University, 14(2), pp. 116–126. doi: 10.1016/j.aci.2017.07.003.

Musen, M. A. (2015a) 'The protégé project: A Look Back and a Look Forward', *AI Matters*, 1(4), pp. 4–12. doi: 10.1145/2757001.2757003.

Musen, M. A. (2015b) 'The Protégé Project: A Look Back and a Look Forward', *AI Matters*, 1(4), pp. 4–12. doi: 10.1145/2757001.2757003.

Nawawi, A. H. and Khalil, N. (2008) 'Post-occupancy evaluation correlated with building occupants' satisfaction: An approach to performance evaluation of government and public buildings', *Journal of Building Appraisal*, 4(2), pp. 59–69. doi: 10.1057/jba.2008.22.

Ning, Y. and Chen, J. (2016) 'Improving residential satisfaction of university dormitories through post-occupancy evaluation in China: A socio-technical system approach', *Sustainability (Switzerland)*, 8(10). doi: 10.3390/su8101050.

Noy, N. F. and McGuinness, D. L. (2001) *Ontology Development 101: A Guide to Creating Your First Ontology, Stanford Knowledge Systems Laboratory*.

O'Connor, M. *et al.* (2005) 'Writing Rules for the Semantic Web Using SWRL and Jess', *Protégé With Rules WS Madrid*, (July 2014). Available at: http://www.med.univ-rennes1.fr/~cgolb/Protege2005/SWRLJessOConnor.pdf.

O'Connor, M., Tu, S., *et al.* (2007) 'Querying the semantic Web with SWRL', *Lecture Notes* *in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4824 LNCS, pp. 155–159. doi: 10.1007/978-3-540-75975-1_13.

O'Connor, M., Shankar, R., *et al.* (2007) 'Using semantic Web technologies for knowledge-driven querying of biomedical data', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4594 LNAI, pp. 267–276. doi: 10.1007/978-3-540-73599-1_36.

O'Connor, M. (2016) *SQWRL*, *GitHub*. Available at: https://github.com/protegeproject/swrlapi/wiki/SQWRL (Accessed: 13 October 2021).

O'Connor, M. and Das, A. (2006) 'The SWRLTab: An Extensible Environment for working with SWRL Rules in Protégé-OWL'.

O'Connor, M. and Das, A. (2009) 'SQWRL: A query language for OWL', in *CEUR Workshop Proceedings*, pp. 3–10.

O'Connor, M. J. and Das, A. (2005) 'The SWRLTab: An Extensible Environment for working with SWRL Rules in Protege-OWL', (January), pp. 2–4. Available at: https://protege.stanford.edu/conference/2007/slides/08.01_OConnor.pdf.

O'Connor, M. J. and Das, A. (2012) 'A pair of OWL 2 RL reasoners', *CEUR Workshop Proceedings*, 849(January 2012).

Öhgren, A. (2004) *Ontology Development and Evolution: Selected Approaches for Small-Scale Application Contexts*. Available at: https://www.diva-portal.org/smash/get/diva2:4295/FULLTEXT01.pdf.

Oriţă, A., Drăghici, G. and Beney, J.-L. (2013) 'Use of ontological classes in the exploration of user needs', *Applied Mechanics and Materials*, 371, pp. 847–851. doi: 10.4028/www.scientific.net/AMM.371.847.

Orozco, A. T. *et al.* (2019) 'An ontology-based thermal comfort management system in smart buildings', *11th International Conference on Management of Digital EcoSystems, MEDES 2019*, pp. 300–307. doi: 10.1145/3297662.3365824.

Palmer, J., Terry, N. and Armitage, P. (2016) *Building Performance Evaluation Programme: Findings from non-domestic projects Getting the best from buildings*, *Innovate UK*. Available at: http://www.carboncredentials.com/wp-content/uploads/2016/04/Non-Domestic_Building_performance_full_report_2016.compressed.pdf.

Parkinson, A. T. *et al.* (2018) ‘Evaluating positivist theories of occupant satisfaction: a statistical analysis’, *Building Research and Information*. Taylor & Francis, 46(4), pp. 430–443. doi: 10.1080/09613218.2017.1314148.

Patel, A. *et al.* (2021) *Covid19-IBO: A Covid-19 Impact on Indian Banking Ontology Along with an Efficient Schema Matching Approach*, *New Generation Computing*. Ohmsha. doi: 10.1007/s00354-021-00136-0.

Pauwels, P. *et al.* (2017) ‘Enhancing the ifcOWL ontology with an alternative representation for geometric data’, *Automation in Construction*. Elsevier B.V., 80, pp. 77–94. doi: 10.1016/j.autcon.2017.03.001.

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

Pauwels, P., Zhang, S. and Lee, Y. C. (2017) ‘Semantic web technologies in AEC industry: A literature overview’, *Automation in Construction*. Elsevier B.V., 73, pp. 145–165. doi: 10.1016/j.autcon.2016.10.003.

Pereira, N. B., Rodrigues, R. C. and Rocha, P. F. (2016) ‘Post-Occupancy Evaluation Data Support for Planning and Management of Building Maintenance Plans’, *Buildings*, 6(4), p. 45. doi: 10.3390/buildings6040045.

Peroni, S. (2017) ‘A simplified agile methodology for ontology development’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10161 LNCS, pp. 55–69. doi: 10.1007/978-3-319-54627-8_5.

Polenghi, A. *et al.* (2021) ‘Ontology-augmented Prognostics and Health Management for shopfloor-synchronised joint maintenance and production management decisions’, *Journal of Industrial Information Integration*. Elsevier Masson SAS, 46(Icict 2014), p. 100286. doi: 10.1016/j.jii.2021.100286.

Preiser, W. F. E. (1995) ‘Post-occupancy Evaluation: How to make buildings works better’, *Facilities*, 13(11), pp. 19–28. doi: 02632779510097787.

Preiser, W. F. E. (2002) ‘Continuous quality improvement through post-occupancy evaluation feedback’, *Journal of Corporate Real Estate*, 5(1), pp. 42–56. doi: 10.1108/14630010310811993.

Preiser, W. F. E. and Schramm, U. (1997) *Building performance evaluation, Time Saver Standards (7th Edition). New York: McGraw Hill*. Springer.

Preiser, W. F. E. and Schramm, U. (2002) 'Intelligent office building performance evaluation', *Facilities*, 20(7/8), pp. 279–287. doi: 10.1108/02632770210435198.

Preiser, W. F. E. and Vischer, J. C. (2005) *Assessing building performance, Elsevier Butterworth-Heinemann*. Edited by W. F. E. Preiser and J. C. Vischer. Elsevier Butterworth-Heinemann. doi: 10.1073/pnas.0703993104.

Preuveneers, D. and Berbers, Y. (2005) 'Automated context-driven composition of pervasive services to alleviate non-functional concerns', *CEUR Workshop Proceedings*, 150(December 2013).

Pritoni, M. *et al.* (2021) 'Metadata Schemas and Ontologies for Building Energy Applications: A Critical Review and Use Case Analysis', *Energies*, 14(7), p. 2024. doi: 10.3390/en14072024.

Protégé (2016) *Protégé, Stanford University*. Available at: https://protege.stanford.edu/ (Accessed: 15 October 2021).

Pruvost, H. and Scherer, R. J. (2017) 'Analysis of Risk in Building Life Cycle Coupling BIM-based Energy Simulation and Semantic Modeling', *Procedia Engineering*. The Author(s), 196(June), pp. 1106–1113. doi: 10.1016/j.proeng.2017.08.068.

Putra, H. C., Hong, T. and Andrews, C. (2021) 'An ontology to represent synthetic building occupant characteristics and behavior', *Automation in Construction*. Elsevier B.V., 125(August 2020), p. 103621. doi: 10.1016/j.autcon.2021.103621.

Reinisch, C. *et al.* (2011) 'ThinkHome Energy Efficiency in Future Smart Homes', *EURASIP Journal on Embedded Systems*, 2011(1), p. 104617. doi: 10.1155/2011/104617.

Reyes-Peña, C. and Tovar-Vidal, M. (2019) 'Ontology: Components and Evaluation, a Review', *Research in Computing Science*, 148(3), pp. 257–265. doi: 10.13053/rcs-148-3-21.

Sachs, E. (2006) *Getting started with Protégé-frames, Standford University*. Available at: ftp://140.118.199.9:2100/public9/ZZproject/zEDB_Buffer/0421_Ontology/REF_forSnapShot/C-Get-started(Protege-tutorial-2006).pdf.

Sanya, I. O. and Shehab, E. M. (2014) 'An ontology framework for developing platform-independent knowledge-based engineering systems in the aerospace industry', *International*

Journal of Production Research. Taylor & Francis, 52(20), pp. 6192–6215. doi: 10.1080/00207543.2014.919422.

