University of Nottingham Malaysia

First Year Annual Review

MPhil Civil Engineering

Using a knowledge graph to semantically enrich the learning state-space of a deep Q-learning HVAC controller towards energy efficient regulation of cooling load in office buildings

Author:
Kevin Luwemba Mugumya

Supervisor:
Dr. Wong Jing Ying

Co-Supervisor:
Prof. Andy CHAN

Co-Supervisor:
Dr. Chun-Chieh Yip

October 8, 2019

Abstract

In recent years, buildings have increasingly become very complex environments for facility managers to monitor and control in terms of their energy performance primarily, due to the corresponding growth in heterogeneity within building information and monitoring systems. In South East Asia, the autonomous heating, ventilation and air conditioning (HVAC) control systems currently installed in the thermal spaces of many buildings fail to efficiently rationalize the cooling demand while adapting to the stochastic indoor and outdoor building environment. This is evident with the unnecessary and excessive cooling of thermal zones even when the desired thermostat set-points have already been reached causing unintended occupant discomfort at the expense of increased energy use. For self-learning control systems, this state of redundancy can be attributed to how they ingest the heterogeneous building information that they use to learn. Because the energy performance of HVAC systems is affected by various heterogeneous building parameters, a holistic picture of this information is necessary for control systems to achieve optimality via end-to-end learning. Of course, a certain degree of such homogeneity in building data has already been realized with the advent of 'Building Information Modeling (BIM)' and the 'Industry Foundation Classes (IFC)' which provide a vendor-neutral and coherent way of representing and exchanging digitized building information. In fact, majority of the research community has rapidly evolved to embrace this model-centric workflow in developing a variety of HVAC control systems that monitor the behaviour of a building and autonomously rationalize the zonal thermal demands effectively (*smart energy-efficient buildings*). However, the underlying structure of the current ecosystem of BIM software still lacks formal explicit and context-aware semantics that are needed to represent a heterogeneous state-space in which an HVAC control algorithm can learn to take more context-aware actions using the latent parametric relationships between occupancy behaviour, sensory data, changes in weather and the characteristics of the building envelope. Towards addressing the above problem, this research intends to replace conventional feature vectors with a *Semantic Web Knowledge Graph* as the data model for representing the state-space of a deep reinforcement learning (DRL) agent in an HVAC control system. An extensive review of past research in this domain has made it evident that very little work has been done to assess how well a knowledge graph performs in a DRL setting of HVAC controllers towards being more energy efficient via end-to-end learning. This research process starts with the development of a novel modular ontology, *'ERLO-Energy Reinforcement Learning Ontology'* through extensions and alignments with already existing modular web ontologies within the AEC domain (BOT,

SSN, SOSA, OPM, BPO). Explicitly from these ontologies, ERLO will adopt specific domain concepts about the heating and cooling load of a building (building topology, envelope details, sensors, actuators, HVAC system controls, geo-location, shading, occupancy behaviour and scheduling). Using 3^{rd} order tensor representation, ERLO is used to develop the input state-space for the DRL agent modelled using the Markov Decision Process. The resulting state-space is updated periodically at defined time-steps using a *Building Controls Virtual Test Bed* BCVTB co-simulation environment in which an EnergyPlus experiment with an accurate building energy model (environment) is embedded with real weather data together with Simulink to implement the HVAC control actions learned by the DRL agent through experience. Finally, an Arduino micro-controller is used to deploy, test and validate the algorithm control performance in a real physical HVAC system within an office building.

Keywords: Ontologies, Linked Data, IFC, deep learning, energy efficiency, machine learning

Outline

The structure of this write-up aims to be self-explanatory and easy to understand for anyone unfamiliar and getting acquainted with BIM and Linked Data principles in the energy domain context.
To this effect, **Chapter 1** briefly introduces the background to the above principles with context to the research problem and a presentation of the corresponding motivation. This is followed by the objectives, scope and expected deliverables of this research.

Chapter 2 provides a detailed literature review on the current trends of linking heterogeneous building information using both BIM and semantic web technologies. Also, a comprehensive overview is provided on their underlying data modelling structures with a critical appraisal on their efficacy in modelling building energy information. This is followed by a review on reinforcement learning and how knowledge graphs can fit into the learning process. A research track is provided on the use of deep reinforcement learning for energy management. This chapter concludes with a justification for developing the research argument.

Having provided an extensive background and literature review putting the research problem in context, **Chapter 3** presents the detailed methodology for carrying out the work. The workflow and tools for the ontology development, preparation of the co-simulation environment and the development and training of the DRL algorithm are presented here. This section is concluded by the validation strategy and the prototype to be used for deploying and testing the algorithm in a real-life building environment. A summarized discussion is also provided on the research progress specifically providing highlights to training activities, conferences and workshops attended. Finally, the projected upcoming 2-year research plan is laid out in tabular format including journal publication targets.

Publications

1. The role of linked building data in aligning augmented reality (AR) with sustainable construction.
   ('Best paper award' Conference Proceedings of the 20th World Conference on Applied Science, Engineering and technology 26th June 2019)

2. The indoor built environment response to haze in Malaysia.
   (Accepted abstract to be presented and published in the conference proceedings for the 15th International Conference on Atmospheric Sciences and Applications to Air Quality - ASAAQ15 28th - 30th Oct 2019)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

The framing of this research is within ongoing efforts to semantically enrich heterogeneous building information into formats that can smartly be ingested by self-learning building control systems towards improving the energy performance of buildings.

With majority of people spending 90% of their daily lives indoors, buildings have become the largest consumers of global energy due to heavy reliance on heating and air conditioning to maintain acceptable indoor air quality and thermal comfort. In the United States, buildings alone account for more than 40% of the total energy consumption with 50% of that responsible for heating and cooling while China experienced a 40% increase in building energy use between 1990 and 2009 making it the second largest building energy consumer after the U.S (Cao et al., 2016). It is with no doubt that the building industry has continued to put pressure on the sustainability equilibrium of the natural environment with increased indirect emissions while being powered by unsustainable energy sources like fossil fuels, biomass and coal. Most notably, 2018 recorded extremely high temperatures and prolonged heat waves in many countries driving up the energy demand for air conditioning which made buildings responsible for 28% of the world's energy-related CO_2 emissions (Dulac et al., 2019).

Global energy efficiency policies have not evolved to keep up with the rapid growth in emerging economies which is evident with the recent fall in the speed of energy intensity reductions within the building sector (2% in 2015 to 0.6% in 2018) (Dulac et al., 2019). A good example to shed light on this is the unnecessary and excessive heating or cooling of thermal zones in many current buildings even if the desired thermostat setpoints have already been reached causing unintended occupant discomfort and increased energy use. Nevertheless, this trend can slowly be reversed if HVAC control systems are designed to be dynamic and adaptive to the stochastic building environment. However, designing such *smart buildings* with adaptive energy control policies is a finicky process that requires an exhaustive amount of thought and care, especially because of the extreme complexity in identifying the uncertain dependencies between the various dynamic and inter-domain impact factors like occupancy behaviour, envelope infiltration rates, air flows patterns, shading, equipment schedules, variable outdoor-indoor temperatures, not forgetting the ever-changing weather patterns (Curry et al., 2012). Because the desired solution aims to ***reduce energy consumption*** while ***maintaining thermal comfort*** and ***acceptable indoor air quality***, qualifies this a ***multi-objective optimization problem*** that requires careful trade-off analysis between the three objectives.

1.2 Research motivation

One solution that has been widely adopted for multi-objective problems is machine learning specifically using ***deep learning*** and ***reinforcement learning*** models that can directly ingest data in its most raw form to extract the most useful features and find complex patterns in the data automatically. However, these machine learning methods are all ***domain-specific*** and tailored to ingest data of the same format for example sound, text, images or videos which is not the case with the building domain which is highly heterogeneous making it necessary to have data model suited to represent such knowledge before machine learning methods can make proper use of it.

To an extent, ***Building Information Modelling*** (BIM) can serve in this role as an intelligent process to effectively handle large amounts of building information centrally within a three-dimensional asset model however, a bottleneck still lies on the critical path of sharing this model information within and outside the construction industry making it hard for other domains to become part of the BIM story (Borrmann et al., 2018; Jeroen et al., 2018; Pauwels et al., 2017b). Literature has attributed this performance bottleneck to the schema design of BIM's data-exchange model, Industry Foundation Classes (IFC) (Barbau et al., 2012; Beetz et al., 2009; El-Mekawy, 2010; Gómez-Romero et al., 2015; Kris et al., 2016). Until recently, the IFC schema was only available in the native EXPRESS format which is incomprehensible, unfamiliar and unusable in other domains outside the Architecture Engineering and Construction (AEC) industry (e.g building automation, sensors and monitoring, Geo-spatial, heritage and facility management) yet majority of the optimization problems within this industry require a homogeneous picture of such inter-domain information. (Pauwels and Terkaj, 2016; Pauwels and Roxin, 2016).

Meanwhile, independent of IFC and outside the AEC industry, other powerful knowledge-representation techniques are trending with various disciplines able to interlink their heterogeneous datasets using principles of the world wide web (Berners-Lee et al., 2001b; Berners-Lee, 2003, 2006). This interlinked web of data (knowledge graph) is idealized as the ***Semantic Web*** and is encoded using the ***Resource Description Framework*** (RDF), a universal data modelling language (Manola et al., 2014). There has been a growing research trend to translate IFC building information into RDF that is extensible and open to other domains (Barbau et al., 2012; Beetz et al., 2009; Pauwels and Roxin, 2016; Pauwels and Terkaj, 2016).

By taking advantage of this semantic enrichment that RDF offers to siloed building information in BIM models, this research argues that the resulting homogeneous knowledge graph can be used to achieve ***end-to-end learning*** in building control systems towards better energy performance.

End-to-end learning is a process in which all of the parameters are trained on the learning agent jointly, rather than step by step in isolation. This is of significant importance in machine learning which does not exactly solve a problem but rather provide an approximated solution and since the learning process begins by breaking down a complex problem into sub-problems that can be solved individually and then chained back together, the learning errors from sub problems can easily compound in the combined solution rendering it useless. The best strategy is to train the system end-to-end by holistically optimizing all parameters together rather than in isolation.

1.3 Problem statement

The autonomous HVAC control systems currently installed in the thermal spaces of many buildings fail to efficiently rationalize the cooling demand while adapting to the stochastic indoor and outdoor building environment. This is evident with the unnecessary and excessive cooling of thermal zones when the desired thermostat set-points have already been reached causing unintended occupant discomfort and increased energy use. For self-learning control systems, this state of redundancy can be attributed to how they ingest the heterogeneous building data that they use to learn. The deep Q-learning model is the most commonly adopted framework for autonomous building energy management and can achieve state-of-the-art performance in developing optimal control strategies when the input heterogeneous building environment is split up (pre-processed) into siloed constituents for learning. In doing so however, a lot of important existential relationships between the heterogeneous parameters are lost after pre-processing and end-to-end learning made almost impossible to achieve. A lot of ongoing research has been done by the semantic web community especially the Linked Building Data Community Group (LBDCG) to semantically enrich heterogeneous building information towards homogeneity using extensible and modular domain ontologies. However, there has been little work exploring how machine learning methods for building control systems can holistically consume heterogeneous data within such knowledge graphs to achieve end-to-end learning.

1.4 Aim

To investigate the effectiveness of using a knowledge graph as the ingestible data format for a state-space of an end-to-end deep reinforcement learning (DRL)-based HVAC control system towards minimizing the energy redundancies associated with cooling in office buildings.

1.5 Objectives

To achieve the above aim, the research objectives below have to be fulfilled.

1. To assess the level of semantic enrichment that RDF avails to a building energy model while encapsulating concepts about the heating and cooling load of an office building.
2. To identify the potential that an RDF knowledge graph provides in achieving end-to-end learning within the deep reinforcement learning architecture of an HVAC controlling agent towards more adaptive and optimal control of the cooling load in an office building.
3. To examine and quantify the cooling energy saving potential within a typical office building achieved by the aforementioned self-learning HVAC control system while comparing the results with conventional rule-based systems.

1.6 Research scope

1.6.1 The building energy demand components

This research is focusing specifically on the electric energy required to ***cool*** spaces within ***large office buildings*** and choosing to ignore the heating demand because the climatic zone in which the study is being conducted is characterized by hot and humid weather (South East Asia).

1.6.2 The HVAC technology in consideration.

The HVAC technology being considered for optimization is the ***Variable Refrigerant Flow (VRF)*** system which uses refrigerant as the heating and cooling medium. The exact mechanical and electrical components are not of significant importance to this research but rather the ***operation schematics***. This research aims to integrate VRFs with a smart controller capable of monitoring and rationalizing the cooling demand within building thermal zones while improving iteratively via end-to-end learning.

1.6.3 Machine learning process

Because HVAC optimization is a multi-objective problem, it fits well under the criterion of ***Reinforcement Learning*** (RL) which involves an agent autonomously learning how to behave through

sequential decision making using experience from actions taken in an environment. This learning process is further supplemented by ***neural networks***[1] that can deal with exceedingly large state spaces like buildings through a process called ***deep reinforcement learning***. The learning problem is explicitly modelled as a ***Markov Decision Process*** (MDP) whose solution is approximated using ***deep Q-learning*** but instead of using feature vectors as input, RDF graphs represented as third-order tensors will be adopted.

1.6.4 Learning Parameters

The cooling demand in a building thermal zone depends on several parameters but only the most important ones are chosen for this study namely;

- Occupancy behaviour in the different building zones and occupant number at different times of the day.
- Weather data (outdoor temperature, humidity and season of the year)
- Indoor environment characteristics (indoor temperature, indoor humidity, Carbon dioxide buildup depending on occupant number and spatial location on cooling unit)
- Building envelope geometry, thermal insulation properties of spaces and infiltration characteristics of the different thermal zones.
- Building material types (glass, wood, metal) and their physical properties (thermal conductivity and Ultra violet Index)
- Indoor monitoring sensors (occupancy, temperature, humidity and Carbon dioxide)
- The high-level operation schematics of the VRF HVAC system.

1.6.5 Knowledge graph encapsulating the learning parameters

To keep in alignment with the World Wide Web Consortium (W3C)'s goal of reusing and extending already existing knowledge graphs (ontologies), this research is developing a knowledge graph from already existing domain ontologies to encapsulate the learning concepts presented in subsection 1.6.4 namely;

- Building Ontology Topology (BOT) for defining relationships between the sub-components of a building (site, building, storey, space and element) using zone hierarchy.
- Semantic Sensor Ontology (SSN) and Sensor, Observation, Sample and Actuator Ontology (SOSA) for describing any sensors in the building, their observations and actuators. These ontologies together extend the BOT ontology to represent sensory information.

[1]Neural networks are a set of algorithms, modeled loosely after the human brain and are designed to recognize patterns in data.

- Ontology for Property management (OPM) to model properties that evolve over time. This ontology is used to extend SSN and SOSA to model sensory time-series information using RDF.
- Building Product Ontology (BPO) to describe information about building elements like HVAC equipment without attaching unnecessary geometry.
- File Ontology for Geometry formats (FOG) is employed with BPO in case geometry descriptions are required.

1.7 Significance of the research

This research is significant in terms of its theoretical and practical applications in extending the boundary of knowledge within the building automation domain. Theoretically, the findings of this study will provide a better understanding of how self-learning building control systems can holistically consume heterogeneous building information in knowledge graphs to minimize the energy redundancies of cooling systems when the thermal comfort and indoor air quality targets have been reached in thermal zones. By developing and testing a novel modular knowledge graph (ERLO) with a reinforcement learning model, a critical appraisal will be provided on how much semantic expressivity is required to achieve end-to-end learning from heterogeneous building data within a knowledge graph. Practically, this research avails facility managers and designers the means to develop more context-aware building controllers that can adapt continuously to the stochastic building environment while utilizing a semantic layer that partially overcomes the conventional black box approach of many current learning systems towards improved maintainability.

1.8 Limitations

- This research is solely focusing on optimizing building control systems for cooling demand only and assumes that the same control policies can be adopted for optimally heating thermal spaces using reverse schematics.
- A building energy simulation in EnergyPlus is used to test different HVAC system control configurations which even after calibration using Bayesian methods, might still carry errors that propagate with learning.
- The core mechanics of the VRF HVAC system are not considered in the optimization process of the automated controller but rather the high-level operation schematics as the former are assumed to be unimportant and out of scope for this study.
- The deep Q learning framework adopted for this research uses a function approximator that only provides estimates and approximations of the most optimal control actions given certain states of the building environment.

1.9 Summarized methodology

The research process starts with the development a novel modular knowledge graph (ontology), ***'ERLO-Energy Reinforcement Learning Ontology'*** that encapsulates specific domain concepts about the cooling load of a building (building topology, indoor envelope details, sensors, actuators, VRF system control schema, Geo-location, shading, occupancy behaviour and infiltration). Specifically, ERLO is achieved by extending, matching and integrating the already existing domain ontologies relevant to the aforementioned concepts namely; Building Topology Ontology-**BOT**, Semantic Sensor Network Ontology-**SSN**, Sensors, Observations, Samples and Actuators Ontology-**SOSA**, Ontology for Property Management-**OPM** and the Building Product Ontology-**BPO**. The resulting RDF knowledge graph, ERLO, is translated into a third-order tensor (2 orders for the subject and object and 1 order for the predicate) that replaces the conventional input feature vector for the deep Q-learning framework. The environment for the deep-Q network is an EnergyPlus experiment of a building energy model loosely coupled with simulink which can directly apply control actions learned by the agent in the environment. This process is iterative and continuous until satisfactory algorithm performance is achieved. Using an arduino micro-controller, the DRL agent is deployed, tested and validated in a real working physical HVAC unit and the results compared to those of a rule-based system for quantification of any energy saving potential.

1.10 Research Questions

This section presents the research questions that drive this research how they are related to each other, and the high-level process taken to address them.

Question 1: *Can semantic web technologies be used to provide an interoperability layer within the heterogeneous parameters encapsulated by a building energy model?*

To address this question, chapter 2 starts by exploring the current trends of linking and exchanging building information using the BIM process (section 2.3) while adopting IFC as the data exchange model (subsection 2.3.1). The limitations associated with IFC data representations are briefly introduced in subsubsection 2.3.1.2 and explained to greater detail in subsection 2.3.2 while focusing on the schema design and structure of the underlying data model for exchanging building information within BIM models and related software. Section 2.4 goes ahead to discuss the evolution of semantic web technologies and the RDF open data model (subsection 2.4.1) as a solution for overcoming the challenges of IFC by providing a semantic layer for heterogeneous building information within Building Information Models (section 2.5 and subsection 2.5.4). This review forms the basis for developing ERLO (section 3.2) that encapsulates concepts about the cooling demand in the thermal zones of a building.