Sanya, I. O. and Shehab, E. M. (2015) 'A framework for developing engineering design ontologies within the aerospace industry', *International Journal of Production Research*, 53(8), pp. 2383–2409. doi: 10.1080/00207543.2014.965352.

Schneider, G. F., Pauwels, P. and Steiger, S. (2017) 'Ontology-Based Modeling of Control Logic in Building Automation Systems', *IEEE Transactions on Industrial Informatics*, 13(6), pp. 3350–3360. doi: 10.1109/TII.2017.2743221.

Schreiber, G. *et al.* (2000) *Knowledge Engineering and Management- The CommonKADS Methodology*. 2nd edn, *Knowledge Engineering and Management*. 2nd edn. MIT Press. doi: 10.7551/mitpress/4073.001.0001.

Schreiber, G., Wielinga, B. J. and Jansweijer, W. N. . (1995) 'The KACTUS View on the ' O ' Word', *IJCAI workshop on basic ontological issues in knowledge sharing*, pp. 159–168.

Scofield, J. H. (2009) 'Do LEED-certified buildings save energy? Not really...', *Energy and Buildings*. Elsevier B.V., 41(12), pp. 1386–1390. doi: 10.1016/j.enbuild.2009.08.006.

Shan, M. and Hwang, B. gang (2018) 'Green building rating systems: Global reviews of practices and research efforts', *Sustainable Cities and Society*, 39(February), pp. 172–180. doi: 10.1016/j.scs.2018.02.034.

Sim, W. W. and Brouse, P. (2014) 'Towards an Ontology-Based Persona-Driven Requirements and knowledge engineering', *Procedia Computer Science*. Elsevier Masson SAS, 36(C), pp. 314–321. doi: 10.1016/j.procs.2014.09.099.

Simeone, D. and Fioravanti, A. (2012) 'An ontology-based system to support agent-based simulation of building use', *Electronic Journal of Information Technology in Construction*, 17(September), pp. 258–270.

Simperl, E. (2009) 'Reusing ontologies on the Semantic Web: A feasibility study', *Data and Knowledge Engineering*. Elsevier B.V., 68(10), pp. 905–925. doi: 10.1016/j.datak.2009.02.002.

Slimani, T. (2015) 'A Study Investigating Typical Concepts and Guidelines for Ontology Building', pp. 1–7. Available at: http://arxiv.org/abs/1509.05434.

Smirnov, A. *et al.* (2021) 'Methodology for Multi-Aspect Ontology Development: Ontology for Decision Support Based on Human-Machine Collective Intelligence', *IEEE Access*. IEEE, 9, pp. 1–1. doi: 10.1109/access.2021.3116870.

Smith, M. K., Welty, C. and McGuinness, D. L. (2004) *OWL Web Ontology Language Guide W3C Recommendation 10 February 2004*.

Staab, S., Studer, R. and Gmbh, O. (2001) 'Knowledge Processes and Ontologies', *Intelligent Systems, IEEE*. IEEE, 16(1), pp. 26–34.

Stavropoulos, T. *et al.* (2012) 'BOnSAI: a Smart Building Ontology for Ambient Intelligence', in *ACM International Conference Proceeding Series*, pp. 32–33. doi: 10.2307/j.ctt9qh818.16.

Stevens, R., Goble, C. A. and Bechhofer, S. (2000) 'Ontology-based knowledge representation for bioinformatics', *Briefings in bioinformatics*, 1(4), pp. 398–414. doi: 10.1093/bib/1.4.398.

Stevenson, F. (2009) 'Post-occupancy evaluation and sustainability: a review', *Proceedings of the ICE - Urban Design and Planning*, 162(DP3), pp. 123–130. doi: 10.1680/udap.2009.162.

Stevenson, F. (2018) 'Embedding building performance evaluation in UK architectural practice and beyond', *Building Research & Information*. Taylor & Francis, 0(0), pp. 1–13. doi: 10.1080/09613218.2018.1467542.

Studer, R., Benjamins, V. R. and Fensel, D. (1998) 'Knowledge Engineering: Principles and methods', *Data and Knowledge Engineering*, 25(1–2), pp. 161–197. doi: 10.1016/S0169-023X(97)00056-6.

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

Svetel, I. and Pejanovic, M. (2010) 'The Role of the Semantic Web for Knowledge Management in the Construction Industry Knowledge in AEC Open standards for AEC Based on years of research on a general data model for', *Knowledge Creation Diffusion Utilization*, 34, pp. 331–336.

Swartout, B. *et al.* (1997) *Toward Distributed Use of Large-Scale Ontologies*, *Ontological Engineering AAAI-97 Spring Symposium Series*.

Szturcová, D. and Rapant, P. (2013) 'Enhanced methodology for ontology development', *Computing and Informatics*, 32(5), pp. 1038–1054.

Tah, J. H. M. and Abanda, H. F. (2011) 'Sustainable building technology knowledge representation: Using Semantic Web techniques', *Advanced Engineering Informatics*. Elsevier Ltd, 25(3), pp. 547–558. doi: 10.1016/j.aei.2011.02.006.

TAKHOM, A. (2019) *A COLLABORATIVE ONTOLOGY DEVELOPMENT FRAMEWORK FOR ENABLING MULTIDISCIPLINARY KNOWLEDGE CONSTRUCTIONS A COLLABORATIVE ONTOLOGY DEVELOPMENT FRAMEWORK FOR ENABLING MULTIDISCIPLINARY KNOWLEDGE CONSTRUCTIONS*. THAMMASAT UNIVERSITY.

Tao, M., Ota, K. and Dong, M. (2017) 'Ontology-based data semantic management and application in IoT- and cloud-enabled smart homes', *Future Generation Computer Systems*. Elsevier B.V., 76, pp. 528–539. doi: 10.1016/j.future.2016.11.012.

Taye, M. M. (2010) 'Understanding Semantic Web and Ontologies: Theory and Applications', *Journal of Computing*, 2(6), pp. 469–482. Available at: http://arxiv.org/abs/1006.4567.

Taylor, J. *et al.* (2018) 'Towards a framework to evaluate the "total" performance of buildings', *Building Services Engineering Research and Technology*, 39(5), pp. 609–631. doi: 10.1177/0143624418762662.

Teasdale-St-Hilaire, A. (2013) 'Post-Occupancy Evaluation Framework for Multi-Unit Residential Buildings', in *Thermal Performance of the Exterior Envelopes of Whole Buildings XII Internal Conference*, pp. 1–12. doi: ISBN : 9781510827837.

Thatcher, A. and Milner, K. (2016) 'Is a green building really better for building occupants? A longitudinal evaluation', *Building and Environment*. Elsevier Ltd, 108, pp. 194–206. doi: 10.1016/j.buildenv.2016.08.036.

Tserng, H. P. *et al.* (2009) 'A study of ontology-based risk management framework of construction projects through project life cycle', *Automation in Construction*. Elsevier B.V., 18(7), pp. 994–1008. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0926580509000855.

Turpin-Brooks, S. and Viccars, G. (2006) 'The development of robust methods of post occupancy evaluation', *Facilities*, 24(5–6), pp. 177–196. doi: 10.1108/02632770610665775.

UK Green Building Council (2013) *PINPOINTING : BUS Methodology*.

UKGBC (2016) *DELIVERING BUILDING PERFORMANCE*.

Ukpe, E. and Mustapha, S. M. F. D. S. (2016) 'An Analysis of Six Standard Ontology Editing

Tools for Capturing Entire Crop Processing', *International Journal of Computer Science and Information Security (IJCSIS)*, 14(1), pp. 18–24.

United Nations Environment Programme (2020) *2020 Global Status Report for Buildings and Construction - Towards a zero-emissions, efficient and resilient buildings and construction sector, Global Status Report*.

Uschold, M. and Gruninger, M. (1996) 'Ontologies : Principles , Methods and Applications', *The Knowledge Engineering Review*, 11(2), pp. 93–136. doi: 10.1.1.111.5903.

Uschold, M. and King, M. (1995) 'Towards a Methodology for Building Ontologies', *Workshop on Basic Ontological Issues in Knowledge Sharing*, (July).

Vigna, I. *et al.* (2020) 'Office occupants' perspective dealing with energy flexibility: A large-scale survey in the province of bolzano', *Energies*, 13(17). doi: 10.3390/en13174312.

Vischer, J. (2001) 'Post-Occupancy Evaluation: A Multifaceted Tool for Building Improvement', in *LEARNING FROM OUR BUILDINGS: A State-of-the-Practice Summary of Post-Occupancy Evaluation*, pp. 23–34. Available at: https://www.researchgate.net/profile/Jacqueline_Vischer/publication/236144016_Post-Occupancy_Evaluation_A_Multifaceted_Tool_for_Building_Improvement/links/55db38a208aeb38e8a8b7da1.pdf?origin=publication_detail&ev=pub_int_prw_xdl&msrp=U0ueTnA5zGMETn73hzbPl.

W3C (2014) *RDF Schema 1.1 W3C Recommendation*. Available at: https://www.w3.org/TR/rdf-schema/#ch_comment.

Wang, H.-H. and Boukamp, F. (2011) 'Ontology-Based Representation and Reasoning Framework for Supporting Job Hazard Analysis', *Journal of Computing in Civil Engineering*, 25(6), pp. 442–456. doi: 10.1061/(asce)cp.1943-5487.0000125.

Wang, X. (2011) 'The Theory of Post-Occupancy Evaluation and A Case Study on A Critical Appraisal of Tsinghua University Fine Arts Academy', *China Academic Journal*, pp. 84–90.

Watson, C. (2007) 'Green Buildings in Use : Post Occupancy Evaluations', *PEB Exchange: The Journal of the OECD Programme on Educational Building*, 12, pp. 1–4. doi: 10.1787/070286367536.

Wicaksono, H., Aleksandrov, K. and Rogalski, S. (2012) 'An Intelligent System for Improving Energy Efficiency in Building Using Ontology and Building Automation Systems', in

Automation. doi: 10.5772/48006.

De Wilde, P. (2014) 'The gap between predicted and measured energy performance of buildings: A framework for investigation', *Automation in Construction*. Elsevier B.V., 41, pp. 40–49. doi: 10.1016/j.autcon.2014.02.009.

Wolosiuk, D. and Mahdavi, A. (2020) 'Application of Ontologically Structured Data for Building Performance Analysis', in *11th Annual Symposium on Simulation for Architecture and Urban Design*, pp. 55–64.

Woo, J (2017) 'A post-occupancy evaluation of a modular multi-residential development in Melbourne, Australia', in *International High-Performance Built Environment Conference - a Sustainable Built Environment Conference 2016 Series,* pp. 365–372. doi: 10.1016/j.proeng.2017.04.195.

Woo, Jin (2017) 'A Post-occupancy Evaluation of a Modular Multi-residential Development in Melbourne, Australia', in *International High-Performance Built Environment Conference - a Sustainable Built Environment Conference 2016 Series*. Elsevier B.V., pp. 365–372. doi: 10.1016/j.proeng.2017.04.195.

Wu, F. *et al.* (2007) 'Improving indoor environmental quality for public health: Impediments and policy recommendations', *Environmental Health Perspectives,* 115(6), pp. 953–957. doi: 10.1289/ehp.8986.

Wu, H. *et al.* (2021) 'Combining computer vision with semantic reasoning for on-site safety management in construction', *Journal of Building Engineering*. Elsevier Ltd, 42(March), p. 103036. doi: 10.1016/j.jobe.2021.103036.

Xiao, J. *et al.* (2018) 'Ontology-Based Knowledge Model to Support Construction Noise Control in China', *Journal of Construction Engineering and Management,* 144(2), pp. 1–17. doi: 10.1061/(ASCE)CO.1943-7862.0001425.

Yan, W. *et al.* (2015) 'Ontology-based knowledge modeling for using physical effects', *Procedia Engineering*. Elsevier B.V., 131, pp. 601–615. doi: 10.1016/j.proeng.2015.12.454.

Yee-King, M. J. *et al.* (2019) 'Technology Enhanced Learning: The Role of Ontologies for Feedback in Music Performance', *Frontiers in Digital Humanities,* 5(January), pp. 1–17. doi: 10.3389/fdigh.2018.00029.

Yoo, D. and No, S. (2014) 'Ontology-based economics knowledge sharing system', *Expert*

Systems with Applications. Elsevier Ltd, 41(4 PART 1), pp. 1331–1341. doi: 10.1016/j.eswa.2013.08.031.

Yu, J., Ma, G. and Chen, Y. (2017) 'Comparison of the Built Environment of Homes for the Aged in China by Postoccupancy Evaluation', *Journal of Architectural Engineering*, 23(3). doi: 10.1061/(ASCE)AE.1943-5568.0000264.

Yu, L. *et al.* (2015) 'The Research of POE System and Method of Urban Green Residential Environmental Performance in Severe Cold and Cold Regions', *Procedia Engineering*. Elsevier B.V., 121, pp. 107–113. doi: 10.1016/j.proeng.2015.08.1030.

Yu, L. *et al.* (2016) 'The Practice Research Based on the POE System of Environmental Performance of Green Residential Building', *Procedia Engineering*, 146, pp. 204–209. doi: 10.1016/j.proeng.2016.06.374.

Žáček, M. and Janošek, M. (2017) 'SBOnto: ONTOLOGY OF SMART BUILDING', *Far East Journal of Electronics and Communications*, 17(5), pp. 1101–1109. doi: 10.17654/EC017051101.

Zero Carbon Hub (2013) *Closing the Gap Between Design and As-built Performance*. Available at: http://www.zerocarbonhub.org/recent-publications.

Zhang, D. *et al.* (2019) 'A semantic and social approach for real-time green building rating in BIM-based design', *Sustainability (Switzerland)*, 11(14), pp. 1–16. doi: 10.3390/su11143973.

Zhang, D., Hu, D. and Xu, Y. (2010) 'A framework for ontology-based product design knowledge management', *2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery*, (Fskd), pp. 1751–1755. doi: 10.1109/FSKD.2010.5569405.

Zhang, J. and El-Gohary, N. M. (2015) 'Automated Information Transformation for Automated Regulatory Compliance Checking in Construction', *Journal of Computing in Civil Engineering*, 29(4), pp. 1–16. doi: 10.1061/(asce)cp.1943-5487.0000427.

Zhang, J. and El-Gohary, N. M. (2016) 'Semantic NLP-Based Information Extraction from Construction Regulatory Documents for Automated Compliance Checking', *Journal of Computing in Civil Engineering*, 30(2), p. 04015014. doi: 10.1061/(ASCE)CP.1943-5487.0000346.

Zhang, L. and Issa, R. R. A. (2013) 'Ontology-Based Partial Building Information Model Extraction', *Journal of Computing in Civil Engineering*, 27(6), pp. 576–584. doi:

10.1061/(asce)cp.1943-5487.0000277.

Zhang, S., Boukamp, F. and Teizer, J. (2014) ‘Ontology-Based Semantic Modeling of Safety Management Knowledge’, *COMPUTING IN CIVIL AND BUILDING ENGINEERING*, pp. 2254–2262.

Zhang, S., Boukamp, F. and Teizer, J. (2015) ‘Ontology-based semantic modeling of construction safety knowledge: Towards automated safety planning for job hazard analysis (JHA)’, *Automation in Construction*. Elsevier B.V., 52, pp. 29–41. doi: 10.1016/j.autcon.2015.02.005.

Zhao, Y. *et al.* (2020) ‘Ontology-based Knowledge Modeling of Post-occupancy Evaluation for Green Building’, in *4th International Conference on Environmental and Energy Engineering (IC3E 2020)*. IOP Conference Series: Earth and Environmental Science, p. 012076. doi: 10.1088/1755-1315/495/1/012076.

Zhao, Y. and Yang, Q. (2022) ‘A Post-occupancy Evaluation of Occupant Satisfaction in Green and Conventional Higher Educational Buildings’, *The 6th International Conference on Environmental Engineering and Sustainable Development (CEESD 2021)*. IOP Conference Series: Earth and Environmental Science, 973(1), p. 012010. doi: 10.1088/1755-1315/973/1/012010.

Zhao, Y. and Yang, Qingping (2021) ‘Development of an ontology-based semantic building post-occupancy evaluation framework’, *International Journal of Metrology and Quality Engineering*. Edited by QingPing Yang, 12(19), p. 19. doi: 10.1051/ijmqe/2021019.

Zheng, Y., Törmä, S. and Seppänen, O. (2021) ‘A shared ontology suite for digital construction workflow’, *Automation in Construction*, 132(August). doi: 10.1016/j.autcon.2021.103930.

Zhong, B. *et al.* (2018) ‘Ontology-based framework for building environmental monitoring and compliance checking under BIM environment’, *Building and Environment*. Elsevier, 141(December 2017), pp. 127–142. doi: 10.1016/j.buildenv.2018.05.046.

Zhong, B. *et al.* (2019) ‘A scientometric analysis and critical review of construction related ontology research’, *Automation in Construction*. Elsevier, 101(September 2020), pp. 17–31. doi: 10.1016/j.autcon.2018.12.013.

Zhong, B. and Li, Y. (2015) ‘An Ontological and Semantic Approach for the Construction Risk Inferring and Application’, *Journal of Intelligent and Robotic Systems: Theory and*

Applications, 79(3–4), pp. 449–463. doi: 10.1007/s10846-014-0107-9.