Question 2: *Can building information in ontological knowledge graphs support the end-to-end learning process of HVAC smart controllers via deep reinforcement learning (DRL) towards improved energy performance?*

This question is addressed by first providing a review on the reinforcement learning (RL) process (subsection 2.6.2) and the modelling of a typical RL problem as a Markov Decision Process in (subsubsection 2.6.2.1) which is followed by a comparison of model-free and model-based RL learning methods in subsubsection 2.6.2.3. The most commonly used RL value approximation method (Q-learning) is introduced in subsubsection 2.6.2.4 while providing an appraisal of its shortfalls in representing large state spaces followed by a review on the combination of RL with function approximators (neural networks) subsubsection 2.6.3.1 that can work well with large state spaces like buildings environments (subsection 2.6.3). The current approaches of utilizing a knowledge graph as a default data model for machine learning are reviewed in subsection 2.6.4 while discussing the potential benefits for learning on data in this kind of format. Finally, the current applications of reinforcement learning in building energy management are reviewed and the potential for knowledge graph implementation identified in subsection 2.6.5.

Question 3: *Can information about algorithmic building control actions be stored in a formalized ontological fashion for reuse and extension to other domains from which valuable insight can be achieved by experts to further improve the performance of the control system?*

Having reviewed the potential applicability of the RDF open data model (subsection 2.4.1) to represent any kind of building information while using Unique Resource Identifiers (URI) subsection 2.4.2 for disambiguation, ERLO has the potential to encapsulate concepts about the algorithmic actions taken by the HVAC controlling agent for inference in the knowledge graph and queried using a SPARQL Endpoint (subsection 2.4.5) by other domain experts to understand how the controller is taking its actions and what relationships it is using to draw conclusions which is not the case with many conventional machine learning systems that adopt a black-box approach. This research will develop and test a conceptualization of this stature inside ERLO with an appraisal provided on the potential benefits and limitations of having a schema that can encode information about algorithmic building control actions.

Question 4: *What is the maintainability of such an end-to-end learning building control system in terms of computational cost and learning time?*

Because machine learning approaches are about solution approximation rather than finding exact solutions, an easily maintainable and understandable system is key in identifying and troubleshooting any errors by facility managers and building owners who might be inexperienced in the core concepts of machine learning. ***Question 3*** partly answers this by the mere fact that this system is adopting an easily understandable knowledge graph as the default data model for learning. The notion of understandability here is mainly for humans however, ***this research aims to check if the same is true for machine understandability during the learning process and the corresponding effect this has on computational cost***.

Question 5: *Does such an end-to-end HVAC control system achieve improved energy performance via its control policies?*

Upon developing the control system prototype, its control actions will be reviewed within a Building Control Virtual Test Bed (BCVTB) in which EnergyPlus and Simulink are loosely coupled. EnergyPlus can model with short time-steps the dynamic physical phenomena and environment parameter variations in a representative office building. On the other hand, Simulink gives the flexibility for integrating a complex control algorithm with the occupants' thermal sensation model which together with EnergyPlus are used to choose a heating or cooling set-point temperature and energy consumption estimated. Furthermore, the algorithmic source code will be deployed on an ***arduino micro-controller*** and tested on a real HVAC unit from which any ***energy saving potentials will be quantified in comparison to rule-based systems***.

1.11 Expected deliverables

1. A building control algorithm in the form of python source code that can be deployed in actuator systems to autonomously monitor and optimize a building's energy performance (heating and cooling) by taking control actions and continuously learning from them to improve with minimal supervision.

2. A schematic template for preparing a co-simulation environment that is used to update the state-space of a deep Q-learning HVAC control algorithm.

3. A turtle file of the Energy Reinforcement Learning Ontology (ERLO).

4. A GUI interface within ShareBIM capable of performing HVAC energy data analytics and visualization while using the control system.

Chapter 2

Literature review

2.1 Introduction

Optimization of building energy has received a significant amount of research interest in recent years with the core emphasis on the development of autonomous building energy management systems aimed at balancing thermal comfort, acceptable air quality and reduced energy use in a heuristic fashion (Chen et al., 2018b). Advancements in this area require the exploration of new knowledge-representation techniques for heterogeneous building data sets (Pauwels et al., 2018) in a linked manner that satisfies the information requirements of autonomous energy optimization. Such information includes but is not limited to building sensor data, device communication protocols, HVAC system models, occupancy behaviour data, Geo-location, weather data and simulation zonal models. This review therefore explores the state of the art with respect to Linked Data Concepts and data exchange models for autonomous building energy management with core emphasis on the use of deep reinforcement learning algorithms.

2.2 The need for linked data in the AEC/FM industry

The AEC/FM (Facility Management) industry is underpinned by a continuous flow and exchange of information during the design, construction and maintenance of the built environment (Borrmann et al., 2018). This information is usually fragmented and domain specific due to the complex and departmental nature of the industry making reliable exchange and stakeholder collaboration a challenge (Pauwels et al., 2018). Furthermore, this fragmentation hinders the integration of expert knowledge among designers, contractors and facility managers diminishing their opportunity to optimally influence the design, construction and management of a built asset. Mohd Nawi et al. (2014) investigated the fragmentation issues of the construction industry in detail and highlighted the resulting implications on project cost, schedule, dispute handling and unsustainable design-build routines. An increasing number of design optimization strategies in the AEC/FM industry need to work with heterogeneous building information generated from various data islands and often existing in unrelated formats. Such information is ineffective if utilized in unintegrated formats and this has led to extensive research efforts in integrating siloed building information with the advent of Building Information Modelling (BIM) (Borrmann et al., 2018; Pauwels et al., 2018).

2.3 Building Information Modelling

The process architecture of the AEC/FM industry has been evolving to embrace the power of digital tools in the design, construction and maintenance of the built environment. Adoption of digitized building information to replace paper-based approaches is an effective strategy towards the realization of Linked Building Data (Jeroen et al., 2018). Current approaches of generating, propagating and exchanging information on construction projects typically involve the handover of technical drawings in form of vertical sections, views and detail drawings which are incomprehensible to several computational methods like simulations, clash detections and consistency checks (Borrmann et al., 2018). Due to the increasingly complex nature of construction processes and with the aforementioned workflow, design changes quickly become a massive source of construction errors, escalated project costs and delays when not tracked and relayed on all related drawings. Furthermore, the semantic richness of such non-digital and siloed information is insufficient to support many heuristic stages of the building life-cycle for example energy analysis and indoor environment simulations (Zhang et al., 2015b), HVAC optimization processes (Chen et al., 2018b; Lu et al., 2019a) and autonomous building energy control (Mason and Grijalva, 2019). This is where Building Information Modelling (BIM) comes into play as a workflow that effectively handles vast amounts of building information by utilizing intelligent 3D model-based processes underpinned by computer technology to provide AEC professionals with the insight and tools to more efficiently plan, design, construct and manage buildings/ infrastructure (Borrmann et al., 2018). The information management protocols offered by BIM dramatically improve the coordination of complex design activities, semantic enrichment of simulations models for training autonomous energy control algorithms (Mason and Grijalva, 2019) and data-driven optimization of asset designs (Lu et al., 2019b). Furthermore, this model-centric workflow reduces manual re-entering of data along the project life-cycle which minimizes costly errors, clashes and data loss as shown in figure 2.1. Today a wide range of BIM software tools exist for geometric design, simulations, HVAC analysis, visualization etc.

To allow a seamless exchange of data between these software requires a vendor-neutral and standardized data exchange format with embedded rules about the semantic representation of asset information, existing hierarchical relationships and loss-free data exchange protocols (Borrmann et al., 2018). A brief overview of BIM's underlying data exchange structure, Industry Foundation Classes (IFC) is provided in subsection 2.3.1 below.

2.3.1 Industry Foundation Classes

As delineated in the section 2.3, the overarching goal of BIM is to solve the heterogeneity issues within the fragmented AEC/FM industry via lossless data exchange protocols. These should be embedded with explicitly defined and standardized semantic rules that are not open to misinterpretation (Pauwels et al., 2018). To this effect, the international organization buildingSMART has progressively developed the Industry Foundation Classes (IFC) as an open, vendor-neutral data exchange format/ schema to support almost any BIM data exchange use-case (Kris et al., 2016) along the building life-cycle whilst adopting an object-oriented approach i.e. the building is broken down into its constituent elements and spaces with well defined hierarchical inter-relationships. Chapter 3 of Borrmann et al. (2018) can be visited for a more elaborate description of object-oriented principles.

Modern BIM systems are able to generate semantically rich representations of buildings in the form of building information models that encapsulate, organize and relate building information in both human and machine-readable format (NBIMS, 2007). IFC on the other hand, adds a common language for the exchange of this model information between different BIM applications in a lossless fashion (see figure 2.2) eliminating the need to manually re-model the same building information during different use cases. IFC can only be used in practice once the software vendor implements it in their underlying import-export structure. Due to the complex and extensive nature of this data model, it is structured into four conceptual layers: Resource, Core, Interoperability, and Domain to improve its maintainability as discussed in chapter 5 of (Borrmann et al., 2018) (see figure 2.3). The aforementioned complexity arises from the generic nature of IFC and it is not uncommon for some software import-export routines to exercise data loss and errors during implementation (Borrmann et al., 2018). In fact Zhang et al. (2015b) highlights how IFC's generality results in the lack of several problem-specific constraints and Kris et al. (2016) delineates how IFC does not cover all data structures to meet the requirements of specific energy-management use cases. BuildingSMART has solved this hurdle by additional development of Model View Definitions (MVD) which map the rules that explicitly define which parts of the IFC data model need to be implemented

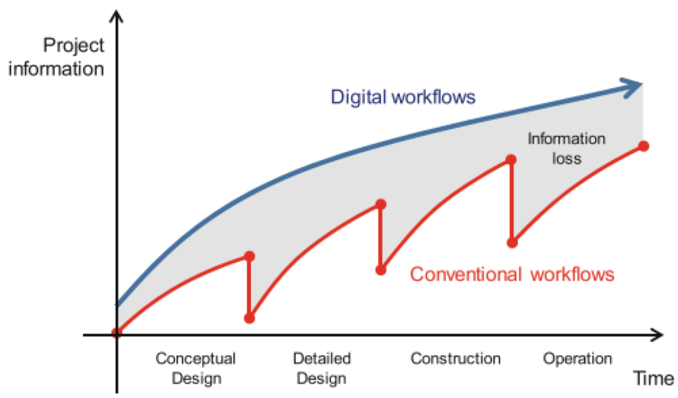

Figure 2.1: Information loss at various stages of the project lifecycle (Borrmann et al., 2018)

for a specific data exchange scenario (Chipman et al., 2016; Wix and Karlshøj, 2010; Zhang et al., 2014). These exchange requirements are first captured using Information Delivery Manuals (IDM) in tabular human-readable form and thereafter translated into a machine-readable format using MVDs before implementation (Chuck et al., 2011). The IFC implementations in BIM software are tested against MVDs as part of the BIM certification process however, software products are not certified for the entire data schema but only for specifically defined sections. A more comprehensive overview of IDM/MVD definitions will be provided in subsection 2.3.2, but first, a brief introduction to EXPRESS, the data modelling language for the IFC standard.

2.3.1.1 EXPRESS: The IFC data modelling language

The IFC data exchange model (also known as schema) is underpinned by technologies from EXPRESS (ISO 10303-11, 2004), an object-oriented data modeling language specifically designed for product modelling. It is based on part 11 of the family of ISO 10303 standards referred to as the International Standard for Exchange of Product Data (STEP) (Pratt, 2001). EXPRESS allows unambiguous product data definition and specification of constraints which makes it a data specification language and not a programming language (ISO 10303-11, 2004). Two overarching steps are involved while creating the IFC schema namely;

1. Specification of the data model using the EXPRESS language (ISO 10303-11, 2004).
2. Description, serialization and exchange of the concrete model instances using the STEP Physical File format (SPF) (ISO 10303-21, 2016).

It is important to remember that it is not possible to instantiate the data model using EXPRESS but rather using the STEP Physical File format defined in part 21 of ISO 10303 (ISO 10303-21, 2016). A detailed description of the EXPRESS structure with regards to IFC is provided by ISO 10303-11 (2004); Pauwels and Terkaj (2016) and only a summary of the fundamental overarching aspects is provided below.

1. EXPRESS relies on the abstraction of real-world objects into classes (called entities or entity types in EXPRESS). *'Objects'* are therefore instances/qualified members of *'entity types'*/classes.
2. Attributes and relationships can be defined for each entity type and used to implement the concept of inheritance i.e. parent class properties and relationships automatically apply to its subclasses.

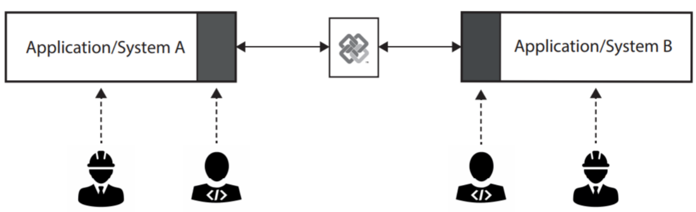

Figure 2.2: IFC exchange (which relies on end-users modelling expertise) between two BIM software via import-export routines implemented by software developers (Zhang, 2019).

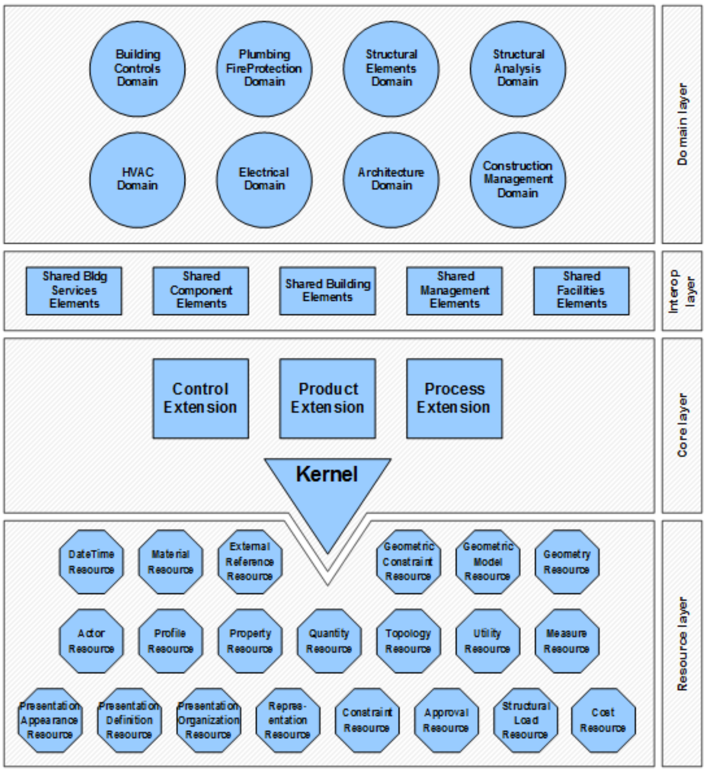

Figure 2.3: Conceptual layers of the IFC data model. Source: buildingSMART (Liebich, 2013).

An example of the inheritance concept via the SUPERTYPE and SUBTYPE declarations is shown in line 3,9 and 12 of the EXPRESS SCHEMA code in figure 2.6(left).

3. EXPRESS can automatically define inverse relationships explicitly without remodelling any new information for example, an indirect association can be inferred between an *'air conditioning (AC)'* object and a *'room'* object by giving the *AC entity type* properties from the *room entity type*. For this case, the EXPRESS parser is able to automatically infer and generate an inverse association between the *AC object* and the *room* object via the defined shared properties (see figure 2.4).

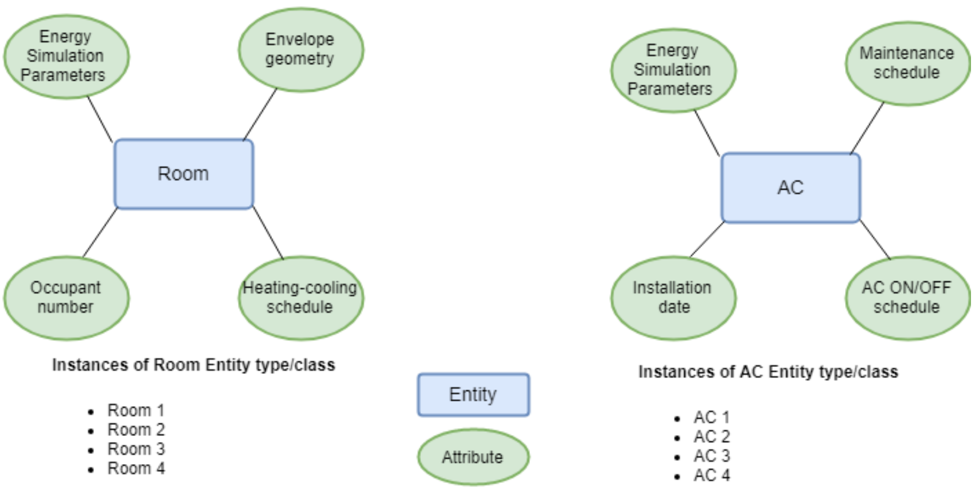

Figure 2.4: Entity relationship diagram showing both the Room (left) and AC (right) Entity types with their respective attributes and instances (below). Instances of the room entity type are indirectly associated with those of AC type via the shared property *Energy Simulation Parameters*.

```
TYPE IfcBoxAlignment = IfcLabel;
  WHERE
    WR1 : SELF IN ['top-left', 'top-middle', 'top-
      right', 'middle-left', 'center', 'middle-right
      ', 'bottom-left', 'bottom-middle', 'bottom-
      right'];
END_TYPE;

TYPE IfcLabel = STRING(255);
END_TYPE;
```

Figure 2.5: An IfcBoxAlignment data type declaration with a WHERE rule restriction,WRI. The rule specifies that instances of the data type can only be made using values in the WHERE clause. Source: Pauwels and Terkaj (2016)

```
SCHEMA Family;

ENTITY Person
   ABSTRACT SUPERTYPE OF (ONEOF (Male, Female));
     name: STRING;
     mother: OPTIONAL Female;
     father: OPTIONAL Male;
END_ENTITY;

ENTITY Female
   SUBTYPE OF (Person);
END_ENTITY;

ENTITY Male
   SUBTYPE of (Person);
END_ENTITY;

END_SCHEMA;
```

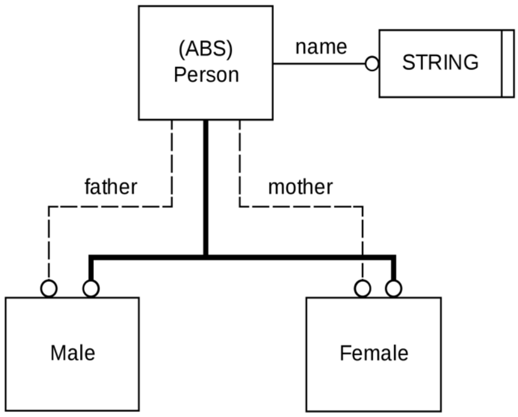

Figure 2.6: A simple example of the EXPRESS data model textually (*left*) and graphically using EXPRESS-G (*right*). Entity Person is an abstract supertype of entities male and female *shown by the thick connecting lines with a circle at the end denoting the direction of inheritance.* Every occurrence of person has a mandatory name attribute and two optional attributes father and mother *denoted by the continuous non-thick line and the dashed line respectively.* Based on principles from ISO 10303-11 (2004).