Zhong, B. T. *et al.* (2012) 'Automation in Construction Ontology-based semantic modeling of regulation constraint for automated construction quality compliance checking', *Automation in Construction*, 28, pp. 58–70. doi: 10.1016/j.autcon.2012.06.006.

Zhong, B. T. *et al.* (2015) 'An ontological approach for technical plan definition and verification in construction', *Automation in Construction*, 55, pp. 47–57. doi: 10.1016/j.autcon.2015.02.002.

Zhou, P. and El-Gohary, N. (2016) 'Ontology-Based Multilabel Text Classification of Construction Regulatory Documents', *Journal of Computing in Civil Engineering*, 30(4), p. 04015058. doi: 10.1061/(ASCE)CP.1943-5487.0000530.

Zhou, Y. (2014) *Comparison of Chinese Green Building Standard with Western Green Building standards, KTH Engineering and Management*. Available at: http://www.diva-portal.org/smash/get/diva2:735240/FULLTEXT01.pdf.

Zhou, Z., Goh, Y. M. and Shen, L. (2016) 'Overview and Analysis of Ontology Studies Supporting Development of the Construction Industry', *Journal of Computing in Civil Engineering*, 30(6), pp. 1–14.

Zimmerman, A. and Martin, M. (2001) 'Post-occupancy evaluation: Benefits and barriers', *Building Research and Information*, 29(2), pp. 168–174. doi: 10.1080/09613210010016857.

Zimring, C. M. and Reizenstein, J. E. (1980) 'POST-OCCUPANCY EVALUATION An Overview', *ENVIRONMENT AND BEHAVIOR*, 12(4), pp. 429–450.

APPENDIX I SWRL rules and SQWRL queries

SWRL Rules Examples
Rule_Scale midpoint lower value
EvaluationCriteria(?ec) ^ swrlb:multiply(?y, ?se, 1.96) ^ hasScaleMidpointValue(?ec, ?smv) ^ hasSE(?ec,?se) ^ swrlb:subtract(?x, ?smv,?y) -> hasScaleMidpointLowerValue(?ec, ?x)
Rule_Scale midpoint upper value
EvaluationCriteria(?ec) ^ hasScaleMidpointValue(?ec, ?smv) ^ swrlb:multiply(?y, ?se,1.96) ^ swrlb:add(?x, ?smv, ?y) ^ hasSE(?ec, ?se) -> hasScaleMidpointUpperValue(?ec, ?x)
Rule_Benchmaker lower value
swrlb:multiply(?y, ?se, 1.96)^swrlb:subtract(?x, ?bmv, ?y)^ hasBenchamrkMeanValue(?ec, ?bmv)^ hasSE(?ec, ?se) ^ EvaluationCriteria(?ec) -> hasBenchmarkLowerValue(?ec, ?x)
Rule_Benchmark upper value
swrlb:multiply(?y, ?se, 1.96) ^ hasBenchamrkMeanValue(?ec, ?bmv) ^ swrlb:add(?x, ?bmv, ?y) ^ hasSE(?ec,?se) ^ EvaluationCriteria(?ec) -> hasBenchmarkUpperValue(?ec, ?x)
Rule_Lower critical limit value
swrlb:lessThan(?bmlv,?smlv)^ EvaluationCriteria(?ec) ^hasScaleMidpointLowerValue(?ec, ?smlv) ^ hasBenchmarkLowerValue(?ec, ?bmlv) -> hasLowerCriticalLimitValue(?ec, ?bmlv)
EvaluationCriteria(?ec)^ hasScaleMidpointLowerValue(?ec,?smlv) ^swrlb:lessThan(?smlv, ?bmlv) ^ hasBenchmarkLowerValue(?ec, ?bmlv) -> hasLowerCriticalLimitValue(?ec, ?smlv)
Rule_Upper critical limit value
EvaluationCriteria(?ec)^hasBenchmarkUpperValue(?ec,?bmuv)^swrlb:greaterThan(?bmuv,?smuv) ^ hasScaleMidpointUpperValue(?ec,?smuv)^ ->hasUpperCriticalLimitValue(?ec, ?bmuv)
EvaluationCriteria(?ec)^swrlb:greaterThan(?smuv,?bmuv)^hasBenchmarkUpperValue(?ec,?bmuv) ^ hasScaleMidpointUpperValue(?ec,?smuv) -> hasUpperCriticalLimitValue(?ec, ?smuv)
Rule_Air movement in summer/ winter:
EvaluatedNonDomesticBuilding(?endb) ^ hasUpperCriticalLimitValue(?cairsmovement, ?uclv)^ swrlb:greaterThan(?mrairsmovement, ?uclv) ^ CriteriaAirMovementInSummer(?cairsmovement) ^ hasMeanResponseValueAIRSMOVEMENT(?endb, ?mrairsmovement) -> hasResultAIRSMOVEMENT(?endb, TooDraughty)
EvaluatedNonDomesticBuilding(?endb) ^ hasLowerCriticalLimitValue(?cairsmovement, ?lclv) ^ CriteriaAirMovementInSummer(?cairsmovement) ^ swrlb:lessThan(?mrairsmovement, ?lclv) ^ hasMeanResponseValueAIRSMOVEMENT(?endb, ?mrairsmovement) -> hasResultAIRSMOVEMENT(?endb, Still)
EvaluatedNonDomesticBuilding(?endb) ^hasLowerCriticalLimitValue(?cairsmovement,?lclv) ^ hasUpperCriticalLimitValue(?cairsmovement,?uclv)^swrlb:lessThanOrEqual(?mrairsmovement, ?uclv)^hasMeanResponseValueAIRSMOVEMENT(?endb,?mrairsmovement)^swrlb:greaterThanOrEqual(?mrairsmovement, ?lclv) ^ CriteriaAirMovementInSummer(?cairsmovement) -> hasResultAIRSMOVEMENT(?endb, Satisfactory)
Rule_Availability of meeting rooms

EvaluatedNonDomesticBuilding(?endb) ^ hasMeanResponseValueMEETING(?endb,?mrmeeting)^ hasUpperCriticalLimitValue(?cmeeting, ?uclv) ^ CriteriaAvailabilityOfMeetingRooms(?cmeeting) ^ swrlb:greaterThan(?mrmeeting, ?uclv) -> hasResultMEETING(?endb, Outstanding)
hasMeanResponseValueMEETING(?endb, ?mrmeeting)^hasLowerCriticalLimitValue(?cmeeting, ?lclv) ^ CriteriaAvailabilityOfMeetingRooms(?cmeeting) ^ swrlb:lessThan(?mrmeeting, ?lclv) ^ EvaluatedNonDomesticBuilding(?endb)->hasResultMEETING(?endb, Unsatisfactory)
swrlb:lessThanOrEqual(?mrmeeting, ?uclv) ^ hasUpperCriticalLimitValue(?cmeeting, ?uclv) ^ hasMeanResponseValueMEETING(?endb,?mrmeeting)^hasLowerCriticalLimitValue(?cmeeting, ?lclv)^CriteriaAvailabilityOfMeetingRooms(?cmeeting)^swrlb:greaterThanOrEqual(?mrmeeting, ?lclv) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultMEETING(?endb, Satisfactory)
Rule_Buildign design overall
hasMeanResponseValueDESIGN(?endb, ?mrdesign) ^ CriteriaBuildingDesignOverall(?cdesign) ^ swrlb:greaterThan(?mrdesign, ?uclv) ^ hasUpperCriticalLimitValue(?cdesign, ?uclv) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultDESIGN(?endb, Outstanding)
hasMeanResponseValueDESIGN(?endb, ?mrdesign) ^ CriteriaBuildingDesignOverall(?cdesign) ^ hasLowerCriticalLimitValue(?cdesign, ?lclv) ^ swrlb:lessThan(?mrdesign, ?lclv) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultDESIGN(?endb, Unsatisfactory)
hasMeanResponseValueDESIGN(?endb, ?mrdesign) ^ CriteriaBuildingDesignOverall(?cdesign) ^ hasLowerCriticalLimitValue(?cdesign, ?lclv) ^ hasUpperCriticalLimitValue(?cdesign, ?uclv) ^ EvaluatedNonDomesticBuilding(?endb) ^ swrlb:lessThanOrEqual(?mrdesign, ?uclv) ^ swrlb:greaterThanOrEqual(?mrdesign, ?lclv) -> hasResultDESIGN(?endb, Satisfactory)
Rule_Cleaning
hasMeanResponseValueCLEANING(?endb,?mrcleaning)^hasUpperCriticalLimitValue(?ccleaning, ?uclv) ^ swrlb:greaterThan(?mrcleaning,?uclv) ^ EvaluatedNonDomesticBuilding(?endb)^ CriteriaCleaning(?ccleaning) -> hasResultCLEANING(?endb, Outstanding)
hasMeanResponseValueCLEANING(?endb,?mrcleaning)^hasLowerCriticalLimitValue(?ccleaning ,?lclv)^swrlb:lessThan(?mrcleaning, ?lclv) ^ EvaluatedNonDomesticBuilding(?endb) ^ CriteriaCleaning(?ccleaning) -> hasResultCLEANING(?endb, Unsatisfactory)
hasMeanResponseValueCLEANING(?endb,?mrcleaning)^hasUpperCriticalLimitValue(?ccleaning, ?uclv) ^ hasLowerCriticalLimitValue(?ccleaning, ?lclv) ^ EvaluatedNonDomesticBuilding(?endb) ^ CriteriaCleaning(?ccleaning) ^ swrlb:greaterThanOrEqual(?mrcleaning, ?lclv) ^ swrlb:lessThanOrEqual(?mrcleaning, ?uclv) -> hasResultCLEANING(?endb, Satisfactory)
Rule_Conditions in summer overall
swrlb:greaterThan(?mrconsover,?uclv)^ hasMeanResponseValueCONSOVER(?endb,?mrconsover) ^hasUpperCriticalLimitValue(?cconsover,?uclv)^CriteriaConditionsInSummerOverall(?cconsover) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultCONSOVER(?endb, Outstanding)
hasLowerCriticalLimitValue(?cconsover, ?lclv) ^ swrlb:lessThanOrEqual(?mrconsover, ?uclv) ^ hasMeanResponseValueCONSOVER(?endb,?mrconsover)^hasUpperCriticalLimitValue(?cconsover,?uclv)^CriteriaConditionsInSummerOverall(?cconsover)^swrlb:greaterThanOrEqual(?mrconsover, ?lclv) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultCONSOVER(?endb, Satisfactory)
EvaluatedNonDomesticBuilding(?endb) ^ hasLowerCriticalLimitValue(?cconsover,?lclv) ^ hasMeanResponseValueCONSOVER(?endb, ?mrconsover) ^ swrlb:lessThan(?mrconsover, ?lclv) ^ CriteriaConditionsInSummerOverall(?cconsover) -> hasResultCONSOVER(?endb, Unsatisfactory)
Rule_Effectiveness of space use

swrlb:greaterThan(?mrspacebuild,?uclv)^CriteriaSpaceUse(?cspacebuild)^hasUpperCriticalLimitValue(?cspacebuild, ?uclv) ^ hasMeanResponseValueSPACEBUILD(?endb, ?mrspacebuild) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultSPACEBUILD(?endb, Outstanding)
hasLowerCriticalLimitValue(?cspacebuild,?lclv)^CriteriaSpaceUse(?cspacebuild)^swrlb:lessThan(?mrspacebuild, ?lclv) ^ hasMeanResponseValueSPACEBUILD(?endb, ?mrspacebuild) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultSPACEBUILD(?endb, UsedIneffectively)
swrlb:lessThanOrEqual(?mrspacebuild, ?uclv) ^ hasLowerCriticalLimitValue(?cspacebuild, ?lclv) ^CriteriaSpaceUse(?cspacebuild)^hasUpperCriticalLimitValue(?cspacebuild,?uclv)^swrlb:greaterThanOrEqual(?mrspacebuild,?lclv) ^ hasMeanResponseValueSPACEBUILD(?endb, ?mrspacebuild) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultSPACEBUILD(?endb, UsedEffectively)
Rule_Humidity in summer / winter
CriteriaHumidityInSummer(?cshum)^swrlb:lessThan(?mrshum,?lclv)^hasMeanResponseValueSHUMIDITY(?endb, ?mrshum) ^ hasLowerCriticalLimitValue(?cshumidity, ?lclv) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultSHUMIDITY(?endb, TooDry)
CriteriaHumidityInSummer(?cshum) ^ hasMeanResponseValueSHUMIDITY(?endb, ?mrshum) ^ swrlb:greaterThan(?mrshum,?uclv)^ hasUpperCriticalLimitValue(?cshum, ?uclv) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultSHUMIDITY(?endb, TooHumid)
EvaluatedNonDomesticBuilding(?endb)^hasMeanResponseValueSHUMIDITY(?endb, ?mrshum)^ hasUpperCriticalLimitValue(?cshum, ?uclv) ^ CriteriaHumidityInSummer(?cshum) ^ swrlb:greaterThanOrEqual(?mrshum,?lclv)^hasMeanResponseValueSHUMIDITY(?endb,?mrshum)^hasLowerCriticalLimitValue(?cshum,?lclv)^ swrlb:lessThanOrEqual(?mrshum, ?uclv) -> hasResultSHUMIDITY(?endb, Satisfactory)
Rule_Image to visitors
hasMeanResponseValueIMAGE(?endb, ?mrimage) ^ hasUpperCriticalLimitValue(?cimage, ?uclv) ^ swrlb:greaterThan(?mrimage, ?uclv) ^ CriteriaImageToVisitors(?cimage) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultIMAGE(?endb, Outstanding)
hasLowerCriticalLimitValue(?cimage, ?lclv) ^ CriteriaImageToVisitors(?cimage) ^ swrlb:lessThan(?mrimage, ?lclv) ^ hasMeanResponseValueIMAGE(?endb, ?mrimage) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultIMAGE(?endb, Poor)
swrlb:lessThanOrEqual(?mrimage, ?uclv) ^ hasLowerCriticalLimitValue(?cimage, ?lclv) ^ CriteriaImageToVisitors(?cimage) ^ hasMeanResponseValueIMAGE(?endb, ?mrimage) ^ hasUpperCriticalLimitValue(?cimage, ?uclv) ^ swrlb:greaterThanOrEqual(?mrimage, ?lclv) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultIMAGE(?endb, Good)
Rule_Indoor air quality summer/winter
swrlb:greaterThan(?msiaq,?uclv)^ CriteriaAirQualityInSummer(?csiaq) ^ hasUpperCriticalLimitValue(?csiaq, ?uclv) ^ EvaluatedNonDomesticBuilding(?endb) ^ hasMeanResponseValueSIAQ(?endb, ?msiaq) -> hasResultSIAQ(?endb, TooSmelly)
hasLowerCriticalLimitValue(?csiaq, ?lclv) ^ swrlb:lessThanOrEqual(?msiaq, ?uclv) ^ swrlb:greaterThanOrEqual(?msiaq, ?lclv) ^ CriteriaAirQualityInSummer(?csiaq) ^ hasUpperCriticalLimitValue(?csiaq, ?uclv) ^ EvaluatedNonDomesticBuilding(?endb) ^ hasMeanResponseValueSIAQ(?endb, ?msiaq) -> hasResultSIAQ(?endb, Satisfactory)
hasLowerCriticalLimitValue(?csiaq, ?lclv) ^ swrlb:lessThan(?msiaq, ?lclv) ^ CriteriaAirQualityInSummer(?csiaq) ^ EvaluatedNonDomesticBuilding(?endb) ^ hasMeanResponseValueSIAQ(?endb, ?msiaq) -> hasResultSIAQ(?endb, Outstanding)
Rule_ Artificial light/Natural light