4. EXPRESS offers a variety of group data handling for example array, list, set, bag via the aggregation data type. This makes it possible to define relationships with groups of objects which is typical of building data that often exists in grouped formats, for example, a list of spatial coordinates, sensor data collected in lists, HVAC maintenance schedules, etc.

5. EXPRESS has the flexibility of describing additional algorithmic rules and restrictions on the data model using optional WHERE constructs which contain domain rules that constrain the values of attributes for every entity instance. For an instance to be deemed valid in the defined domain, it should not violate any rule defined within the WHERE construct as shown in figure 2.5.

6. Besides the text notation, EXPRESS also has the ability to model data graphically using EXPRESS-G a graphical notation language improving human readability and maintainability of the schema. EXPRESS-G, however, is not able to represent all details that can be formulated in text form. See figure 2.6(right).

2.3.1.2 Limitations and extensibility mechanisms of the IFC-EXPRESS schema

The IFC data model aims to achieve a semantic richness that supports a wide range of exchange use-cases and domain applications but only a few domain-specific concepts are explicitly modelled/covered on the schema level (Kris et al., 2016; Zhang et al., 2014). To circumvent this dilemma,

1. Firstly, IFC adopts a generic structure with only very few formalized constraints on the data model i.e. almost all attributes are OPTIONAL in the IFC specification which means that hardly any attribute requires the mandatory provision of a value to be deemed valid at any stage of the lifecycle or exchange scenario. For specialized exchange use cases, *model view definitions (MVDs)* are used to narrow down this native generic and wide scope of IFC by determining which OPTIONAL attributes need to have values asserted to satisfy the requirements of that specific exchange use-case.

2. Secondly, IFC provides attribute extension mechanisms via *'property sets'* and *'proxies'*. As already mentioned, a syntactically correct IFC instance might miss important attributes for a specific use-case, for example, the IfcDoor (an entity for modelling doors in IFC) only has two mandatory attributes: 'GlobalId' and 'OwnerHistory', IfcWindow only has 'GlobalId' as a mandatory attribute which information can only be used to identify and manage revisions of those object models. All the other information such as OverallWidth, OverallHeight, fire safety class, thermal performance, price, and material types is regarded as unnecessary for syntactic validity of the underlying data model. This is where *'property sets'* come in as an extension mechanism by dynamically creating new properties to supplement the already defined static attributes within the schema. The new individual properties are defined using *'IfcProperSingleValue'* a subproperty of *'IfcProperty'*, and thereafter grouped into an *'IfcPropertySet'* which can be assigned to the an object via *'IfcRelDefinesByProperties'*. In addition to property sets is *'IfcProxy'* a placeholder that permits dynamic definition of semantic information that is not yet defined by IFC (Borrmann et al., 2018).

A further means of extending the IFC model is provided by the externally referenced properties in libraries such as bSDD (buildingSMART Data Dictionary). Semantic web technologies (see section 2.4) and Internet of Things (IoT) suggested in Debruyne et al. (2017); Jeroen et al. (2018); Pauwels et al. (2018); Zhang et al. (2015b) are also steadily emerging as a means of providing more flexible semantic extension opportunities for the IFC schema. The above overview is by no means exhaustive but highlights the most significant underlying concepts of IFC data modelling using the EXPRESS language in an easy to understand fashion with the aim of putting the research problem in context.

2.3.2 Information Exchange requirements

The IFC schema is comprehensive and generic which makes it extremely powerful in catering for different needs of presenting building information. However, this not only makes it a complex data model but also never entirely complete i.e. the generic flexibility gives undesired freedom for domain end users and application implementers by limiting the number of problem-specific constraints at the schema level. It is therefore imperative to assign additional restrictions and constraints on the data model that determine who provides which information when and to whom with a goal of satisfying specific data exchange scenarios e.g. energy simulations, acoustic performance, structural analysis etc. Unlike building models that have structured formats and methodologies to define them thanks to IFC, additional requirements and restrictions at the schema level are naturally written in text-based documents using Information Delivery Manuals (IDM) which are then translated to machine-readable formats for processing using Model View Definitions (MVD).

2.3.2.1 Information Delivery Manuals and Model View Definitions

BuildingSMART developed standardized methodologies for capturing information exchange requirements using ***Information Delivery Manuals (IDMs)*** as specified in ISO 29481-1 (2016). The first stage of this process requires no technical knowledge of the underlying IFC schema but rather domain expertise, good knowledge and experience of best practices from past projects (Karlshøj et al., 2012; Petrova et al., 2017). The ***exchange requirements*** (ERs) are structured in a semi-formal tabular template using natural language, general-purpose diagram editors, word processing applications and spreadsheets. A ***process map*** captures these requirements holistically with their respective actors, inter-dependencies and assigned responsibilities for a specific exchange scenario as shown in figure 2.7. The means of exchange is also specified which can include but is not limited to documents and models based on agreed standards. For example for a use case of neural network energy optimization, standards such as LEAD, BREEAM and ASHRAE can be defined as mandatory compliance regulations. A process map serves as a preparatory framework for the formalization of the plain text ERs into computer processable formats known as ***Model View Definitions*** (MVDs) using mvdXML (Chipman et al., 2016; Chuck et al., 2011; Weise et al., 2016). MVDs set the threshold to be reached during the BIM certification process as briefly discussed in subsubsection 2.3.2.2.

2.3.2.2 BIM software certification using MVDs

MVDs serve as technical specifications for software vendors who wish to implement IFC exchange routines within their import-export schemas. Furthermore, they are the core of buildingSMART's quality assurance mechanism that ensures a high standard of data exchange within the ecosystem of BIM software (Borrmann et al., 2018). Zhang (2019) however, highlights that this certification procedure cannot control the quality of building model instances created by end users and it is, therefore, imperative to have a means of validating their work along the building lifecycle to ensure data reliability before exchange. It is important to note that no certification scheme can guarantee an error-free data exchange, in fact, (Borrmann et al., 2018) highlights that external independent software tests by users can identify issues not discovered by buildingSMART's certification procedure. Novel model checking technologies with reliance on modularized and extensible open-data techniques like the semantic web have been studied by (Zhang et al., 2015a; Zhang, 2019).

2.3.2.3 Domain-level requirements for information exchange

Besides the IDM-MVD routines of defining exchange requirements at the schema level, other sets of business rules known as BIM Standards often at a national level (CIC, 2015; EUBIM Task Group, 2016; Statsbygg, 2013) have been developed to check the semantic integrity and validity of models created by end users. On top of this, several construction companies define additional in-house BIM standards (MT Højgaard, 2016; Port of Portland, 2015) for their specific use-cases especially if the conventional IDM-MVD approach is not satisfactory. Such customized standards will continue to grow due to the industry's response towards the increasingly complex nature of construction projects and associated optimization problems. It is confident to say that BIM standards at the national level adhere to the already existent checking systems but the same cannot be said about the dynamic case by case in-house standards (Zhang, 2019) made by smaller individual companies mainly because;

1. The scope of domain knowledge required for a specific exchange might extend outside the IFC schema.

2. New terminologies that are outside the schema might also be introduced to define such domain knowledge.

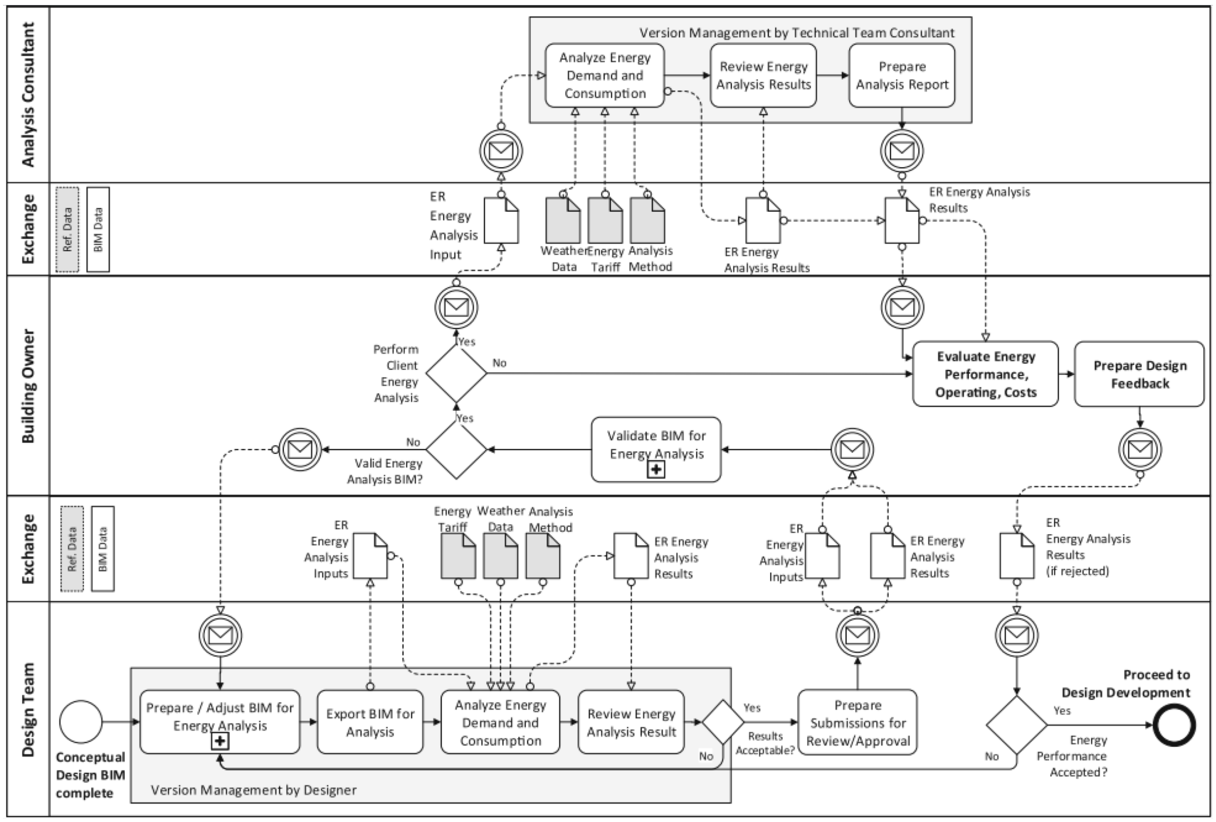

Figure 2.7: A Process Map defining the exchange requirements and actor relationships for an energy analysis exchange use case. Source: Concept Design Phase Energy Analysis IDM, developed jointly by GSA (USA), Byggforsk (Norway) and Senatii (Finland) (Borrmann et al., 2018).

Therefore more flexible and extensible ways of representing shared knowledge in the AEC/FM industry are required to cater for such dynamic exchange constraints at the domain level. In contrast to current standards, data in the IFC schema is captured statically using EXPRESS and the STEP file format. The first fundamental step in any specific exchange scenario is accessing and analysing related objects, properties and relationships from an IFC building model. The complexity of such information retrieval depends on the underlying knowledge-representation format of the data model. Several standards like EXPRESS-X (ISO 10303-14, 2005), Standard Data Access Interface (SDAI) (ISO 10303-22, 1999) and buildingSMART's IfcChecking tool have already been developed for accessing and querying specific data from an IFC model, however all these methodologies are completely reliant on STEP which is a closed and inextensible ecosystem that requires hard coding and maintainability with only very few external tools supporting it.

Semantic translations of the native EXPRESS schema into universal ontology languages (Barbau et al., 2012; Pauwels and Terkaj, 2016) will, therefore, provide more coherent models for knowledge-representation and retrieval of heterogeneous building information which can further automate requirement checking systems to cater for such dynamically defined exchange constraints along the building's lifecycle (Zhang, 2019).

Summary: In attempting to address the shortcomings of heterogeneity and fragmentation within the AEC/FM industry, Building Information Modelling emerged as a model-centric approach for propagating and handling information in a holistic fashion along the building lifecycle. Of course with the advent of BIM, a standardized way of representing and exchanging building information emerged as an open and vendor-neutral standard, Industry Foundation classes (IFC) developed by the international organization, buildingSMART. Since the first version, IFC 1.0 in 1997 to the current IFC 4, it has matured extensively into a popular data model with more than 160 software implementing it. This maturity has no wonder caught the attention of international bodies thus making it a fully operational ISO standard (ISO 16739:2016, 2016) and in fact it has become a mandatory data exchange format during construction tendering in some countries (AEC-UK, 2012). To cater for a wide range of use cases, the IFC data model is very generic with only a few internally defined constraints providing users with the flexibility of representing building information in a variety of ways depending on the use case. This, however results in a very large and complex data model for software implementers. To this effect, buildingSMART further developed Model View Definitions (MVDs) which reduce this complexity by explicitly specifying which parts of the data model need to be implemented for a specific data exchange routine. In fact, this is the basis for buildingSMART's certification process of BIM software. It is evident that the AEC/FM industry is responding to the ever-increasing complexity of the AEC industry by embracing linked and inter-operable semi-automated workflows however Pauwels and Terkaj (2016); Pauwels et al. (2017a, 2018) highlights that IFC considerably improves ***but does not*** solve information inter-operability within the AEC industry because of the lack of formal explicit and context-aware semantics in EXPRESS (Barbau et al., 2012) therefore making ontological semantic extensions for the underlying IFC data model a necessity.

With context to the main focus of this research, autonomous building energy management, the heating and cooling load of a building's indoor environment is dependent on a number of complex heterogeneous parameters ranging from interior to exterior. More importantly, most of these parameters are dependent with unknown relationships. Holistic optimization of a building's energy performance should be treated as a continuous process along the building lifecycle with iterative and heuristic (self-learning) sub-processes as the parameters in question are also very dynamic in nature. BIM's IFC data model is just the start of endless possibilities into the web of linked OPEN data technologies that utilize ontologies with flexible semantic extension capabilities (Barbau et al., 2012; Beetz et al., 2009; El-Mekawy and Östman, 2010; El-Mekawy, 2010; Gómez-Romero et al., 2015; Grimm et al., 2011; Karan and Irizarry, 2015; Kris et al., 2016; Pauwels and Terkaj, 2016; Pauwels et al., 2017a; Zhang, 2019). These provide a coherent and comprehensive knowledge base for better understanding of the existential relationships between the aforementioned dynamic parameters prior to developing an optimization scheme.

2.4 Semantic Web Technologies

The ecosystem of current BIM software is closed and optimized only for the AEC/FM industry making it difficult for other disciplines to become part of the BIM story (Jeroen et al., 2018). Considering that optimization problems within the industry are reliant on several domain experts who generate a lot of heterogeneous information, having explicit interdisciplinary collaboration is of paramount importance.

Unlike domain-specific Building Information Models (Pauwels et al., 2018), a methodology that allows various disciplines to interlink their knowledge on a data level is already existent with principles based on the classic ***World Wide Web*** (WWW) (Berners-Lee et al., 2001a,b). The common framework that allows such heterogeneous knowledge integration, sharing and re-use is called the ***Semantic Web***. Its aim is to harmonize semantic ambiguity and discrepancies in heterogeneous data schemata by adding standardized machine-readable semantics (Barbau et al., 2012) using the ***Resource Description Framework*** (RDF) data model (see subsection 2.4.1). For a building energy optimization use case, this means that non-geometrical heterogeneous data sets from other domains can be used to supplement an energy analysis building model with valuable attributes. Building sensor data, geographical and weather data, occupant behaviour information, space usage and equipment on-off schedules are examples of such heterogeneous supplementary information. Homogeneity of this nature cannot be achieved using the current BIM(IFC-EXPRESS) schema therefore necessitating schema translations into open and extensible data structures using Semantic Web technologies (Pan and Ren, 2004; Pauwels et al., 2010; Yang and

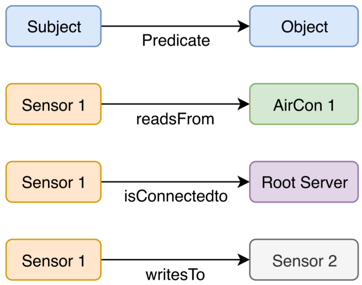

Figure 2.8: RDF triples in the form *subject-predicate-object*. The arrows implies directionality of the relationship.

Zhang, 2006). It is not possible nor desirable to give a full overview on the 'Semantic Web' as the concepts will go too far and quickly become irrelevant to the main research questions therefore only a brief but comprehensive enough introduction to the underlying structure of this open and extensible knowledge-representation structure is provided.

2.4.1 Resource Description Framework

The RDF data model (Manola et al., 2014) is in parallel with object-oriented modelling approaches in IFC where notions of *entities/classes* related by *associations* are respectively represented in RDF using ***concepts*** related with ***properties*** (Pauwels and Terkaj, 2016). Anything described in the semantic web context is called a ***resource*** meaning that concepts and properties are all defined as resources (Studer et al., 2007). RDF provides a way of semantically describing these resources by making simple statements about them. These statements are called ***triples*** and syntactically take the ***'subject-predicate-object'*** format (Manola et al., 2014) as shown in figure 2.8. It is also obvious that multiple statements about the same resource increase its semantic meaning and richness as shown in figure 2.8 and figure 2.9.

2.4.2 Uniform Resource Identifiers, literals and QNames

Another characteristic of the semantic web is the ability to uniquely identify each resource in an RDF graph using a ***Uniform Resource Identifier*** (URI). This makes the graphs explicitly labelled and allows publishing of resources anywhere on the web without any ambiguity (Berners-Lee et al., 2001b,a). Apart from URIs, exists ***Literals*** with values of a certain data type e.g. strings, integers, boolean, etc. The subject is always identified by a URI while the object might be identified by a URI or Literal.

Using figure 2.9 as an illustrative example; the nodes and edges have only been labelled with simple names such as 'Sensor1' and 'writesTo' which are not explicit enough for use on the world wide web of linked data i.e. there could be another 'Sensor 1' that writes to 'RootServer' meaning that it is necessary to explicitly identify which 'Sensor 1' and which 'RootServer' is in question. A better representation for the subject 'Sensor 1', predicate 'writesTo' and the object 'RootServer' would therefore be;

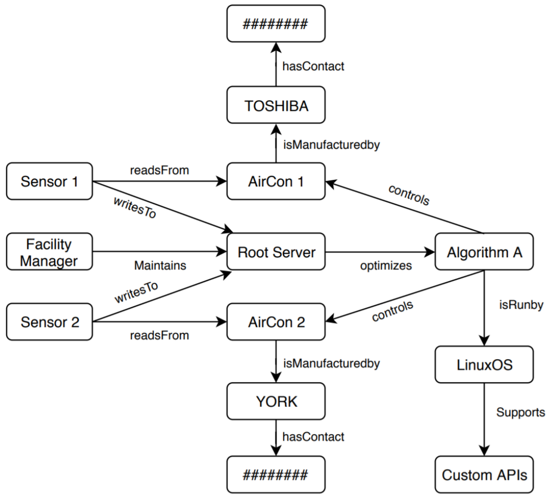

Figure 2.9: An example of an RDF graph (combination of triples) describing some information about sensors in a building connected to different air conditioning units and managed by a root server.