swrlb:lessThan(?mrltart, ?lclv) ^ EvaluatedNonDomesticBuilding(?endb) ^ CriteriaAmountOfArtificialLight(?cltart) ^ hasLowerCriticalLimitValue(?cltart, ?lclv) ^ hasMeanResponseValueLTART(?endb, ?mrltart) -> hasResultLTART(?endb, TooLittle)
EvaluatedNonDomesticBuilding(?endb) ^ hasUpperCriticalLimitValue(?cltart, ?uclv) ^ CriteriaAmountOfArtificialLight(?cltart) ^ swrlb:greaterThan(?mrltart, ?uclv) ^ hasMeanResponseValueLTART(?endb, ?mrltart) -> hasResultLTART(?endb, TooMuch)
swrlb:greaterThanOrEqual(?mrltart, ?lclv) ^ CriteriaAmountOfArtificialLight(?cltart) ^ hasUpperCriticalLimitValue(?cltart, ?uclv) ^ swrlb:lessThanOrEqual(?mrltart, ?uclv) ^ hasLowerCriticalLimitValue(?cltart, ?lclv) ^ hasMeanResponseValueLTART(?endb, ?mrltart) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultLTART(?endb, Satisfactory)
Rule_Lighting glare from sun and sky
hasLowerCriticalLimitValue(?cltnatngl,?lclv)^hasMeanResponseValueLTNATNGL(?endb, ?mrltnatngl) ^ swrlb:lessThan(?mrltnatngl, ?lclv) ^ CriteriaGlareFromSunAndSky(?cltnatngl) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultLTNATNGL(?endb, Outstanding)
swrlb:greaterThan(?mrltnatngl, ?uclv) ^ hasMeanResponseValueLTNATNGL(?endb, ?mrltnatngl) ^ hasUpperCriticalLimitValue(?cltnatngl, ?uclv) ^ CriteriaGlareFromSunAndSky(?cltnatngl) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultLTNATNGL(?endb, TooMuch)
swrlb:lessThanOrEqual(?mrltnatngl, ?uclv)^swrlb:greaterThanOrEqual(?mrltnatngl, ?lclv)^hasLowerCriticalLimitValue(?cltnatngl,?lclv)^ hasMeanResponseValueLTNATNGL(?endb, ?mrltnatngl) ^ hasUpperCriticalLimitValue(?cltnatngl, ?uclv) ^ CriteriaGlareFromSunAndSky(?cltnatngl) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultLTNATNGL(?endb, Satisfactory)
Rule_Perceived health
CriteriaPerceivedHealth(?chealth) ^ hasLowerCriticalLimitValue(?chealth, ?lclv) ^ hasMeanResponseValueHEALTH(?endb, ?mrhealth) ^ swrlb:lessThan(?mrhealth, ?lclv) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultHEALTH(?endb, LessHealthy)
CriteriaPerceivedHealth(?chealth) ^ hasLowerCriticalLimitValue(?chealth, ?lclv) ^ hasMeanResponseValueHEALTH(?endb, ?mrhealth) ^ swrlb:greaterThanOrEqual(?mrhealth, ?lclv) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultHEALTH(?endb, Healthy)
Rule_Personal control over cooling
hasLowerCriticalLimitValue(?ccntco, ?lclv) ^ swrlb:lessThan(?mrcntco, ?lclv) ^ EvaluatedNonDomesticBuilding(?endb) ^ CriteriaPersonalControlOverCooling(?ccntco) ^ hasMeanResponseValueCNTCO(?endb, ?mrcntco) -> hasResultCNTCO(?endb, NoControl)
swrlb:greaterThan(?mrcntco, ?lclv)^ hasLowerCriticalLimitValue(?ccntco, ?lclv) ^ EvaluatedNonDomesticBuilding(?endb)^CriteriaPersonalControlOverCooling(?ccntco)^hasMeanResponseValueCNTCO(?endb,?mrcntco)->hasResultCNTCO(?endb, SatisfactoryOnControl)
Rule_Personal safety
CriteriaPersonalSafety(?csafety)^hasLowerCriticalLimitValue(?csafety,?lclv)^swrlb:lessThan(?mrsafety, ?lclv) ^ hasMeanResponseValueSAFETY(?endb, ?mrsafety) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultSAFETY(?endb, Poor)
hasUpperCriticalLimitValue(?csafety, ?uclv) ^ CriteriaPersonalSafety(?csafety) ^ swrlb:greaterThan(?mrsafety, ?uclv) ^ hasMeanResponseValueSAFETY(?endb, ?mrsafety) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultSAFETY(?endb, Outstanding)
hasUpperCriticalLimitValue(?csafety, ?uclv) ^ CriteriaPersonalSafety(?csafety) ^ swrlb:greaterThanOrEqual(?mrsafety, ?lclv) ^ swrlb:lessThanOrEqual(?mrsafety, ?uclv) ^

hasLowerCriticalLimitValue(?csafety, ?lclv) ^ EvaluatedNonDomesticBuilding(?endb) ^ hasMeanResponseValueSAFETY(?endb, ?mrsafety) -> hasResultSAFETY(?endb, Good)
Rule_Productivity at work
CriteriaProductivityAtWork(?cprod) ^ hasMeanResponseValuePROD(?endb, ?mrprod) ^ EvaluatedNonDomesticBuilding(?endb) ^ hasUpperCriticalLimitValue(?cprod, ?uclv) ^ swrlb:greaterThan(?mrprod, ?uclv) -> hasResultPROD(?endb, MoreProductivity)
hasMeanResponseValuePROD(?endb,?mrprod)^hasLowerCriticalLimitValue(?cprod, ?lclv) ^ swrlb:lessThan(?mrprod, ?lclv) ^ CriteriaProductivityAtWork(?cprod) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultPROD(?endb, LessProductivity)
EvaluatedNonDomesticBuilding(?endb)^ hasMeanResponseValuePROD(?endb, ?mrprod) ^ hasUpperCriticalLimitValue(?cprod, ?uclv) ^ hasLowerCriticalLimitValue(?cprod, ?lclv) ^ swrlb:lessThanOrEqual(?mrprod, ?uclv) ^ swrlb:greaterThanOrEqual(?mrprod, ?lclv) ^ CriteriaProductivityAtWork(?cprod) -> hasResultPROD(?endb, NoDifference)
Rule_Space at work area
EvaluatedNonDomesticBuilding(?endb) ^ CriteriaAdequacyOfSpaceAtWorkArea(?cspacedesk) ^ hasMeanResponseValueSPACEDESK(?endb,?mrspacedesk)^swrlb:greaterThan(?mrspacedesk,?uclv)^hasUpperCriticalLimitValue(?cspacedesk, ?uclv)->hasResultSPACEDESK(?endb, TooMuch)
EvaluatedNonDomesticBuilding(?endb) ^ CriteriaAdequacyOfSpaceAtWorkArea(?cspacedesk) ^ hasMeanResponseValueSPACEDESK(?endb,?mrspacedesk)^ swrlb:lessThan(?mrspacedesk, ?lclv) ^ hasLowerCriticalLimitValue(?cspacedesk, ?lclv) -> hasResultSPACEDESK(?endb, TooLittle)
swrlb:lessThanOrEqual(?mrspacedesk,?uclv)^hasMeanResponseValueSPACEDESK(?endb,?mrsacedesk)^CriteriaAdequacyOfSpaceAtWorkArea(?cspacedesk)^swrlb:greaterThanOrEqual(?mrspacedesk,?lclv)^hasUpperCriticalLimitValue(?cspacedesk,?uclv)^EvaluatedNonDomesticBuilding(?endb)^hasLowerCriticalLimitValue(?cspacedesk,?lclv)->hasResultSPACEDESK(?endb,Satisfactory)
Rule_Storage arrangements
hasMeanResponseValueSTORAGE(?endb, ?mrstorage)^hasLowerCriticalLimitValue(?cstorage, ?lclv) ^ CriteriaStorageArrangements(?cstorage) ^ swrlb:lessThan(?mrstorage, ?lclv) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultSTORAGE(?endb, Unsatisfactory)
swrlb:greaterThan(?mrstorage, ?uclv) ^ hasMeanResponseValueSTORAGE(?endb, ?mrstorage) ^ CriteriaStorageArrangements(?cstorage) ^ hasUpperCriticalLimitValue(?cstorage, ?uclv) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultSTORAGE(?endb, Outstanding)
swrlb:lessThanOrEqual(?mrstorage,?uclv)^hasMeanResponseValueSTORAGE(?endb, ?mrstorage) ^ hasLowerCriticalLimitValue(?cstorage, ?lclv) ^ CriteriaStorageArrangements(?cstorage) ^ hasUpperCriticalLimitValue(?cstorage, ?uclv) ^ swrlb:greaterThanOrEqual(?mrstorage, ?lclv) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultSTORAGE(?endb, Satisfactory)
Rule_Temperature in summer/winter
hasMeanResponseValueTSHOT(?endb, ?mrtshot) ^ swrlb:lessThan(?mrtshot, ?lclv) ^ CriteriaTemperatureInSummer(?ctshot) ^ hasLowerCriticalLimitValue(?ctshot, ?lclv) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultTSHOT(?endb, TooHot)
hasMeanResponseValueTSHOT(?endb, ?mrtshot) ^ hasUpperCriticalLimitValue(?ctshot, ?uclv) ^ swrlb:greaterThan(?mrtshot, ?uclv) ^ CriteriaTemperatureInSummer(?ctshot) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultTSHOT(?endb, TooCold)
hasMeanResponseValueTSHOT(?endb, ?mrtshot) ^ hasUpperCriticalLimitValue(?ctshot, ?uclv) ^ swrlb:greaterThanOrEqual(?mrtshot, ?lclv) ^ CriteriaTemperatureInSummer(?ctshot) ^

hasLowerCriticalLimitValue(?ctshot, ?lclv) ^ swrlb:lessThanOrEqual(?mrtshot, ?uclv) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultTSHOT(?endb, Satisfactory)
Rule_Thermal comfort in summer/winter
EvaluatedNonDomesticBuilding(?endb) ^ hasMeanResponseValueTSOVER(?endb, ?mrtsover) ^ swrlb:lessThan(?mrtsover, ?lclv) ^ CriteriaThermalComfortInSummer(?ctsover) ^ hasLowerCriticalLimitValue(?ctsover, ?lclv) -> hasResultTSOVER(?endb, Uncomfortable)
CriteriaThermalComfortInSummer(?ctsover) ^ hasUpperCriticalLimitValue(?ctsover, ?uclv) ^ swrlb:greaterThan(?mrtsover, ?uclv) ^ hasMeanResponseValueTSOVER(?endb, ?mrtsover) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultTSOVER(?endb, Outstanding)
swrlb:greaterThanOrEqual(?mrtsover, ?lclv) ^ CriteriaThermalComfortInSummer(?ctsover) ^ hasUpperCriticalLimitValue(?ctsover, ?uclv) ^ hasLowerCriticalLimitValue(?ctsover, ?lclv) ^ hasMeanResponseValueTSOVER(?endb, ?mrtsover) ^ swrlb:lessThanOrEqual(?mrtsover, ?uclv) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultTSOVER(?endb, Comfortable)
Rule_Ventilation in summer/winter
swrlb:greaterThan(?mrsventilation?uclv)^hasMeanResponseValueSVENTILATION(?endb, ?mrsventilation)^CriteriaVentilationInSummer(?csventilation)^hasUpperCriticalLimitValue(?csventilation, ?uclv) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultSVENTILATION(?endb, Stuffy)
CriteriaVentilationInSummer(?csventilation) ^ EvaluatedNonDomesticBuilding(?endb) ^ swrlb:lessThan(?mrsventilation, ?lclv) ^ hasLowerCriticalLimitValue(?csventilation, ?lclv) ^ hasMeanResponseValueSVENTILATION(?endb, ?mrsventilation) -> hasResultSVENTILATION(?endb, OutstandingFresh)
swrlb:greaterThanOrEqual(?mrsventilation, ?lclv)^hasLowerCriticalLimitValue(?csventilation, ?lclv) ^ hasMeanResponseValueSVENTILATION(?endb, ?mrsventilation) ^ CriteriaVentilationInSummer(?csventilation) ^ hasUpperCriticalLimitValue(?csventilation, ?uclv) ^ swrlb:lessThanOrEqual(?mrsventilation, ?uclv) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultSVENTILATION(?endb, Satisfactory)
Rule_Response to requests speed
hasMeanResponseValueSPEED(?endb, ?mrspeed) ^CriteriaSpeedOfResponseToRequests(?cspeed) ^ swrlb:lessThan(?mrspeed, ?lclv) ^ hasLowerCriticalLimitValue(?cspeed, ?lclv) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultSPEED(?endb, TooSlow)
hasMeanResponseValueSPEED(?endb, ?mrspeed) ^CriteriaSpeedOfResponseToRequests(?cspeed) ^ hasUpperCriticalLimitValue(?cspeed, ?uclv) ^ swrlb:greaterThan(?mrspeed, ?uclv) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultSPEED(?endb, Outstanding)
hasMeanResponseValueSPEED(?endb, ?mrspeed) ^CriteriaSpeedOfResponseToRequests(?cspeed) ^ hasUpperCriticalLimitValue(?cspeed, ?uclv) ^ swrlb:lessThanOrEqual(?mrspeed, ?uclv) ^ hasLowerCriticalLimitValue(?cspeed, ?lclv) ^ swrlb:greaterThanOrEqual(?mrspeed, ?lclv) ^ EvaluatedNonDomesticBuilding(?endb) -> hasResultSPEED(?endb, Satisfactory)
Rule_RIR air movement in summer
hasMeanResponseValueAIRSMOVEMENT(?endb,?mrsairmove)^swrlb:subtract(?rgsairmove, ?bmsairmove, ?mrsairmove)^swrlb:abs(?argsairmove, ?rgsairmove)^EvaluatedNonDomesticBuilding(?endb)^CriteriaAirMovementInSummer(?csairmove)^hasBenchamrkMeanValue(?csairmove,?bmsairmove)^ swrlb:divide(?rirsairmove, ?argsairmove, ?mrsairmove) -> hasRIRAIRSMOVEMENT(?endb, ?rirsairmove)
Rule_RIR artificial light/Natural light

hasMeanResponseValueLTART(?endb, ?mrltart) ^ swrlb:subtract(?rgltart, ?bmltart, ?mrltart) ^ swrlb:abs(?argltart, ?rgltart) ^ CriteriaAmountOfArtificialLight(?cltart) ^ hasBenchamrkMeanValue(?cltart, ?bmltart) ^ EvaluatedNonDomesticBuilding(?endb) ^ swrlb:divide(?rirltart, ?argltart, ?mrltart) -> hasRIRLTART(?endb, ?rirltart)
Rule_RIR cleaning
hasMeanResponseValueCLEANING(?endb,?mrclean)^swrlb:subtract(?rgclean, ?bmclean, ?mrclean)^swrlb:abs(?argclean,?rgclean)^CriteriaCleaning(?cclean)^hasBenchamrkMeanValue(?cclean, ?bmclean) ^EvaluatedNonDomesticBuilding(?endb) ^ swrlb:divide(?rirclean,?argclean,?mrclean)-> hasRIRCLEANING(?endb, ?rirclean)
Rule_RIR effectiveness of response to requests
hasMeanResponseValueEFFECT(?endb,?mreffect)^swrlb:subtract(?rgeffect, ?bmeffect, ?mreffect) ^swrlb:abs(?argeffect,?rgeffect)^CriteriaEffectivenessOfResponseToRequests(?ceffect)^hasBenchamrkMeanValue(?ceffect,?bmeffect)^EvaluatedNonDomesticBuilding(?endb)^swrlb:divide(?rireffect,?argeffect,?mreffect)-> hasRIREFFECT(?endb, ?rireffect)
Rule_RIR humidity in summer/winter
hasMeanResponseValueSHUMIDITY(?endb,?mrshum)^swrlb:subtract(?rgshum,?bmshum,?mrshum)^swrlb:abs(?argshumidity, ?rgshumidity)^CriteriaHumidityInSummer(?cshum)^hasBenchamrkMeanValue(?cshum, ?bmshum)^EvaluatedNonDomesticBuilding(?endb)^swrlb:divide(?rirshum, ?argshum, ?mrshum) -> hasRIRSHUMIDITY(?endb, ?rirshumidity)
Rule_RIR IAQ in summer/winter
hasMeanResponseValueSIAQ(?endb, ?mrsiaq) ^ swrlb:subtract(?rgsiaq, ?bmsiaq, ?mrsiaq) ^ swrlb:abs(?argsiaq, ?rgsiaq) ^ CriteriaAirQualityInSummer(?csiaq) ^ hasBenchamrkMeanValue(?csiaq, ?bmsiaq) ^ EvaluatedNonDomesticBuilding(?endb) ^ swrlb:divide(?rirsiaq, ?argsiaq, ?mrsiaq) -> hasRIRSIAQ(?endb, ?rirsiaq)
Rule_RIR perceived health
hasMeanResponseValueHEALTH(?endb,?mrhealth)^swrlb:subtract(?rghealth, ?bmhealth, ?mrhealth)^swrlb:abs(?arghealth,?rghealth)^CriteriaPerceivedHealth(?chealth)^hasBenchamrkMeanValue(?chealth,?bmhealth)^EvaluatedNonDomesticBuilding(?endb)^swrlb:divide(?rirhealth,?arghealth,?mrhealth)-> hasRIRHEALTH(?endb, ?rirhealth)
Rule_RIR personal control cooling / heating / lighting / noise
hasMeanResponseValueCNTCO(?endb, ?mrcntco) ^ swrlb:subtract(?rgcntco, ?bmcntco, ?mrcntco) ^swrlb:abs(?argcntco,?rgcntco)^CriteriaPersonalControlOverCooling(?ccntco)^hasBenchamrkMeanValue(?ccntco,?bmcntco)^EvaluatedNonDomesticBuilding(?endb)^swrlb:divide(?rircntco,?argcntco,?mrcntco)->hasRIRCNTCO(?endb, ?rircntco)
Rule_RIR productivity
hasMeanResponseValuePROD(?endb, ?mrprod) ^ swrlb:subtract(?rgprod, ?bmprod, ?mrprod) ^ swrlb:abs(?argprod, ?rgprod) ^ CriteriaProductivityAtWork(?cprod) ^ hasBenchamrkMeanValue(?cprod, ?bmprod) ^ EvaluatedNonDomesticBuilding(?endb) ^ swrlb:divide(?rirprod, ?argprod, ?mrprod) -> hasRIRPROD(?endb, ?rirprod)
Rule_RIR meeting room availability
hasMeanResponseValueMEETING(?endb,?mrmeeting)^swrlb:subtract(?rgmeeting, ?bmmeeting, ?mrmeeting)^swrlb:abs(?argmeeting,?rgmeeting)^hasBenchamrkMeanValue(?cmeeting, ?bmmeeting) ^ EvaluatedNonDomesticBuilding(?endb) ^ CriteriaAvailabilityOfMeetingRooms(?cmeeting) ^ swrlb:divide(?rirmeeting, ?argmeeting, ?mrmeeting) -> hasRIRMEETING(?endb, ?rirmeeting)