'http://unmcsharepoint/KevinLM/energyRLO/erlo.ttl*/Sensor1'*,
'http://unmcsharepoint/KevinLM/energyRLO/erlo.ttl*/writesTo'* and
'http://unmcsharepoint/KevinLM/energyRLO/erlo.ttl*/RootServer'* respectively using URIs.
URIs are however very long making triples less human-readable and may contain prohibited characters for resource labeling. Therefore, ***QNames*** (Qualified Names) (Bray et al., 2009) are often adopted as abbreviations to URIs. A QName has two parts; a ***namespace*** and an ***identifier*** in the form *'namespace:identifier'*. The namespace is just the URI reference to someplace hosting the definitions used in a specific RDF model and can be further abbreviated using an arbitrary namespace ***prefix*** (W3C, 2013). The identifier on the other hand simply pinpoints the exact location of a resource in that namespace. With reference to the above definition,
the URI ***'http://unmcsharepoint/KevinLM/energyRLO/erlo.ttl'***, is evidently the namespace/ reference to the repository holding the vocabulary and resources that will be constructed in this research. This URI namespace is obviously too long and can be abbreviated using a random prefix 'erlo'(energy reinforcement learning ontology). Therefore using QNames, one can explicitly refer to the predicate *'http://unmcsharepoint/KevinLM/energyRLO/erlo.ttl/writesTo'* by just declaring 'erlo:writesTo' where 'erto' is the namespace prefix and 'writesTo' is the explicit resource identifier. The nomenclature of graphs in figure 2.9 can therefore be transformed accordingly as shown in figure 2.10.

2.4.3 Turtle

Turtle is a textual serialization[1] format (Beckett and Berners-Lee, 2011) used to store sets of triples from an RDF graph in a compact form that can be published on the web as Linked Data documents while adopting QName abbreviation methods (Bray et al., 2009). Systems that use such documents require parsers[2] (W3C, 2006) that can read the used serializations and convert them back to RDF triples. Parsing

[1]Process of translating the RDF ontology graph structures into a simple format that can be stored.
[2]Parsing is the opposite of serialization i.e. The process of reading a stored turtle file and writing /converting it back to graph format

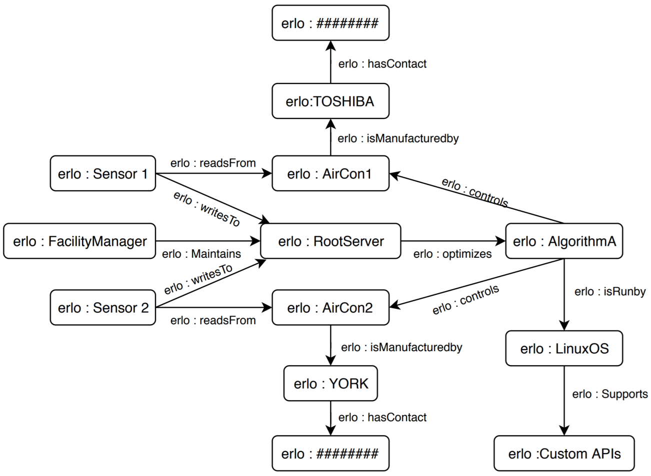

Figure 2.10: The RDF graph with resources labelled using QNames.

speed depends on the size of the documents to be read and complexity of the underlying serialization format Studer et al. (2007). Parsing techniques such as 'streaming' make it possible for RDF triples to be processed as soon as they are read (Apache Jena, 2009). This allows very large documents to be parsed even when they don't fit in the available memory. N3 JavaScript (Berners-Lee and Connolly, 2011) supports parsing via streaming which makes Turtle a beneficiary as it is part of the N3-like serializations. This research is therefore adopting the turtle serialization to store RDF triples because of its good human readability and parsing speed compared to other formats like RDF/XML (Beckett, 2014) and JSON-LD (Kellogg and Champin, 2019).

2.4.4 RDF Schema (RDFS), Ontologies and the Ontology Web Language (OWL)

RDF[3] is just a conceptual data model that can only make statements about resources in the form of triples (see subsection 2.4.1) but lacks the semantics to support data validations and advanced machine reasoning. The RDF Schema (RDFS)[4] (Brickley and Guha, 2014) therefore extends the semantics of an RDF model by providing additional vocabulary to;

1. structure related resources under classes which can be instantiated (see subsubsection 2.4.4.1).

2. assert domain and range constraints on the use of properties (see subsubsection 2.4.4.3).

3. introduce class and property hierarchies using the subClassOf and subPropertyOf constructs (subsubsection 2.4.4.2)

[3]RDF is also defined as a standard vocabulary with a set of definitions reused for basic data descriptions. An example of such definitions is **rdf:type** described in footnote 3.

[4]RDFS is also standard vocabulary just like RDF and helps to define or describe classes using definitions like **rdfs:Class**, **rdfs:subClassOf**, **rdfs:domain** and **rdfs:range**.

Subject	Predicate	Object
erlo:buildingEquipment	rdf:type	rdfs:Class
erlo:energyConsumingEquip	rdf:type	rdfs:Class
erlo:airCon	rdf:type	erlo:buildingEquipment
erlo:airCon	rdf:type	erlo:energyConsumingEquip

Table 2.1: Triples showing the Individual-Class relationship via rdf:type property

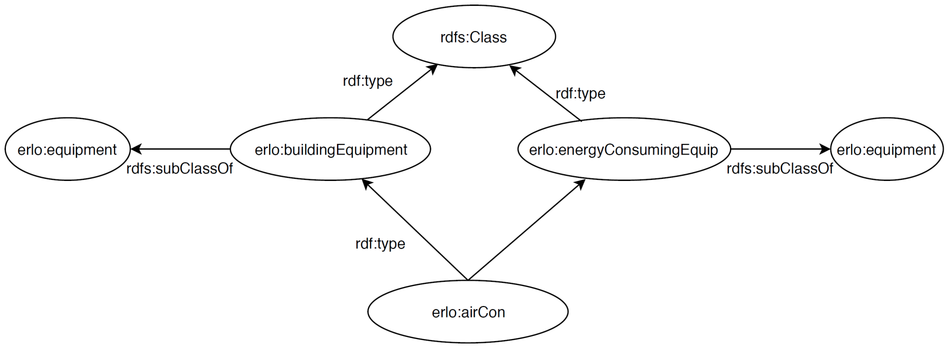

Figure 2.11: Graphical Representation of table 2.1

2.4.4.1 Classes and individuals

Classes provide the mechanism for grouping related resources. The grouped resources therefore become instances/individuals of the grouping class. A resource can be an instance of multiple classes for example, the triples in table 2.1 show that via the **rdf:type**[5] property/predicate, both *'erlo:buildingEquipment'* and *'erlo:energyConsumingEquip'* are classes and the individual *'erlo:airCon'* is a member of both classes. All classes belong to a meta-class **rdfs:Class** and all resources are instances of a meta-class **rdfs:Resource**.

2.4.4.2 Class and Property hierarchies

Classes can be organized in hierarchies by using the **rdfs:subClassOf** construct. Class hierarchies allow definition of classes from a very generic level to a more specific level. This also applies to properties using the **rdfs:subPropertyOf** construct (Brickley and Guha, 2014). For example if the specific class *'erlo:buildingElement'* is a subclass of a more generic class *'erlo:equipment'*, then all instances of *'erlo:buildingElement'* are also instances of *'erlo:equipment'* (see figure 2.11). Similarly, if a property *'erlo:readsFromMachine'* is a subproperty of *'erlo:readsFrom'*, anything that relates to something else via *'erlo:readsFromMachine'* would also relate to it via *'erlo:readsFrom'*.

2.4.4.3 Properties and constraints

All property types are individuals of the core class **rdf:Property** and can be constrained via the **rdfs:domain** and **rdfs:range** constructs (Brickley and Guha, 2014). For clarity, this is shown graphically

[5]Individual-Class relationship is expressed by the predicate/ property **rdf:type** usually abbreviated by 'a' for better human readability

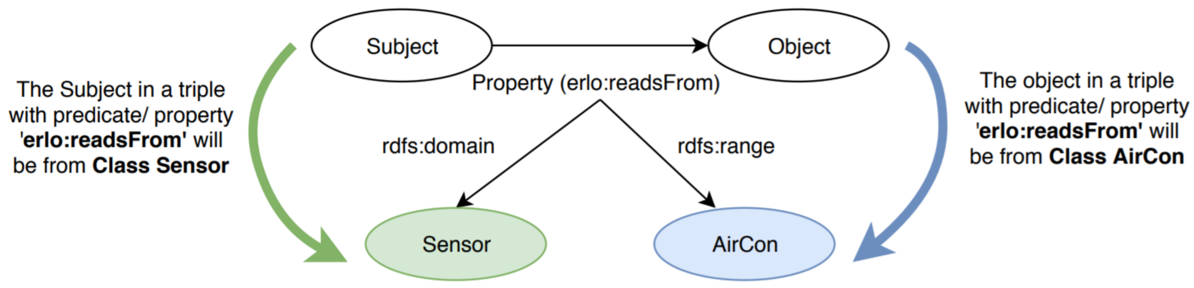

Figure 2.12: Property constraints via rdfs:domain and rdfs:range.

in figure 2.12. Logically, if a certain subject, 'equipment' is connected to a certain object, 'unit' via the property 'erlo:readsFrom', one could infer that the subject, 'equipment' must be from the class, 'sensor' which is the property domain and the object, 'unit' must be from the class, 'AirCon' which is the respective property range. For clarity, this principle is explained in greater detail by Brickley and Guha (2014).

2.4.4.4 Web Ontology Language (OWL) and Ontologies

RDFS is built on top of RDF and OWL is built on top of RDFS. This hierarchical buildup is a response to the semantic and expressivity[6] demands of the required knowledge base model (Pauwels et al., 2018). It is therefore evident that OWL (L. McGuiness and van Harmelen, 2004) extends the expressive power of RDFS for describing RDF data by availing description logic (DL)[7] (Baader et al., 2003) reasoning that can be exploited by computer programs. OWL contains vocabularies that allow more complex RDF statements to be made via cardinality restrictions, class disjointness and complex class expressions which cannot be provided by RDF or RDFS. It is important to note that both RDFS and OWL are ontology languages that provide vocabularies[8] for the description of ontologies[9] depending on the expressive power of the knowledge base required. Several OWL profiles (Motik et al., 2012) exist with different expressive power however not all of them are currently supported computationally. To this effect, this research will remain within the OWL 2 DL profile (Hitzler et al., 2012; W3C OWL Working Group, 2012) which is the most expressive while retaining computational decidability using tools provided by the semantic web community (Pauwels and Terkaj, 2016).

2.4.4.5 Terminology Box (TBox) and Assertion Box (ABox)

With ontologies, it is important to delineate the boundary between resources that are instances and those that are concepts (classes or properties) using the **Terminology Box (TBox)** and **Assertion Box (ABox)** respectively Giacomo and Lenzerini (1996). The logic behind is that assertions in ABox are

[6]The expressive power of a language is the breadth of ideas that can be represented and communicated in that language. The more expressive a language is, the greater the variety and quantity of ideas it can be used to represent

[7]DLs are used in artificial intelligence to describe and reason about the relevant concepts of an application domain. It is of particular importance in providing a logical formalism for ontologies and the Semantic Web (Borrmann et al., 2018)

[8]There is no clear distinction between ontologies and vocabularies within the semantic web context, they are used interchangeably quite often. In fact, a common ontology defines the vocabulary with which queries and assertions are exchanged/re-used among users

[9]An ontology is an explicit and formal specification of a conceptualization. A conceptualization is a simplified view of the world that we wish to represent for some purpose.

made against a well-crafted set of terminologies in a TBox. TBox statements remain static over time as these represent the underlying schema or taxonomy of the domain at hand whereas ABox assertions keep changing depending on validity after inference[10] in the TBox. For example 'every sensor is a building equipment' is a TBox statement while 'temperature monitor is a sensor' is a typical ABox statement. Therefore, one can infer that temperature monitor is a 'building equipment'. Principles of ABox and TBox are discussed in greater detail by Giacomo and Lenzerini (1996).

2.4.5 Querying Mechanisms for Semantic Web Data

The complexity and vastness of semantic web models necessitate a methodology for searching, filtering out and validating the information from them. A number of languages exist currently but only SPARQL (SPARQL Protocol and RDF Query Language) is going to be reviewed because it the only standardized and widely used query language for retrieving and manipulating data stored in RDF formats using triple formats (Harris and Seaborne, 2013; W3C SPARQL Working Group, 2013). A basic SPARQL query is shown in listing 2.1 consisting of;

1. URI prefix declarations that indicate the vocabularies used during the query.
2. Dataset definition stating which RDF graph(s) will be queried using 'FROM' and 'FROM NAMED' clauses[11].
3. A result clause (SELECT, INSERT, CONSTRUCT, ASK etc.) identifying what information/ variables (preceded by a '?') to return from the query.
4. The triple patterns to which the variables have to comply using the 'WHERE' construct.
5. Query modifiers that provide means of slicing, ordering, and rearranging query results e.g. ORDER, PROJECTION, DISTINCT, REDUCED, OFFSET and LIMIT.

```
PREFIX erlo: <http://unmcsharepoint/KevinLM/energyRLO/erlo.ttl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
FROM <http://unmcsharepoint/KevinLM/energyRLO/erlo.ttl>
SELECT ?ACunits ?sensor ?value
WHERE {
?erlo:ACunits rdf:type ?erlo:BuildingEquipment .
?erlo:erlo:Sensor erlo:readsFrom ?erlo:ACunits .
?erlo:Sensor erlo:hasValue ?erlo:AboveThresholdSensorValue .
}
```

[10]Inference is a reasoning process in semantic web applications based on a chain analysis of triples to discover new relationships between resources based on implicit information described within the TBox. This is the basis for consistency, rule checking and querying using languages like SPARQL

[11]A SPARQL query may specify the dataset to be used for matching by using the 'FROM' clause and the 'FROM NAMED' clause to describe the RDF dataset. If these clauses are omitted, the query would therefore use the default dataset

listing 2.1: A basic SPARQL query retrieving AC units whose sensor value is above a certain set threshold. No specific query modifiers have been defined

SPARQL provides a powerful mechanism for visualizing specific parts of a rather large and complex data model for example, a facility manager could want to identify all air conditioning units in a building whose sensor values gave a specific value (see listing 2.1). That information could thereafter be used for other decision-making purposes like identifying why certain rooms are cooler or warmer, or if certain warm rooms have more windows facing the sun etc. Such a holistic picture of queryable interconnected building data provides a rich knowledge base for optimization of a building's energy performance which is influenced by a number of factors which have indirect inter-dependencies.

2.5 Applying Linked Data technologies to Building Information Models

Linked Data principles hae been introduced in the preceding sections and it is evident that when applied to such a fragmented and multi-disciplinary AEC industry, significant progress will be realized towards the development of well-organized, more open and extensible semantic structures of representing, sharing and re-using building information (Elghamrawy and Boukamp, 2008, 2010; Pan and Ren, 2004; Pauwels et al., 2017b).

2.5.1 Industry track towards semantic inter-operability of BIMs

Currently, the industry's way of achieving inter-operability via Building Information Modelling (BIM) (Chuck et al., 2011) is insufficient because of IFC's (see subsection 2.3.1) complex and inextensible structure which is optimized only for the AEC industry. This makes it hard for other disciplines like GIS, FM, heritage and energy domain to become part of this closed BIM story. Semantic inter-operability requires the adoption of formal, explicit and context-aware semantic data definitions that can be understood across various disciplines (Yang and Zhang, 2006) unlike domain-specific BIMs relying solely on IFC.

Several early efforts to embrace open knowledge-representation schemes within the industry emerged with reliance on project-specific ontologies that were hard to re-use or extend formally to other domains because of the different vocabularies and taxonomies employed. Some of these works include the e-COGNOS project from which the e-COGNOS ontology emerged (Wetherill et al., 2002), the inteliGrid project ontology for sharing semantics between applications (Dolenc et al., 2007), Yang and Zhang (2006)'s proposal of an early prototype to support inter-operability of BIMs and project data, Elghamrawy and Boukamp (2008, 2010)'s ontologically driven model that supports management of and learning from construction problems by holistically integrating project data. Other notable research in this area can be found in Abdul-Ghafour et al. (2007); Le and David Jeong (2016); Pauwels et al. (2010); Scherer et al. (2012); Shah et al. (2011) and Venugopal et al. (2015).

2.5.2 Standardization efforts towards reusable ontologies

Specific to IFC, a recommendable and reusable OWL translation of IFC was proposed by Pauwels and Terkaj (2016) which was later agreed upon by the Linked Data Working Group (LDWG) (W3C, 2014). Prior to this however, several efforts to convert IFC-STEP to RDF were made by Agostinho et al. (2007); Beetz et al. (2005); Krima et al. (2009); Pauwels et al. (2015); Schevers and Drogemuller (2005) and Zhao and Liu (2008) which proposals, in fact, formed the basis for Pauwels and Terkaj (2016)'s work.

The ifcOWL ontology has further been modified by Pauwels et al. (2017a) for better representation of geometric data. Terkaj and Šojić (2015) proposed an extension to ifcOWL in which EXPRESS WHERE rules were translated to OWL and included in the ifcOWL ontology. In addition, Gómez-Romero et al. (2015) proposed a fuzzy logic-based extension to the ifcOWL ontology that provides support for imprecise knowledge representation and retrieval which is characteristic of ontologies. Since building data is now available in a simple formalized semantic graph rather a complicated IFC schema, it can be restructured and simplified to better match requirements of practical use cases and several approaches of doing this have been proposed by Pauwels and Roxin (2016) using ifcOWL.

2.5.3 Towards simplicity, modularity and extensibility of ontologies.

The ifcOWL ontology is very large as it encapsulates the entire IFC schema and without doubt, can often prove to be redundant in several use cases or even hard to query. To this effect, W3C's Linked Building Data Community Group (W3C, 2014) has developed simpler, modular and extensible ontologies with intent to cover the IFC schema in smaller and more manageable modules namely;

1. Building Topotology Ontology (BOT)
2. Product Ontology (PRODUCT)
3. Properties Ontology (PROPS)
4. Geometry Ontology (GEOM)

The BOT ontology (Rasmussen et al., 2017a,b, 2019) serves as the key ontology for capturing the building topology (core of IFC) which is extensible to other domain ontologies like the building device automation domain (Bonino and De Russis, 2018; Schneider, 2017; Villalón and Castro, 2017), sensor domain (Haller et al., 2017), geo-spatial domain (McGlinn et al., 2017), energy performance, indoor climate, HVAC Installations and Facility Management domains. These extensions are aided by both

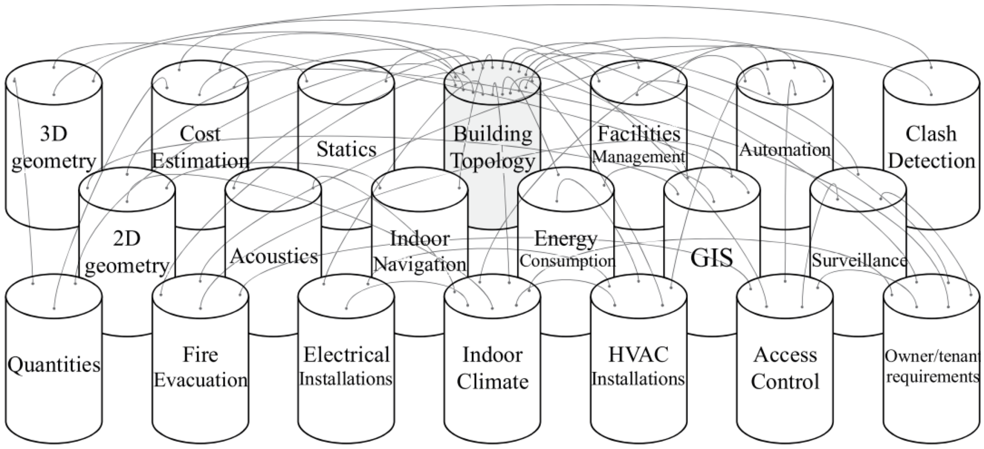

Figure 2.13: The BOT ontology extending to other domain data sets with inter-dependencies. (Extracted from Rasmussen et al. (2017b).

the Product ontology (Hepp, 2008; W3C-Linked Data Community Group, 2018b) and the Properties ontology (Rasmussen et al., 2018) which capture the 'semantics' and 'properties' of any tangible object in a building respectively. In addition, the GEOM ontology, which is still in development by the W3C-Linked Data Community Group (2018a), captures the 3D building data. While adopting this single data modelling (RDF) scheme of representing building data via ifcOWL and other modular ontologies cannot address bad implementation and usage practices, it might be the ideal technical means towards semantic domain inter-operability while allowing extensibility and adaptability of the continuously evolving semantic structures of the AEC industry.

2.5.4 Applying Linked Data technologies to optimizing the energy performance of buildings

Buildings are becoming increasingly complex and so is their management and operation (Curry et al., 2012). With context to this research, energy optimization of a building is a multi-domain problem encapsulating several trade-off issues when trying to balance thermal comfort, indoor air quality and optimal energy use. Of course, it is evident that such a problem scope requires an extensive knowledge base inspired by linked open data-driven schemes that provide designers and facility managers with a holistic picture of the various heterogeneous parameters that affect a building's energy performance.

Several research efforts have emerged to embrace semantic web approaches in solving such dynamic problems. For instance, Curry et al. (2012) combined Linked Data with scenario modelling to support inter-operability during optimization of building performance. Kris et al. (2016) analysed 33 EU projects that utilized BIM-based energy management plus their data requirements in order to identify those that can benefit from open linked data structures. Anzaldi et al. (2018) proposed a holistic knowledge-based approach for intelligent building energy management using a combination of ontologies, algorithms and simulations. Radulovic et al. (2015) even went ahead to present a set of best practices and guidelines for generating and publishing Linked Data with BIMs in the context of energy consumption in buildings. Corry et al. (2015) and Scherer et al. (2012) developed a performance assessment ontology that structures heterogeneous building data into semantically enriched information which can support energy management of buildings. A unified energy representation for smart cities via the DogOnt was proposed by Bonino and De Russis (2018) by integrating several sub-domains of energy representation namely; electrical, thermal and city level energy profiles. Dibley et al. (2011) and Dibley et al. (2012) coupled a multi-agent system with an ontology, 'OntoFM' to support real-time monitoring of building sensors in an automated and holistic way. Their work inherited principles from a building ontology based on IFC, a sensors ontology (OntoSensor) (Russomanno et al., 2005) and a general purpose ontology SUMO (Suggested Upper Merged Ontology) (Niles and Pease, 2001) which captures domain-independent concepts.

To support inter-operability and exchange of data between building energy simulation tools, 'SimModel', an XML based data model, was proposed by Donnell et al. (2011). Pauwels et al. (2014a,b) then went ahead to avail this model as RDF graphs which can be combined with other RDF data. Tah and Abanda (2011) developed an ontology to represent information about photo-voltaic systems which are a renewable energy technology that transforms energy from the sun into electricity using photovoltaics (also known as solar panels). Reinisch et al. (2011) and Kofler et al. (2012) proposed a comprehensive 'ThinkHome system' that relies on an extensive ontological knowledge base to store all information needed to fulfil goals of energy efficiency and user comfort in future smart homes. This multi-agent system interacts with the knowledge base via SPARQL queries and DL inference to autonomously control a smart home. Much of the ThinkHome Ontology is inspired by DomoML-env (Sommaruga et al., 2005), an ontology for human-home interaction aiming to connect household appliances to each other and share information about their usage. The aforementioned ontologies can also be combined with a set of SWRL (Semantic Web Rule Language) rules that automatically apply energy management strategies through inference with the knowledge base Rossello-Busquet et al. (2011). Specifically, these rules enable the inference engine to infer if there are any anomalous activities occurring (e.g. 'air conditioners' that are 'working' AND 'windows' that are 'open'). A SPARQL endpoint can even be put on top of this rule engine so that the user only has to query for the results of the rules. Other systems utilizing the same SWRL approach to manage smart home appliances have been proposed by Ricquebourg et al. (2007) and Tomic et al. (2010).

2.6 Deep Reinforcement Learning for Autonomous Building Energy Management

2.6.1 Introduction

Another growing trend in the building energy domain is the use of machine learning specifically a combination of ***deep learning*** and ***reinforcement learning*** (RL) as a means to automate energy optimization processes through ***sequential decision making*** (LeCun et al., 2015; Sutton, 1988). RL is a mathematical framework for autonomous experience-driven learning and although it has had numerous successful attempts in crafting responsive and context-aware AI systems (Han et al., 2018; Kohl and Stone, 2004; Mason and Grijalva, 2019; Yang et al., 2015; Yu and Dexter, 2010), previous approaches lacked scalability to high dimensional problems due to memory and computational complexity (Strehl et al., 2006). The advent of deep learning has provided tools that overcome those challenges through function approximation using deep neural networks modelled loosely after the human brain. Simply put, these are sets of algorithms designed to recognize patterns and perform specific tasks by learning from example datasets autonomously without being programmed with any task-specific rules (LeCun et al., 2015). The use of such deep learning algorithms within RL defines the field of ***Deep Reinforcement Learning*** (DRL). This has become a promising approach towards solving real-world optimization control tasks within high dimensional state-action spaces like buildings where efficient rationalization of heating and cooling demand can quickly become complex due to dependence on stochastic parameters like changing weather seasons, variable occupant behaviour and number, different airflow patterns and building material properties (Mason and Grijalva, 2019).

The next section aims to provide an overview on the concepts behind reinforcement learning and the general modelling process of RL problems which will be followed by an examination of the contributions of deep neural networks to RL.

2.6.2 Reinforcement Learning

Reinforcement learning in general consists of an ***agent*** interacting in an ***environment***, learning what ***actions*** to take depending on the ***state*** of the environment (see figure 2.14 left). The learning process is through trial and error with a reward for taking desirable actions. The obvious goal is to maximize ***long term reward*** through ***exploration and exploitation*** (Sutton and Barto, 2018). Exploitation is when the agent takes the best known action most of the time (maximizing immediate reward) but occasionally with a ***probability (ε)***, the agent explores randomly through unknown actions so as to discover new rewards even if it means sacrificing an already known immediate reward as illustrated in figure 2.14 (right).

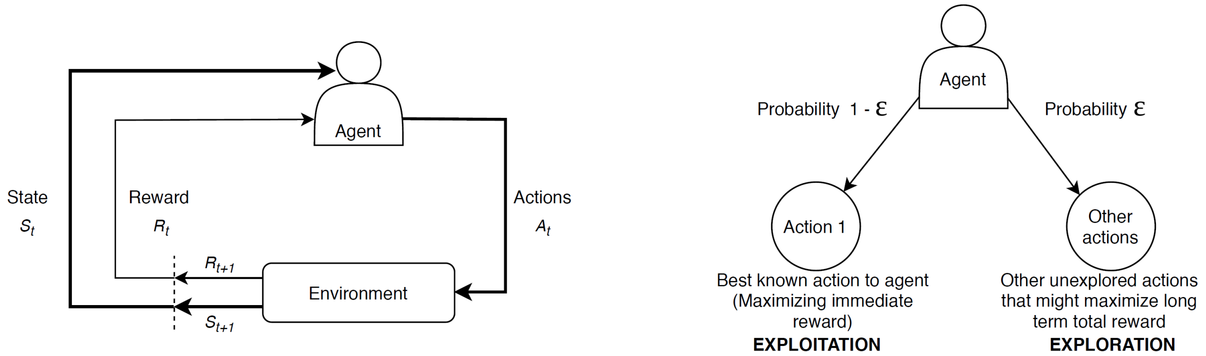

Figure 2.14: *(Left)* Agent Environment Interaction. *(Right)* If $\varepsilon = 0.2$, this means that for 80% of the time the agent chooses to take the best known action (1) while trying to maximize immediate reward, and 20% of the time trying to explore other actions (with equal chance) with the hope of maximizing long term total reward. Based on principles from François-lavet et al. (2018).

2.6.2.1 Reward-driven behaviour in the Markov Decision Process

Generally, a reinforcement learning problem is modelled as a ***Markov Decision Process*** (Bellman, 1957) which is a 5-tuple *($\mathcal{S}$,$\mathcal{A}$,$\mathcal{P}$,$\mathcal{R}$,γ)* with $\mathcal{S}$ being the set of states the environment can be, $\mathcal{A}$ being a set of possible actions, $\mathcal{P}$ being the state transition probability function (2.1) specifying the probability that taking action ***a*** in state ***s*** at time step ***t*** will lead to state ***s'*** at time step ***t+1***,

$$\mathcal{P}^a_{ss'} = P[S_{t+1} = s' | S_t = s, A_t = a] \tag{2.1}$$

$\mathcal{R}$ is the reward function (2.2) that returns the reward received when the agent transitions to ***s'*** after taking action ***a*** in state ***s***,

$$\mathcal{R}^a_s = E[R_{t+1} | S_t = s, A_t = a] \tag{2.2}$$

and $\gamma \in$ [0,1) being the discount rate which decides the future value of rewards as discussed later.

Any RL problem modelled as an MDP follows the ***Markov Property*** which states that the future is independent of the past given the present i.e $P[S_{t+1} \mid S_t] = P[S_{t+1} \mid S_1, S_2, S_3, ..., S_t]$ meaning that the current state carries sufficient information about the environment from the past necessary for the agent to take the best action. This property however, only holds if the states are fully observable which is the case for most RL problems. In a Partially Observable Markov Decision Process (POMDP), there is no certainty about the current state which makes selecting actions based on that state invalid. The dynamics of a POMDP are the same as that of an MDP however, instead of directly observing the current state, the state gives the agent an observation which provides a hint about what state it is in.

2.6.2.2 Policy, value functions and optimality

The agent always interacts with the environment in discrete time steps using a ***policy** (π)*[12] which determines which actions the agent will take. The goal is to learn a policy that maximizes reward over the long run, the ***optimum policy*** $\pi^* \in \Pi$, where Π is the set of all possible policies. The ***state-value function*** given in (2.3) calculates the value of a policy π in a given state $\boldsymbol{s}$. This value is used by the agent to assess how good it is to be in a given state. Similarly the ***action value-function*** in (2.4) defines how good it is to perform a certain action $\boldsymbol{a}$ in a given state $\boldsymbol{s}$ under a policy π. Conventionally, this action-value function $Q^{\pi}(s,a)$ is referred to as the ***Q-function*** which outputs a ***Q-value*** for any given state-action pair. The notion of 'how good' is defined in terms of the expected future rewards the agent will obtain starting from state $\boldsymbol{s}$, taking action $\boldsymbol{a}$ and thereafter following policy π.

$$V^{\pi}(s) = \mathbb{E}[\textstyle\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s, \pi] \tag{2.3}$$

$$Q^{\pi}(s,a) = \mathbb{E}[\textstyle\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s, a, \pi] \tag{2.4}$$

$\mathbb{E}$ = expected future return, $\boldsymbol{t}$ = is any time step, $\boldsymbol{k}$ = number of time steps in the future, γ = ***discount rate***, (the amount of weighting the agent gives to future rewards between 0 and 1 i.e it determines the present value of future rewards). A discount factor close to 0 means the agent is short-sighted by only considering current/ immediate rewards, while a factor approaching 1 will make it strive for long term rewards. In general, short-sighted agents have reduced access to future rewards which in turn reduces the expected return. The value function for an optimal policy π^* in state s is given by (2.5) calculated using the Bellman equation (Sutton and Barto, 2018). Similarly, the Bellman optimal action-value function is given in (2.6).

$$V^{\pi^*}(s) = max\mathbb{E}[r_{t+1} + \gamma V^{\pi^*}(s_{t+1}) | s, \pi^*] \tag{2.5}$$

$$Q^{\pi^*}(s,a) = \mathbb{E}[r_{t+1} + \gamma \max_{a'} Q^{\pi^*}(s', a') | s, a, \pi^*] \tag{2.6}$$

2.6.2.3 Model-Free and Model-Based RL algorithms

If all the elements in an MDP tuple *($\mathcal{S}$,$\mathcal{A}$,$\mathcal{P}$,$\mathcal{R}$,γ)* are known, the solution to the MDP can just be computed easily without taking any actions in the environment. However, this is never the case as the agent does not know all the elements of the MDP. Specifically, the agent does not know how the environment will change in response to its actions (transition dynamics function $\mathcal{P}$), nor the immediate reward it will receive for doing so (reward function $\mathcal{R}$). The agent will just have to

[12]A policy is a stochastic rule by which the agent selects actions as a function of states i.e a = π(s) where a is an action and s is a state.

randomly try different actions while observing what happens in the environment and somehow find a good policy from doing so. Without $\mathcal{P}$ and $\mathcal{R}$, the agent cannot ***plan***[13] a solution that finds a good policy.

Model-based RL agents have to first learn a model of how the environment works ($\mathcal{P}$ and $\mathcal{R}$) and thereafter, together with a planning algorithm, decide what actions to take (François-lavet et al., 2018). The way the agent learns this model is by initially taking an action $\boldsymbol{a_t}$ in state $\boldsymbol{s_t}$ and observing the transition of the environment to state $\boldsymbol{s_{t+1}}$ with a reward $\boldsymbol{r_{t+1}}$. Iteratively, that information improves the estimate of (2.1) and (2.2) via supervised learning until the agent has adequately learned the environment dynamics. The agent will always use this learnt model to make predictions about the next state and reward before taking an action (Sutton and Barto, 2018). This model-based approach is rarely used as it is difficult to have an accurate model of the environment dynamics.

Model-free approaches on the other hand do not need to develop a model of the environment but rather learn a policy via trial and error using value functions until an optimum policy is approximated (Arulkumaran et al., 2017). These are the most commonly used RL algorithms and are the only ones reviewed in this write-up forexample Q-Learning (Rummery and Niranjan, 1994) and SARSA (State Action Reward State Action) (Kröse, 1995).

2.6.2.4 Q-Learning

Q-Learning is a model-free RL algorithm that aims to learn an optimal Q-function which directly estimates the optimal Q-values associated with each state-action pair (Rummery and Niranjan, 1994). From this, an optimal policy may be derived by choosing an action with the highest Q-value (reward) in the current state. For discrete search spaces, these Q-values are stored in a look up table which is a matrix of dimensions $\mathcal{S}$ x $\mathcal{A}$. The Q-learning algorithm iteratively updates the stored Q-values using the Bellman equation until the Q-function converges to the optimal Q-function. This update process is done using ***value iteration*** while adopting the ***epsilon greedy strategy*** to balance exploitation with exploration as discussed in subsection 2.6.2 earlier. Initially, the exploration rate is set at $\varepsilon = 1$ so that the agent can explore with 100% certainty since it starts without knowing anything about the environment to exploit. ε gradually decays by some rate to favour exploitation (greedy behaviour) as the agent learns more about the environment (Sutton and Barto, 2018). The agent updates its Q-table according to Equation (2.7), where α is the ***learning rate*** $\alpha \in$(0,1) which determines the extent to which newly learned Q-values override old ones.

$$Q^{new}\left(s_t, a_t\right) = (1-\alpha)\underbrace{Q\left(s_t, a_t\right)}_{\text{old value}} + \alpha\overbrace{\left(r_{t+1} + \gamma \max_{a'} Q\left(s', a'\right)\right)}^{\text{learned value}} \tag{2.7}$$

According to Sutton and Barto (2018), a Q-Learning algorithm exhibits ***off-policy*** learning meaning

[13] Planning is any method that utilizes a model to produce or improve a policy (Sutton and Barto, 2018).

that the agent learns from actions that are outside the current policy through random choice of actions, rendering a policy useless. On the other hand, SARSA adopts an on-policy learning approach in which the agent estimates its policy value at any time while using that particular policy.

2.6.3 Combining neural networks with reinforcement learning

Storing Q-values in a look up table is convenient enough for small state spaces however, for more complex and continuous search spaces, the table quickly becomes large and computationally inefficient. One commonly adopted solution to this problem is replacing the Q-table with a function approximator for estimating the optimal Q-function. One such approximator is a ***neural network*** (see figure 2.18) which can deal with exceedingly large state spaces. This is the basis for most of the work in ***deep reinforcement learning (DRL)*** and Atari games (Mnih et al., 2015). Before introducing DRL, a brief overview on neural networks and their learning architecture is provided below.

2.6.3.1 Artificial Neural Network (ANN)

An ANN is simply a set of algorithms designed to recognize patterns and perform specific tasks by learning from example datasets autonomously without being programmed with any task-specific rules (LeCun et al., 2015). In more detail, it is a computing system comprised of a collection of connected calculation units (nodes) called ***artificial neurons*** which are organized in layers. Generally, an ANN has an input layer which accepts input data, a hidden layer(s) which applies different transformations to the input data and an output layer that delivers predictions made by the network. The notion of *'artificial'* in ANN comes from loosely replicating the behaviour of neurons in the biological brain as shown in figure 2.15. Just like a biological ***synapse*** in the brain allows a neuron to pass a chemical signal to another neuron, ANNs adopt ***weighted values*** for the same task while using real numbers as signals.