Rule_RIR space at desk
hasMeanResponseValueSPACEDESK(?endb?mrspace)^swrlb:subtract(?rgspace,?bmspace, ?mrspace)^swrlb:abs(?argspace,?rgspace)^hasBenchamrkMeanValue(?cspace,?bmspace)^EvaluatedNonDomesticBuilding(?endb)^CriteriaAdequacyOfSpaceAtWorkArea(?cspace)^swrlb:divide(?rirspace,?argspace,?mrspace)->hasRIRSPACEDESK(?endb, ?rirspace)
Rule_RIR temperature in summer/winter
hasMeanResponseValueTSHOT(?endb, ?mrtshot) ^ hasBenchamrkMeanValue(?ctshot, ?bmtshot) ^ EvaluatedNonDomesticBuilding(?endb) ^ swrlb:subtract(?rgtshot, ?bmtshot, ?mrtshot) ^ swrlb:abs(?argtshot,?rgtshot)^CriteriaTemperatureInSummer(?ctshot)^swrlb:divide(?rirtshot,?argtshot,?mrtshot)-> hasRIRTSHOT(?endb, ?rirtshot)
Rule_RIR thermal comfort in summer/winter
hasMeanResponseValueTSOVER(?endb,?mrtsover)^hasBenchamrkMeanValue(?ctsover, ?bmtsover) ^ EvaluatedNonDomesticBuilding(?endb) ^ swrlb:subtract(?rgtsover, ?bmtsover, ?mrtsover) ^ swrlb:abs(?argtsover,?rgtsover)^CriteriaThermalComfortInSummer(?ctsover)^swrlb:divide(?rirtsover,?argtsover,?mrtsover) -> hasRIRTSOVER(?endb, ?rirtsover)
Rule_RIR ventilation in summer/winter
hasMeanResponseValueSVENTILATION(?endb,?mrsventilation)^swrlb:subtract(?rgsventilation,?bmsventilation,?mrsventilation)^swrlb:abs(?argsventilation,?rgsventilation)^CriteriaVentilationInSummer(?csventilation)^hasBenchamrkMeanValue(?csventilation,?bmsventilation)^EvaluatedNonDomesticBuilding(?endb) ^ swrlb:divide(?rirsventilation, ?argsventilation, ?mrsventilation) -> hasRIRSVENTILATION(?endb, ?rirsventilation)
SQWRL Queries Examples
Query_Select criteria lower & upper limit value
hasUpperCriticalLimitValue(?criteria, ?UpperCriticalLimitValue) ^ EvaluationCriteria(?criteria) ^ hasLowerCriticalLimitValue(?criteria, ?LowerCriticalLimitValue) -> sqwrl:select(?criteria, ?LowerCriticalLimitValue, ?UpperCriticalLimitValue)
Query_RIR overall in summer conditions
EvaluatedNonDomesticBuilding(?endb)^hasRelativeImprovementRatioSIAQ(?endb,?rirsiaq) ^ hasRIRTSHOT(?endb,?rirtshot)^hasRIRSHUMIDITY(?endb,?rirshum)^hasRIRSVENTILATION(?endb,?rirsventilation)^hasResultTSHOT(?endb, ?rtshot)^hasResultSIAQ(?endb, ?rsiaq) ^ hasResultSVENTILATION(?endb, ?rsventilation)^hasResultSHUMIDITY(?endb, ?rshum) ^ hasResultAIRSMOVEMENT(?endb, ?rsairmove)^hasRIRAIRSMOVEMENT(?endb,?rirsairmove) ->sqwrl:select(?endb, ?rtshot, ?rirtshot, ?rsiaq, ?rirsiaq, ?rshum, ?rirshum, ?rsventilation,?rirsventilation, ?rsairmove, ?rirsairmove)
Query_Assess IAQ need
hasResultSIAQ(?endb,?rsiaq)^hasResultAIRWMOVEMENT(?endb,?rairwmovement)^hasResultAIRSMOVEMENT(?endb,?rairsmovement)^hasResultWIAQ(?endb?rwiaq)^EvaluatedNonDomesticBuilding(?endb)->sqwrl:select(?endb, ?rsiaq, ?rwiaq, ?rairsmovement, ?rairwmovement)
Query_Assess thermal need
hasResultWHUMIDITY(?endb, ?rwhum) ^ hasResultWVENTILATION(?endb, ?rwventilation) ^ hasResultTSHOT(?endb, ?rtshot) ^ hasResultSVENTILATION(?endb, ?rsventilation) ^ EvaluatedNonDomesticBuilding(?endb) ^ hasResultSHUMIDITY(?endb, ?rshum) ^ hasResultTWHOT(?endb, ?rtwhot)->sqwrl:select(?endb, ?rtshot, ?rsventilation, ?rshum, ?rtwhot, ?rwventilation, ?rwhum)

Query_Assess winter overall condition
EvaluatedNonDomesticBuilding(?endb) ^ hasResultCONWOVER(?endb, ?rconwover) ^ hasResultTWOVER(?endb, ?rtwover) ^ hasResultWHUMIDITY(?endb, ?rwhumidity) ^ hasResultTWHOT(?endb, ?rtwhot) ^ hasResultWIAQ(?endb, ?rwiaq) -> sqwrl:select(?endb, ?rwhumidity, ?rtwhot, ?rwiaq, ?rconwover, ?rtwover)
Query_Select benchmark lower & upper limit value of evaluation criteria
EvaluationCriteria(?criteria) ^ hasBenchmarkUpperValue(?ec, ?BenchmarkUpperValue) ^ hasBenchmarkLowerValue(?ec, ?BenchmarkLowerValue) -> sqwrl:select(?criteria, ?BenchmarkLowerValue, ?BenchmarkUpperValue)
Query_Building system location and ID
EvaluatedBuilding(?eb) ^ hasSystem(?eb, ?system) ^ locatedAt(?system, ?room) ^ hasID(?system, ?id) -> sqwrl:select(?eb, ?system, ?room, ?id)
Query_Action in winter with system information
hasSetValueWinterTemperature(?endb, ?svwt) ^ EvaluatedNonDomesticBuilding(?endb) ^ hasResultTWHOT(?endb, ?rtwhot) ^ hasResultWHUMIDITY(?endb, ?rhum) ^ hasSetValueWinterHumidity(?endb, ?svwh) ^ hasActionWinterTemperature(?endb, ?whotaction) ^ hasActionWinterHumidity(?endb, ?whumidityaction) ^ hasSystemHeating(?endb, ?system) ^ locatedAt(?system,?location)^hasID(?system,?id)->sqwrl:select(?endb, ?svwt,?rtwhot,?whotaction, ?svwh, ?rhum, ?whumidityaction, ?system, ?id, ?location)
Query_Assess light conditions
hasResultLTNATNGL(?endb, ?rltnatngl) ^ hasResultLTNAT(?endb, ?rltnat) ^ hasResultLTOVER(?endb, ?ltover) ^ EvaluatedNonDomesticBuilding(?endb) -> sqwrl:select(?endb, ?rltnat, ?rltnatngl, ?ltover)

APPENDIX II POE Ontology Excerpt

<?xml version="1.0"?>

<rdf:RDF xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
xmlns:swrla="http://swrl.stanford.edu/ontologies/3.3/swrla.owl#"
xmlns:swrl="http://www.w3.org/2003/11/swrl#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:poe="http://www.semanticweb.org/poe#"
xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns:dc="http://purl.org/dc/elements/1.1/"
xml:base="http://www.semanticweb.org/PostOccupancyEvaluationOntology"
xmlns="http://www.semanticweb.org/PostOccupancyEvaluationOntology#">

<owl:Ontology
rdf:about="http://www.semanticweb.org/PostOccupancyEvaluationOntology">

<dc:creator xml:lang="en">Yuanhong Zhao</dc:creator>

<dc:title xml:lang="en">Post Occupancy Evaluaiton Ontology</dc:title>

<rdfs:comment xml:lang="en">This is a post-occupancy evaluaiton ontolgoy model witht the focus on occupants satisfaction assessment.</rdfs:comment>

<owl:versionInfo

rdf:datatype="http://www.w3.org/2000/01/rdf-schema#Literal">1.0.0</owl:versionInfo>