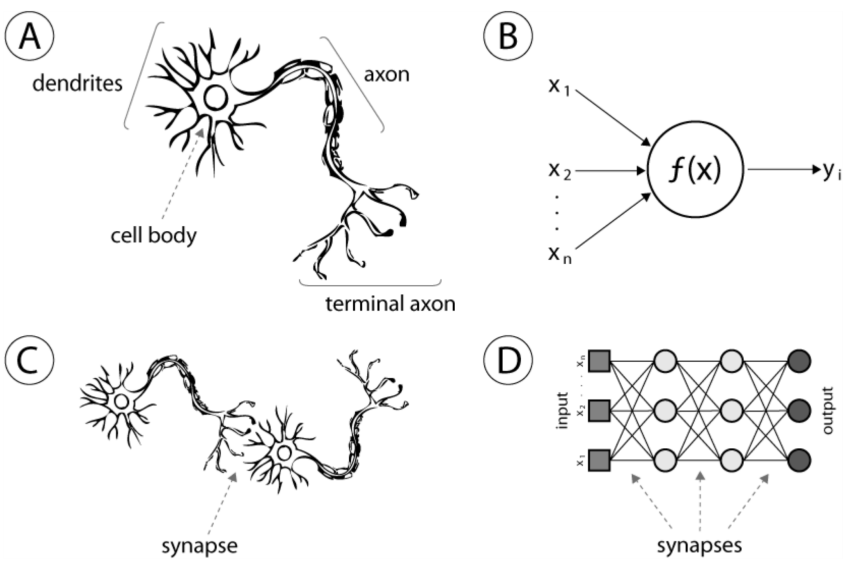

Figure 2.15: Comparison of biological neural networks with artifical neural networks. This image is by Felipe Perucho and licensed under CC-BY 3.0

Explicitly, this process starts with a neuron receiving an input vector (feature vector) and then processing it by performing a certain calculation using an ***activation function*** which adds ***non-linearlity*** to it to produce an output (Erhan et al., 2009; Olah et al., 2017). Without activation, neural output signals would simply be linear functions unable to learn complex functional mappings from big non-linear datasets like images, video and speech. The most commonly adopted activation functions include Sigmoid or logistic functions, Tanh - Hyperbolic tangent function and ReLu - Rectified Linear units function. Upon activation, the resulting neural output value is then multiplied by a ***connection weight*** followed by addition of a ***bias value*** before becoming an input to other connected neurons in the next layer for new activation. *Weight values* simply signify the importance/ strength of a connection between two nodes by adjusting the steepness of the curve representing the activation function (see figure 2.16). This means that if the connection weight value from neuron 1 to neuron 2 is high, neuron 1 has a great influence over neuron 2. On the other hand, the *bias value* allows the activation function to better fit the data by applying a left or right offset to the activation function curve (see figure 2.17). The bias value is always 1 but can be adjusted accordingly using the ***bias weight***.

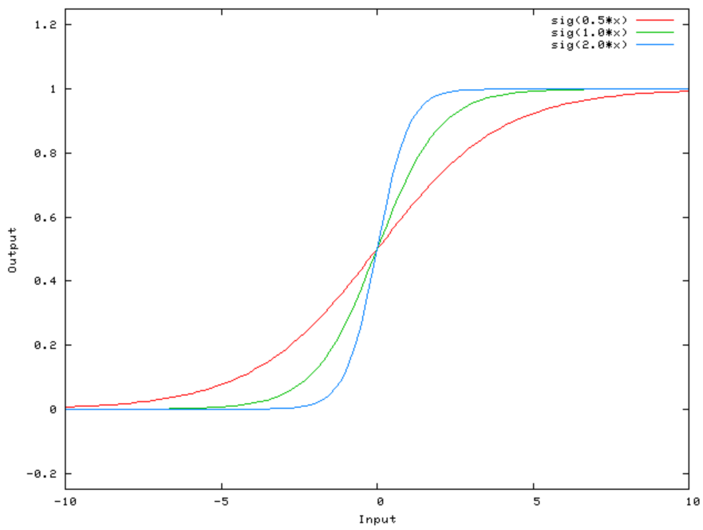

Figure 2.16: Three sigmoid curves - same inputs, different connection weights and no baises. The output sigmoid function is given by *sig(W_0*x)* where x is the neural input and W_0 is the weight (Kohl, 2010).

2.6.3.2 Learning process of an artificial neural network

Neural calculations can be learned using a combination of ***(un)supervised learning*** and ***back-propagation*** (Rumelhart et al., 2013). The process of supervised learning starts by giving the ANN a large amount of example datasets (analogous to questions provided with correct answers) with a goal for it to try and guess the same answer as the one provided in the examples. Initially, the neural calculations output wrong guesses through *forward propagation* which error is computed and via back-propagation, the steps initially taken in the ANN are revisited while slowly adjusting the synapse weight values and neural calculations with the to gradually reduce the prediction error on the next forward propagation. This process is done iteratively using thousands or even millions of examples until the ANN

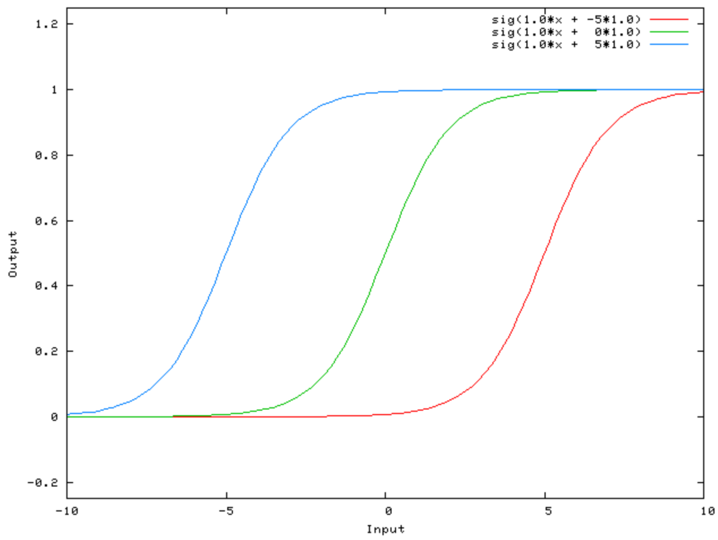

Figure 2.17: Three sigmoid curves - same inputs, same connection weights and different bais weights. The output sigmoid function is given by *sig(W_0*x + W_1*1.0)* where x is the neural input, W_0 is the weight and W_1 is the bias weight value. Bias is always 1 (Kohl, 2010).

has learnt to guess the right answer on new unseen data (generalization).

2.6.3.3 Deep Learning

Previously when computing power was not as advanced as it is now, a neural network had only one input layer, *one hidden layer* and one output layer however with increasing levels of abstraction and advances in parallel computing (Krizhevsky et al., 2012), *more hidden layers* can be added to the network forming a ***deep neural network*** which is the basis of ***deep learning*** as shown in figure 2.18. Each hidden layer learns to transform its input data into a more abstract representation with an objective of progressively extracting higher level features from raw input data depending on the design of each layer (Erhan et al., 2009; Olah et al., 2017). The advent of deep learning has led to the development of different variants of specialized neural networks for example Convolutional Neural Networks (CNN) (Cun et al., 1990; Krizhevsky et al., 2012; Le, 2013; Lowe, 1999) which are heavily adopted in the field of computer vision and Recurrent Neural Networks (RNN) for context-aware tasks like Natural language processing (Graves et al., 2013) and reinforcement learning that require a sense of built-in memory within the network.

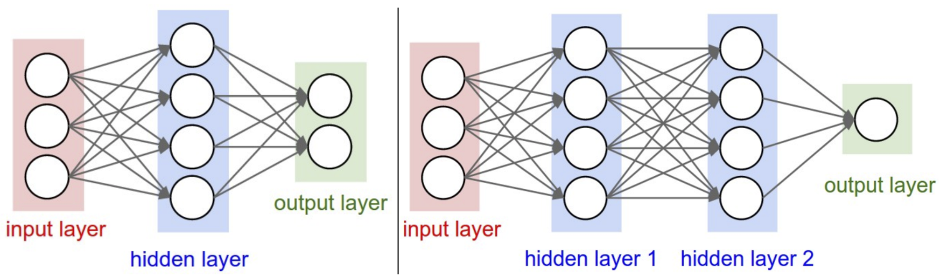

Figure 2.18: **Left:** A 2-layer neural Network (one hidden layer of 4 neurons and one output layer with 2 neurons), and three inputs. **Right:** A 3-layer deep neural network with three inputs, two hidden layers of 4 neurons each and one output layer.

2.6.3.4 Deep Q-Learning

As introduced in subsection 2.6.3, a deep neural network can be used to estimate Q-values and approximate an optimal Q-function through a process called deep Q-learning. The resulting deep Q network (DQN) (Mnih et al., 2015) accepts as input, the states from a given environment from which the it (DQN) outputs estimated Q-values for each action that can be taken from that state (see figure 2.19).

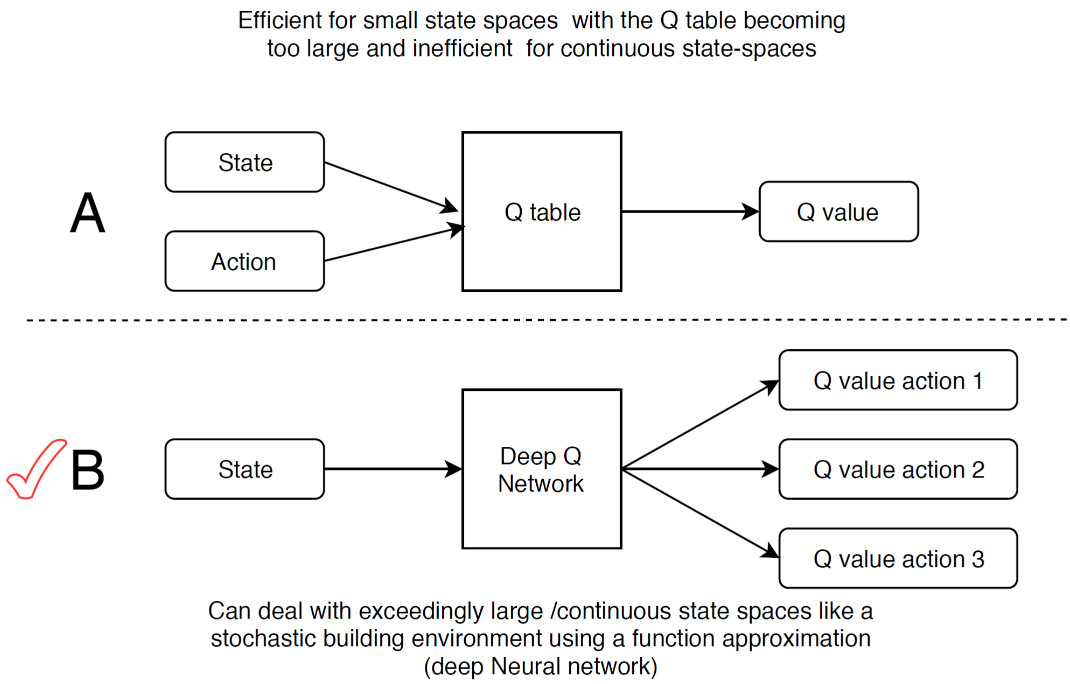

Figure 2.19: A comparison between Deep Q-learning and normal Q-learning

2.6.4 Knowledge graphs as the data model for machine learning

The potential for semantic web knowledge graphs in machine learning is something hypothetical that current research is trying to answer. Wilcke et al. (2017) reviewed the potential that such heterogeneous knowledge containers have in availing end-to-end learning for machine learning models. By being able to model incomplete knowledge using the open world assumption (Berners-Lee et al., 2001b,a), knowledge graphs are well suited for modelling real world data in a machine learning (ML) setting without being concerned how incomplete knowledge should be dealt with as is the case for many traditional ML workflows using K-nearest neighbours (Cunningham and Delany, 2007), multiple amputation (Sterne et al., 2009) and maximum likelihood methods (Allison, 2012) as discussed by Priya et al. (2015). The second argument relates to how knowledge graphs have the flexibility of representing implied facts from explicitly declared knowledge without the need to include the implied statements in the RDF data model. This means that as a learning model, a knowledge graph can achieve high levels of semantic expressivity without being redundant, overly large and complex at the expense of representing many facts.

The emergence of deep learning models has paved way for workflows that deal with extremely large raw data to automatically learn relevant features without the need for too much pre-processing for example, Convolutional Neural Networks for image processing (Cun et al., 1990; Krizhevsky et al., 2012; Le, 2013; Lowe, 1999) and audio analysis using Natural Language Processing (NLP) (Graves et al., 2013) without the need for POS-tagging and parsing (Nguyen and Grishman, 2015). All these deep learning can achieve state-of-the art learning performance when fed with the aforementioned raw data that contains all relevant and irrelevant information however, it is important to note that this information is all domain-specific (images, sound, language). When faced with heterogeneous knowledge, deep learning models struggle and often rely on manual pre-processing, a step at which a lot of vital learning information (hidden relationships) can be lost (Wilcke et al., 2017).

Recently, the machine learning community has taken keen interest in making the knowledge graph part of the learning process. Some methods still require a great deal of pre-processing while while others try to work with knowledge graphs more naturally (Wilcke et al., 2017). The former include graph embeddings which use substructure counting graph kernels (Lösch et al., 2012) to generate feature vectors from knowledge graphs in a fashion similar to k-neighbourhood methods in Cunningham and Delany (2007). A drawback of these substructure counting methods is that the size of the feature vector grows with the size of the data which led to a proposal of RDF2Vec (Ristoski and Paulheim, 2016) that deals with large graphs more efficiently. More natural workflows of dealing with knowledge graphs include representing RDF triples as a 3$^{\text{rd}}$ order tensor and adopting Graph Convolution Networks (Kipf and Welling, 2016) to model relational data in knowledge bases as described by Schlichtkrull et al. (2017).

Nickel et al. (2016) provides a very comprehensive review on the use Statistical Relational Learning (SRL) on knowledge graphs. Traditionally, machine learning methods take as input a feature vector, which represents an object in numerical or categorical terms as basis for learning a function that maps this vector to an output. This review discusses SRL methods that can work with object representations with embedded relationships to other objects like knowledge graphs. The main goal of SRL is the prediction of missing edges, nodes and node clustering based on connectivity patterns and this is the focus of Nickel et al. (2016)'s discussion where they present how SRL methods can be applied to existing knowledge graphs to learn a model that can predict new facts (edges) given existing facts.

2.6.5 Current applications of RL in building energy management

There have been quite many successful applications of RL to HVAC control starting as early as 1996 with Anderson et al. (1996) applying Q-learning to a Proportional Integral (PI) controller to modulate the output of the PI controller for a heating coil. The PI controller performed better with the RL agent. Liu and Henze (2006) combined Q-learning with Model Predictive Control (MPC) to control the HVAC operation at the Energy Resource Station Laboratory building in Iowa. This control architecture worked better than MPC and Q-learning in isolation. Du and Fei (2008) applied an actor-critic neural network with a PID HVAC controller and realized significant improvements in the energy performance for both heating and cooling. In 2010, Yu and Dexter (2010) proposed a model-free RL scheme with fuzzy discretization of the state-space variables to tune an HVAC controller online. Urieli and Stone (2013) and Ruelens et al. (2015) designed adaptive reinforcement learning agents that utilize a tree search look ahead to apply new control strategy for a heat-pump thermostat. Urieli and Stone (2013)'s work resulted in 7-14% energy savings and 4-9% for Ruelens et al. (2015) compared to rule-based systems. Barrett and Linder (2015) applied Q-learning in HVAC control with Bayesian Learning for occupancy prediction and resulted in 10% energy savings over a programmed controller.

Jia et al. (2019) proposed a framework that uses deep RL to automatically learn building energy control strategies using simulation data availed by 'Energy plus' (Crawley et al., 2001). Although EnergyPlus can run powerful simulations, it has limited capabilities for algorithm development making it difficult to implement RL-based control strategies within its environment (Jia et al., 2019). A co-simulation approach[14] via 'Building Controls Virtual Test Bed (BCVTB)' (Wetter, 2008, 2011) had to be employed to enable RL-algorithms developed in python to be tested on the Energy Plus Models. Chen et al. (2018b, 2019a,b) developed a control system that coordinates natural ventilation with the operation of HVAC systems using the reinforcement learning approach specifically via model free Q-learning. This system evaluates both the indoor and outdoor temperature and responds with the best control decision. Zhang and Lam (2018) utilized a physics-based model for a heating system to train a DRL agent which was deployed in an actual heating system and a smartphone app which lets occupants submit their thermal preferences to the DRL agent. For this work, it was found that the DRL agent saved 16.6 -18.2% heating demand. Zhang et al. (2018) also proposed a novel DRL framework to use a building energy model (BEM) for model-based optimal control of building energy and achieved a 15% energy saving by controlling the heating system supply water temperature with the prototype system. Wei et al. (2017) developed a data-driven approach that utilizes DRL to intelligently learn the effective strategy for operating building HVAC systems while maintaining acceptable indoor thermal comfort. Several other efforts for using DRL for optimal control of low energy buildings can be found in Chen et al. (2018a); Gao et al. (2019); Lu et al. (2019b); Yang et al. (2015) with more comprehensive reviews provided by Han et al. (2018) and Mason and Grijalva (2019)

Much as there have been so many research efforts to semantically enrich building information using RDF, very little work have been done to assess how well this data model performs in a deep Reinforcement Learning setting for HVAC controllers towards more adaptive and context-aware building control. This is the basis for most of the work in this thesis.

[14]Co-simulation is a simulation methodology that allows users to couple simulation software together and simulation software with actual hardware while collaboratively exchanging information between each other.

2.6.5.1 Research gap identified

The review above has made it evident that one aspect that is not well researched while using machine learning for building energy optimization is the use of semantic web technologies and Linked Data structures (RDF) to support the learning process of the DRL agents. It is still yet to be determined up to what extent DRL agents in HVAC controllers can utilize the semantic richness availed by knowlegde graphs towards improved end-to-end learning and context-aware control of buildings.

2.6.5.2 Summary

The journey towards smart energy efficient buildings is paved with complex cyber-physical-human systems operating in a very stochastic environment of changing occupancy behaviour, variable indoor-outdoor temperatures and different degrees of occupant thermal comfort among others.

Uncertain/ latent relationships between the HVAC systems and this stochastic environment need to be addressed holistically within the learning process of their DRL controllers towards improved energy efficiency by reducing HVAC operational redundancy especially when the thermal comfort set-points in a building zone have already been reached.

Chapter 3

Methodology

Much of the previous chapter has put the research problem in context with various principles introduced and will therefore not be repeated in documentation of the methodology, only references will be provided where deemed relevant. A graphical flowchart summarizing the overall workflow is also be presented in figure 3.8.

3.1 Methodology breakdown

Because this work is going to refer to several other ontologies that are readily available[1], extensible and managed by W3C[2] to model building information, section 3.2 will briefly provide highlights of each, a choice justification plus a summary of the classes and properties to be adopted. The tools for working with these ontologies and extending them will also be presented including the RDF serialization format used for storing the generated triples. Also, the query languages and rules used to retrieve specific information from the RDF triple store (knowledge base) will also be presented.

The next section 3.3 will briefly introduce the modelling approach adopted for the deep reinforcement learning (DRL) problem, followed by a discussion of how the knowledge graph in section 3.2 is incorporated into the DRL model. A delineation of the modelling tools, programming languages and software used in developing the DRL algorithm is also provided in subsection 3.3.4.

Finally, the planned implementation strategy of encapsulating the overall learning framework into a working prototype that can be tested and validated will be provided in section 3.4.

[1] Re-use of ontologies is very important to keep a certain level of homogeneity and it is often encouraged to create extensions basing on already existing ontologies rather than create new vocabulary from scratch. Their nature favours re-use and extensions unlike the IFC schema which is fixed.

[2] https://www.w3.org/community/lbd/

3.2 ERLO ontology development

As the overarching goal of this research is to work with building data in the form of RDF triples, it is essential to use already existing ontologies because they are open and extensible rather than create new ones from scratch (see footnote 1). The IFC schema already exists in RDF format via ifcOWL but since this is a very large ontology encapsulating everything about a building, it will not be used for this work in its entirety. Rather, smaller modules[3] are adopted, extended and integrated accordingly as shown in figure 3.1. For this energy optimization problem, core emphasis is placed on the concepts already presented in subsection 1.6.4 as summarized below;

- Building topology and geometry.
- Sensors in a building that monitor air quality (Carbon dioxide buildup), indoor and outdoor temperature and occupancy activity.
- Heating and cooling energy consuming equipment, their properties and usage schedules.

3.2.1 Modular ontologies used

The choice of modular ontologies used to model ERLO (Energy Reinforcement Learning Ontology) is based on their efficacy for modelling the information highlighted in subsection 1.6.4 and summarized in section 3.2 above. To this effect, the ontologies below are adopted.

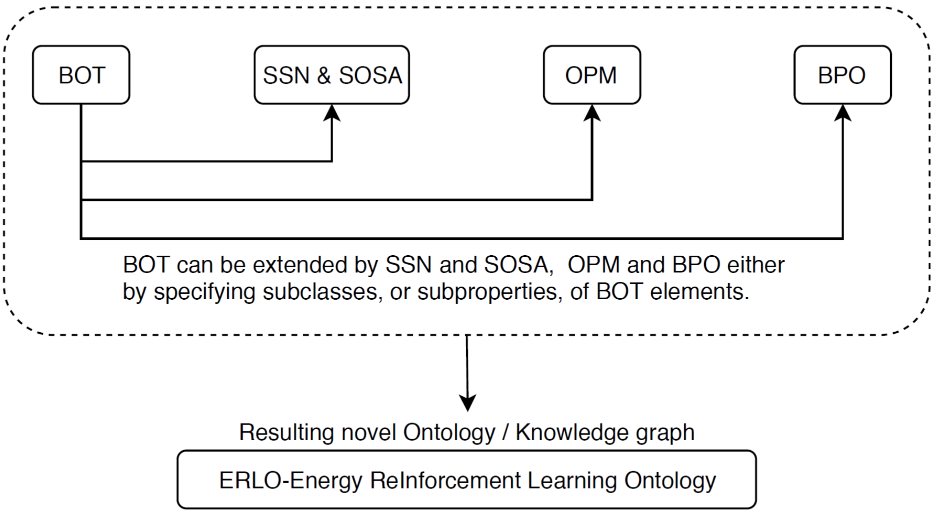

Figure 3.1: Development of ERLO by extending the core ontology BOT with SSN, SOSA, OPM and BPO

[3]Ontology modularization is a common method used in ontology engineering to segment an ontology into smaller parts. Ontology modularization aims at providing users of ontologies with the knowledge they require, reducing the scope as much as possible to what is strictly necessary in a given use case.

3.2.1.1 Building Topology Ontology (BOT)

BOT[4] is the key ontology to define relationships between the sub-components of a building (Rasmussen et al., 2017b, 2019). It is chosen because of its simplicity and adaptability to existing and non-existing buildings. In BOT, a building consists of zones in a hierarchy. The subclass of a zone is a site which contains a building(s), storey(s), and a space(s). A zone can be adjacent to another zone or even contain other zones. It can also be bounded by physical building elements or even contain them. Building elements can also host other elements for example a wall can host a door. The classes from BOT to be utilized throughout this research are summarized in table 3.1. The goal is to extend BOT either by specifying sub classes, or sub properties, of BOT elements. Examples of such an extension with sosa is shown in listing 3.1 and listing 3.2.

Classes (domain)	Properties	Classes(range)
bot:Zone	bot:containsZone	bot:Zone
	bot:adjascentZone	bot:Zone
bot:Site	bot:hasBuilding	bot:Building
bot:Building	bot:hasStorey	bot:Storey
bot:Storey	bot:hasSpace	bot:Space
bot:Space	bot:containsElement	bot:Element
bot:Element	bot:hostsElement	bot:Element

Table 3.1: BOT classes and properties to be adopted

```
@prefix bot:  <https://w3id.org/bot#> .
@prefix sosa: <http://www.w3.org/ns/sosa/> .
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl:  <http://www.w3.org/2002/07/owl#> .

sosa:Sensor a owl:Class ;
rdfs:subClassOf bot:Element .

```

listing 3.1: An example of bot:Element being extended by SOSA element (sosa:Sensor) using the rdfs:subClassOf property

```
@prefix bot:  <https://w3id.org/bot#> .
@prefix sosa: <http://www.w3.org/ns/sosa/> .
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl:  <http://www.w3.org/2002/07/owl#> .

sosa:Sensor a owl:Class .
sosa:isObservedBy a owl:ObjectProperty ;
rdfs:subPropertyOf bot:containsElement ;
rdfs:range sosa:Sensor .

```

listing 3.2: An example of a bot property bot:containsElement being extended by SOSA property (sosa:isObservedBy) using the rdfs:subPropertyOf property

[4] https://github.com/w3c-lbd-cg/bot(accessed26-6-2019)

3.2.1.2 Semantic Sensor Ontology (SSN)

SSN[5] is chosen for describing sensors and their observations, properties as well as actuators. At its core, it includes a lightweight but self-contained ontology **(Sensor, Observation, Sample and Actuator (SOSA))** for its elementary classes (Haller et al., 2017). This ontology is chosen because of its ability to support a wide range of applications and use cases since the optimization problem at hand is influenced by both parameters from indoor and outdoor. SSN is also a simpler alternative to the larger SSNO (Semantic Sensor Network Ontology)[6] as it offers several smaller ontology subsets that can be integrated only when needed via modularization. This modularization of SSN can be via both vertical or horizontal segmentation using the *'owl:import'* statement. Via vertical segmentation, vertical modules build upon each other and directionally import lower level modules. This makes lower level modules independent of the higher ones. Horizontal segmentation allows modules to be imported at the same level which means that they may depend on each other. Some of the classes and properties that will be used from this ontology are summarized in the table 3.2. Class restrictions, property ranges and domains are adopted from (Haller et al., 2017) but not presented here (subject to change with progress). See listing 3.1 and listing 3.2 for an example extension with BOT.

Classes	Properties
sosa:Observation	sosa:observedProperty
sosa:Sensor	sosa:observes
sosa:ActuatableProperty	sosa:isObservedBy
ssn:Stimulus	sosa:madeObservation
sosa:Actuation	sosa:madeBySensor
sosa:Sampling	ssn:isProxyFor
sosa:FeatureOfInterest	ssn:wasOriginatedBy
sosa:ObservableProperty	ssn:detects
sosa:Sample	sosa:actsOnProperty
sosa:Actuator	sosa:isActedOnBy
ssn:Stimulus	sosa:madeActuation
sosa:Sampler	sosa:madeByActuator
ssn:Property	sosa:isSampleOf
ssn:Stimulus	sosa:hasSample
	sosa:hasFeatureOfInterest
	sosa:isFeatureOfInterestOf
	ssn:hasProperty
	ssn:isPropertyOf
	sosa:hasSample

Table 3.2: SSN and SOSA classes and properties to be adopted

[5] https://www.w3.org/TR/vocab-ssn/
[6] https://www.w3.org/TR/vocab-ssn/#bib-SSNO

3.2.1.3 Ontology for Property Management (OPM)

In addition to SSN and SOSA, the OPM ontology (Rasmussen et al., 2018) will be adopted to model properties that evolve over time because an optimization strategy valid at one point in time might be invalid in the future. OPM coupled with SSN and SOSA can capture such validity information in the knowledge base. Since this ontology is going to be adopted as a supplement to SSN and SOSA, the classes and properties to be utilized from this ontology will depend on the required mergence with SSN and SOSA. An example is shown in listing 3.3 declaring <someROOM> as a bot:Space and assigning an area property via opm:hasPropertyState, opm:Assumed and opm:CurrentPropertyState. This <someROOM> can now contain a sensor via the bot:containsElement property since sosa:Sensor has already been declared in listing 3.1 as a subclass of bot:Element. This means that the sensor in <someROOM> can now benefit from the existing relationship of area assigned by OPM.

```
@prefix cdt:   <http://w3id.org/lindt/custom_datatypes#> .
@prefix ex:    <https://example.org/opmTest/> .
@prefix bot:  <https://w3id.org/bot#> .
@prefix opm:   <https://w3id.org/opm#> .
@prefix prov:  <http://www.w3.org/ns/prov#> .
@prefix props: <https://w3id.org/product/props/> .
@prefix xsd:   <http://www.w3.org/2001/XMLSchema#> .

<someROOM> a bot:Space;
props:area ex:property_efafede9-53d3-425f-89f3-ed8a40b5995a .

ex:property_efafede9-53d3-425f-89f3-ed8a40b5995a
a opm:Property ;
opm:hasPropertyState ex:state_8e889bba-bc25-4b1a-ac34-330f776892fa .

ex:state_8e889bba-bc25-4b1a-ac34-330f776892fa
a opm:Assumed , opm:CurrentPropertyState ;
schema:value "20 m2"^^cdt:area ;
prov:generatedAtTime "2018-05-28T16:41:17.711+02:00"^^xsd#dateTime .' }

```

listing 3.3: An example of a bot property bot:containsElement being extended by SOSA using the rdfs:subPropertyOf property

3.2.1.4 Building Product Ontology (BPO)

This ontology will be utilized to define concepts of describing building elements in a schematic way not including any geometry but with the ability to attach properties to any component without restricting their types. At the time of writing, this ontology contains three subdivisions for modelling such information namely; Building Elements, MEP and Furniture (Wagner et al., 2019; W3C-Linked Data Community Group, 2018b). This ontology works in unison with the buildingSMART Data Dictionary (bsDD) for classification purposes and should the geometric descriptions of products be required, the File ontology for geometry formats (FOG) can be employed as an extension. BPO is a perfect match for modelling information about energy consuming equipment (e.g. air conditioning units, heaters, cooking equipment etc) in a building without attaching unnecessary geometry as is the case with IFC.

ERLO will encapsulate all the concepts deemed necessary to represent a building energy model that can accurately capture the problem state-space for the deep reinforcement learning agent. **Protégé**[7] (a free, open-source ontology editor) will be used in the development of ERLO to manipulate and extend the modular ontologies presented serializing[8] the resulting knowledge graph (ERLO) using the Turtle format (*.ttl*) due to its good human readability and parsing[9] speed (see subsection 2.4.3 for more details). **StarDog**[10] will be adopted as the RDF triple store as it has inbuilt reasoning support for RDFS, OWL flavours and rule languages that can perform ontology consistency checks but most important, it is embedded with powerful query engines like SPARQL and GEOSPARQL. Communications with the RDF triple store will be made using a SPARQL endpoint[11] via HTTP. It will follow the basic architecture in figure 3.2 with the adoption of stardog's HTTP method in table 3.3. An example of a typical SPARQL query is shown in listing 3.4.

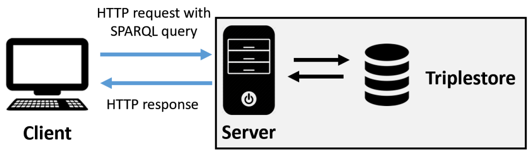

Figure 3.2: The communication protocol between the SPARQL query and the triple store.

Triple Store	Type	HTTP method	Default local endpoint URL
StargDog	Read	GET/POST	http://localhost:5820/MyDB/query
	Update	POST	http://localhost:5820/MyDB/update

Table 3.3: StarDog's SPARQL communication protocol to be adopted

```
PREFIX erlo: <http://unmcsharepoint/KevinLM/energyRLO/erlo.ttl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
FROM <http://unmcsharepoint/KevinLM/energyRLO/erlo.ttl>
SELECT ?ACunits ?sensor ?value
WHERE {
?erlo:ACunits rdf:type ?erlo:BuildingEquipment .
?erlo:erlo:Sensor erlo:readsFrom ?erlo:ACunits .
?erlo:Sensor erlo:hasValue ?erlo:AboveThresholdSensorValue .
}

```

listing 3.4: A basic SPARQL query retrieving AC units whose sensor value is above a certain set threshold. No specific query modifiers have been defined

[7] https://protege.stanford.edu/
[8] Process of translating the RDF ontology graph structures into a simple format that can be stored. For this research, the format taken is turtle (.ttl)
[9] Parsing is the opposite of serialization i.e. The process of reading a stored turtle file and writing /converting it back to graph format
[10] https://www.stardog.com/
[11] An endpoint is an entry point for HTTP access to shared RDF data using SPARQL as a query language

3.3 Algorithm development and training

3.3.1 Modelling of the learning problem

The reinforcement learning problem is going to be modelled as a Finite Markov Decision Process (MDP) whose underlying concepts have already been presented in subsection 2.6.2. A typical HVAC system is operated to maintain a desired temperature within a zone based on current indoor-outdoor temperature. The zone temperature at the next time step depends only on the current building state and the conditioned air from the HVAC system which is independent from the previous states of the building. This validates why HVAC control is being modelled as an MPD.

Because this is assumed to be continuous learning problem with no explicit terminal state, value-based deep Q-learning is chosen as it is best suited for continuously estimating the rewards the learning agent gets at each time step. See subsection 2.6.3 and figure 3.3 for the summarized overview.

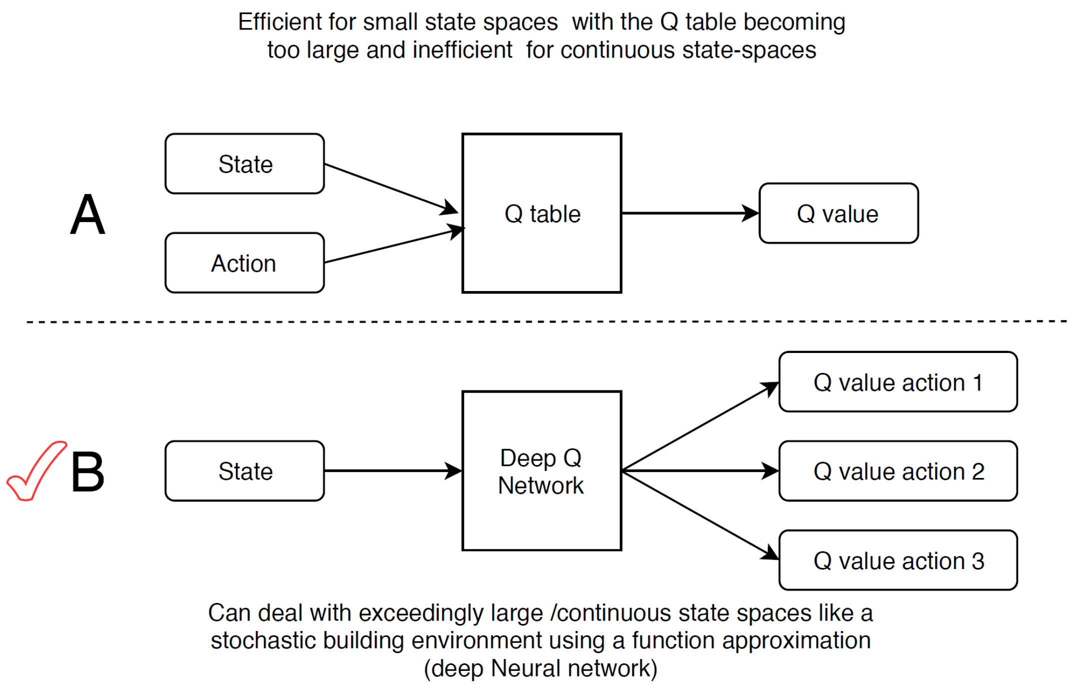

Figure 3.3: A comparison between Deep Q-learning and normal Q-learning

3.3.1.1 HVAC Control action space

This research considers an office building with z temperature zones equipped by a VRF (Variable Refrigerant Flow) HVAC system. In each zone, the HVAC system can vary the temperature by choosing from a set of discrete temperatures $\mathrm{T} = \mathrm{t}^1,\mathrm{t}^2,\mathrm{t}^3,...,\mathrm{t}^\mathrm{m}$.

The corresponding action space of the HVAC control system in its entirety $\mathrm{A} = \mathrm{A}^1,\mathrm{A}^2,\mathrm{A}^3,...,\mathrm{A}^\mathrm{n}$ includes all possible combinations of temperatures in every zone where $\mathrm{n} = \mathrm{m}^\mathrm{z}$.

3.3.1.2 System or building state space

The HVAV optimal control actions are dependent on the observation of the current system state. The states this research is considering are building zone occupancy, zone temperature, outdoor ambient temperature and current physical time. The concept of time can hold valuable information about electricity price changes, occupant activities, equipment operation scheduling and variable weather data along the year.

3.3.1.3 Reward function

The goal the DRL control system is to maximize reward in by minimizing energy cost while maintaining optimal thermal comfort. Because the goal of minimizing energy cost conflicts with the goal of maintaining thermal comfort, the immediate reward function r will be designed to take this into account. During HVAC operation, the maximum cumulative reward the control system can get by taking an action a in state s is given by the optimallity value from equation (3.1).

$$Q^{\pi^*}(s,a) = \mathbb{E}[r_{t+1} + \gamma \max_{a'} Q^{\pi^*}(s',a')|s,a,\pi^*] \tag{3.1}$$

State transition is very stochastic in building environments and hard to measure. Q-learning together with a deep neural network is adopted to estimate the value function calculated by (3.2).

$$Q^{new}(s_t,a_t) = (1-\alpha)\underbrace{Q(s_t,a_t)}_{\text{old value}} + \alpha\overbrace{\left(r_{t+1} + \gamma \max_{a'} Q(s',a')\right)}^{\text{learned value}} \tag{3.2}$$

3.3.2 Modelling the input tensor for the system state space using a knowledge graph (ERLO)

A ***tensor*** is simply a vector or matrix of n-dimensions that represents all types of data. All values within a tensor hold identical data type with a known shape. Within Tensorflow, all computations involve tensors and for input into the a neural network, ***feature vectors*** are typically used to populate a tensor. These values flow within the network graph while undergoing different transformations until an output is produced.

This research is adopting the ***knowledge graph*** from subsection 3.2.1 as the primary input to populate the tensor representing the state space introduced in subsubsection 3.3.1.2 instead of conventional feature vectors (see figure 3.5). An RDF triple has three elements and a ***third-order tensor*** can be used to map them: two orders for entities (subject and object) and another order for the predicate/ properties with The intersection of all three orders represents a single statement (see figure 3.4).

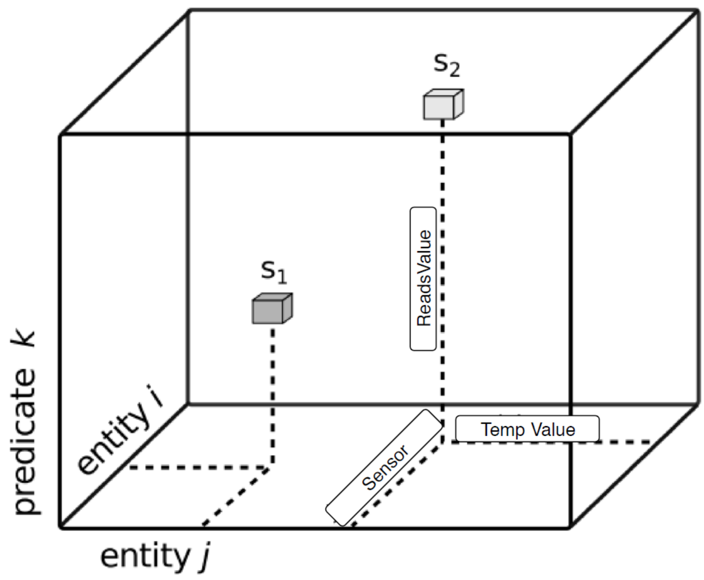

Figure 3.4: Third order tensor representation of RDF statements

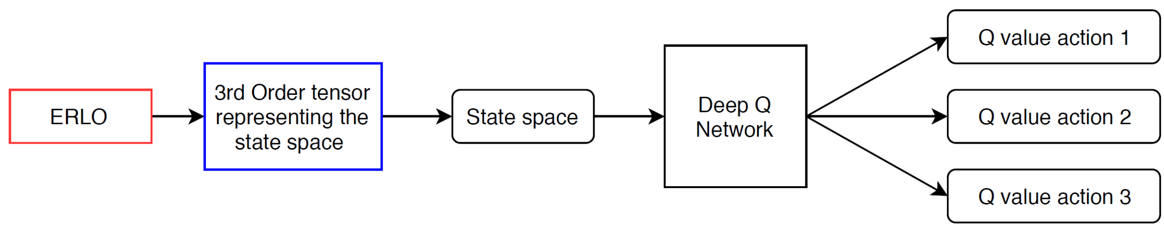

Figure 3.5: Incorporation of ERLO into the learning process

This tensor representation holds all possible combinations of RDF triples even those which are false. To deal with this issue, principles from an adjacency matrices are used: the value at any intersection holds the truth value of that RDF statement (1.0 if it is true, 0 otherwise). ***Tensor decomposition*** is used to predict which unknown statements might be true by decomposing a tensor into multiple second order vectors from which latent (hidden) features emerge. These tensors are multiplied to create an estimate of the original tensor however, some intersection values that were 0 will now have a value between 0 and 1. The efficacy of this workflow in availing end-to-end learning within smart HVAC controllers is the main hypothesis that this research is assessing to quantify any room for improvements in optimizing the cooling demand of buildings using better adaptive control.

3.3.3 The overall learning framework

Because deep Q-learning is a model-free learning system, it relies only on the state space of the environment and a reward function to choose optimal actions that maximize the long term cummulative reward in (3.1). This process starts with random exploration of the state-action space while gradually exploiting rewarding actions more frequently as the learning progresses.

This research is adopting an offline training approach that is data-driven using experiments with detailed EnergyPlus models encapsulating real weather data parameters, pricing data, occupancy schedules and zone usage. The EnergyPlus models are fine-tuned using Bayesian calibration before being adopted for training the algorithm. Because Energy Plus does not support algorithm development, it will be loosely coupled with Simulink that allows integration of algorithmic control. This is done within a programmable co-simulation Building Control Virtual Test Bed (BCVTB)[12](Wetter, 2008).

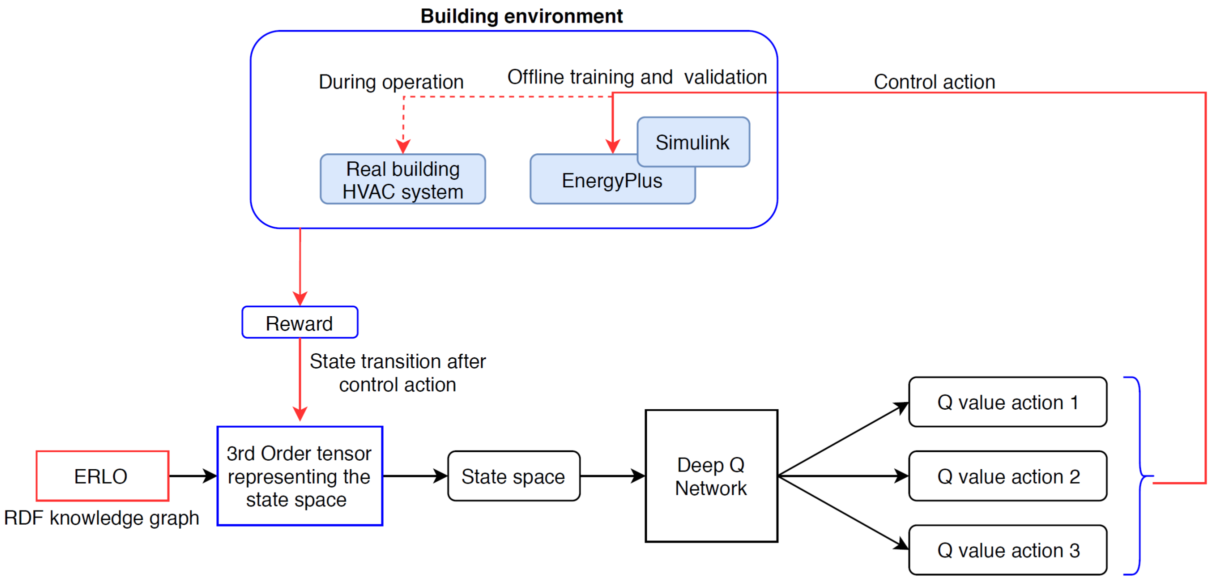

Figure 3.6: Third order tensor representation of RDF statements

The DRL algorithm must be tested and validated outside the simulation environment. The prototype adopted for this task is presented in section 3.4

3.3.4 Modelling tools used

The deep reinforcement learning model will be implemented using python while adopting **TensorFlow**[13] which has a comprehensive and flexible ecosystem of tools and libraries to be utilized for complex machine learning optimization problems. Specifically at the core, the library **Keras**[14], a high-level neural network API capable of running on top of TensorFlow will be utilized to develop and train the deep Q network. Like the modular ontologies presented in section 3.2, Keras is also extensible, modular, user-friendly and most importantly it can be run on both the GPU (Graphics Processing Unit) and CPU (Central Processing Unit). This is especially important for this optimization research where the building state space can quickly become large and computationally expensive. Keras will therefore make it possible to benefit from powerful GPU computing capabilities (e.g. NVIDIA's parallel computing) while training the deep Q network.

[12] https://simulationresearch.lbl.gov/bcvtb/FrontPage
[13] https://www.tensorflow.org/
[14] https://www.tensorflow.org/guide/keras

3.4 Research validation

While detailed EnergyPlus are highly accurate and suitable for offline training and validation, their high complexity makes them unsuitable for real-time control. Therefore, a proof of concept needs to be developed within the context a physical building to validate the performance of the DRL controlling agent. The assessment is done in comparison with conventional rule-based HVAC control systems.

3.4.1 Proof of concept

For this task, the **'Arduino programmable circuit board (micro-controller)'**[15] is chosen as the perfect fit for deployment and augmentation of the algorithm in a real-life environment for testing. Unlike most programmable circuit boards, arduino does not need a separate piece of hardware to load code onto the board, code is simply loaded via USB in the arduino IDE. Also important for this research, the board can control and interact with a variety of sensors (light, temperature, carbon dioxide, humidity and barometric pressure). Additionally, extra pre-built circuit boards (**shields**[16]) can be added on top of arduino should extended capabilities like internet and wireless connection be required with the possibility to even make a customized shield starting with source code of an existing one. The arduino board will be deployed on a working AC unit to control and monitor the algorithm performance while assessing the energy saving potential delivered while cooling thermal zones in an office building (see figure 3.7 for schematics).

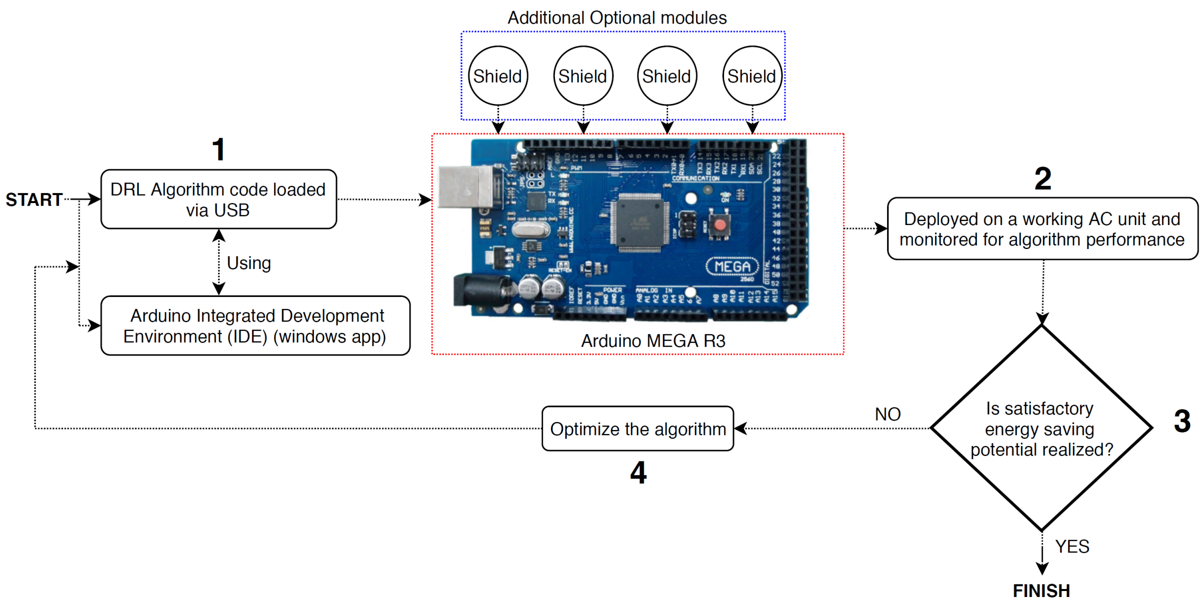

Figure 3.7: Schematic diagram of prototype for algorithm validation and testing.

[15]https://www.arduino.cc/
[16]https://www.arduino.cc/en/Main/arduinoShields

3.5 Progress

The detailed preparation that was carried out while putting the methodology together is described in section 3.2 to section 3.4 and will not be repeated here, however the training activities that supported this process are summarized below. Graduate school training activities are not included (Please see certificate).

3.5.1 Core research training activities and conferences

1. Attended and presented at the **20th World Conference on Applied Science, Engineering and technology (WCASET)** in Kuala Lumpur, Malaysia and emerged with a **best paper award** for the publication *'The role of linked building data in aligning augmented reality (AR) with sustainable construction.'- To be published* (26th June 2019)

2. Attended a week long **summer school** and **conference** themed **'Linked Data in Architecture and Construction(LDAC)' in Lisbon, Portugal**[17]. It was organized by W3C's linked building data community group[18]. The main aim of this event was to evaluate the current status of the different available ontologies to capture building data. A wide scope of domain ontologies was covered and several industry use-cases were presented with practical applications in facility management, design, construction and renovation. The summer school, on the other hand, was a hands-on experience delivered by experts and researchers on several topics e.g. ontology development exercises using RDF, RDFS and OWL, SPARQL, GEOSPARQL query languages, Apache Jena, ontology matching and software training. (19th - 22nd June 2019)

3. Participated in a **hackathon coding challenge**[19] organized by the Erasmus+ project 'Virtual Learning Factory Toolkit(VLFT)[20]' 'Stardog' and 'LDAC' in Portugal, Lisbon (**First Runner Up**). A variety of complex linked data usecases were presented to be solved by programming and specifically the authors's problem statement was **'Generation of a 3D Virtual Reality Scene from Linked Building Data'**. (17th - 18th June 2019)

4. Online course in **'Linked Data Engineering'** by Prof. Dr. Herald Stack on OpenHPI[21] covering principles of ontologies and RDF knowledge representations based on Semantic Web technologies, SPARQL query language and data modelling using RDF, RDFS and OWL (3rd - 16th May 2019)

5. Online training course in **machine learning** using **Python and C-Sharp programming languages** on UDEMY (' the Python Bootcamp'[22] and 'C-sharp master class'[23]). The core concepts

[17] http://www.linkedbuildingdata.net/ldac2019/index.html
[18] https://www.w3.org/community/lbd/
[19] https://github.com/linkedbuildingdata/SummerSchoolOfLDAC/blob/master/Notebooks/02-04-Generation-VR-Coding-Challenge.ipynb
[20] https://www.vlft.eu/
[21] https://open.hpi.de/courses/semanticweb2016
[22] https://www.udemy.com/complete-python-bootcamp/
[23] https://www.udemy.com/course/complete-csharp-masterclass/

of machine learning development using TensorFlow and Keras libraries were covered. (12^{th} April - ongoing)

6. **'Feature Manipulation Engine (FME)'**[24] training by 'Safe Software' at GIS Innovation Sdn Bhd in Subang Jaya. FME is a data manipulation software aimed towards integration and automation of worklows especially with regards to Building Information Modelling and geo-spatioal data. (22^{nd} March 2019)

7. Attended **buildingSMART**[25] webinars while using their online documentation to get aquainted with the principles of modelling building data with IFC - EXPRESS, a neutral standard and data model for exchange of building data (10^{th} November 2018 - ongoing)

3.5.2 Publications

1. The role of Linked Building Data in aligning Augmented Reality (AR) with sustainable construction. *(Accepted for publication-Best Paper Award)*

2. The indoor built environment response to haze in Malaysia. *(Accepted abstract to be presented and published as conference proceedings for the 15th International Conference on Atmospheric Sciences and Applications to Air Quality (ASAAQ 15))*

3.5.3 Future publications plan

Journal publications in the future will specifically aim at the two scopes below.

1. A 'proof of concept' ontology for training a reinforcement algorithm in the building energy context.

2. Co-simulation environment for extraction and reuse of simulation data using ontologies.

3.5.4 Summary of tools to be used for the research

1. Protege ontology editor. (*Free academic License*)

2. Stardog software for triple storing and performing SPARQL queries.(*1 year free academic license*)

3. EnergyPlus simulation software for generating data to train the algorithm. (*Free license and open-source*)

4. Python programming language for developing the algorithm. (*Free license*)

5. TensorFlow API, a python library for designing the deep learning algorithm. (*Freely available, open source and extensible*)

6. Keras python library on top of TensorFlow for designing the neural network of the algorithm. (*Freely available, open source and extensible*)

[24] https://www.safe.com/
[25] https://technical.buildingsmart.org/

7. Arduino Mega R3 micro-controller with additional shields for testing the algorithm.(*Available commercially-Exact price budget will be confirmed in due time*)

8. Arduino Integrated Development Environment (IDE) for managing and deploying code into the arduino. (*Freely available*)

9. Autodesk Revit for modellinging 3D geometry. (*3 year free academic license available*)

3.6 Projected research plan

A tabular template showing the projected duration for the various research activities is shown in table 3.4 below.

	Jan	**Feb**	**Mar**	**Apr**	**May**	**June**	**July 3rd**	**Aug**	**Sept**	**Oct 3rd**	**Nov**	**Dec**
First Year 2018										**First Year starts** Registration	Graduate Training **MES** meeting 1 (Dec) Literature Review Starts	
Second Year 2019	-Literature review and methodology development continues -**MES** Progress meeting 2 (Mar) -**MES** Progress meeting 3 (June) -buildingSMART webinars (IFC-EXPRESS) training. -LDAC conference and summer school in Lisbon Portugal (17th-22nd June). -WCASET conference, Kuala Lumpur (26th June) -Python + C sharp training, UDEMY (12th-21st April) -Linked Data Engineering course, Open HPI (3rd-16th May) -FME training by Safe software, GIS Sdn. Bhd (22nd March)						**-1st Year Annual Review** -Submit ASAAQ15- full paper (by 30th July)	**-Second Year Starts (Oct 3rd 2019)** -Prepare co-simulation and test environment -EnergyPlus BEM dev starts -Algorithm dev starts (coding) -**MES** Progress Meeting 4 (Sept) -ASAAQ15 conference (28-30 Oct)			-Ontology dev starts. -Algorithm dev (Coding) -Co-simulation further tests -**MES** Progress Meeting 5(Dec) -Journal Publication 1	
Third Year 2020	-Ontology testing -EnergyPlus BEM tests -Algorithm training tests -Co-simulation (READY)		-**MES** Progress Meeting 6 (Mar) -EnergyPlus BEM simulation (READY) -Algorithm Training Model (READY) -Ontology (READY) -Start full simulation and algorithm training. - Journal Publication 2 -Finalize Methodology Chapter		-**MES** Progress Meeting 7 (June) **-2nd Year Annual Review (July 3rd)** -**MES** Progress Meeting 8 (Sept) **-Third Year starts (Oct 3rd 2020)** -**MES** Progress Meeting 9 (Dec) -EnergyPlus BEM sim + Ontology + Algorithm training (Fully functional and running) -Arduino validation deployment is initiated. -Publication 3 (By end of July 3rd) -Publication 4 (By end of Oct) -Publication 5 (By end of Dec) -Start Results Chapter and discussion.							
Fourth Year 2021	-Algorithm training & simulation running & Arduino validation -Start validation chapter. -**MES** Progress Meeting 10 (Mar)				-**MES** Progress meeting 11 (June) -**MES** Progress meeting 12 (Sept) -Compiling the final thesis and 2 publications -Finalize by end of Oct 2021 **-THIRD YEAR ENDS (Oct 3rd 2021)**							

Table 3.4: Tentative breakdown of the research plan showing key dates. *(Future conferences and training activities are not explicitly shown but will be attended and added in due course.)*

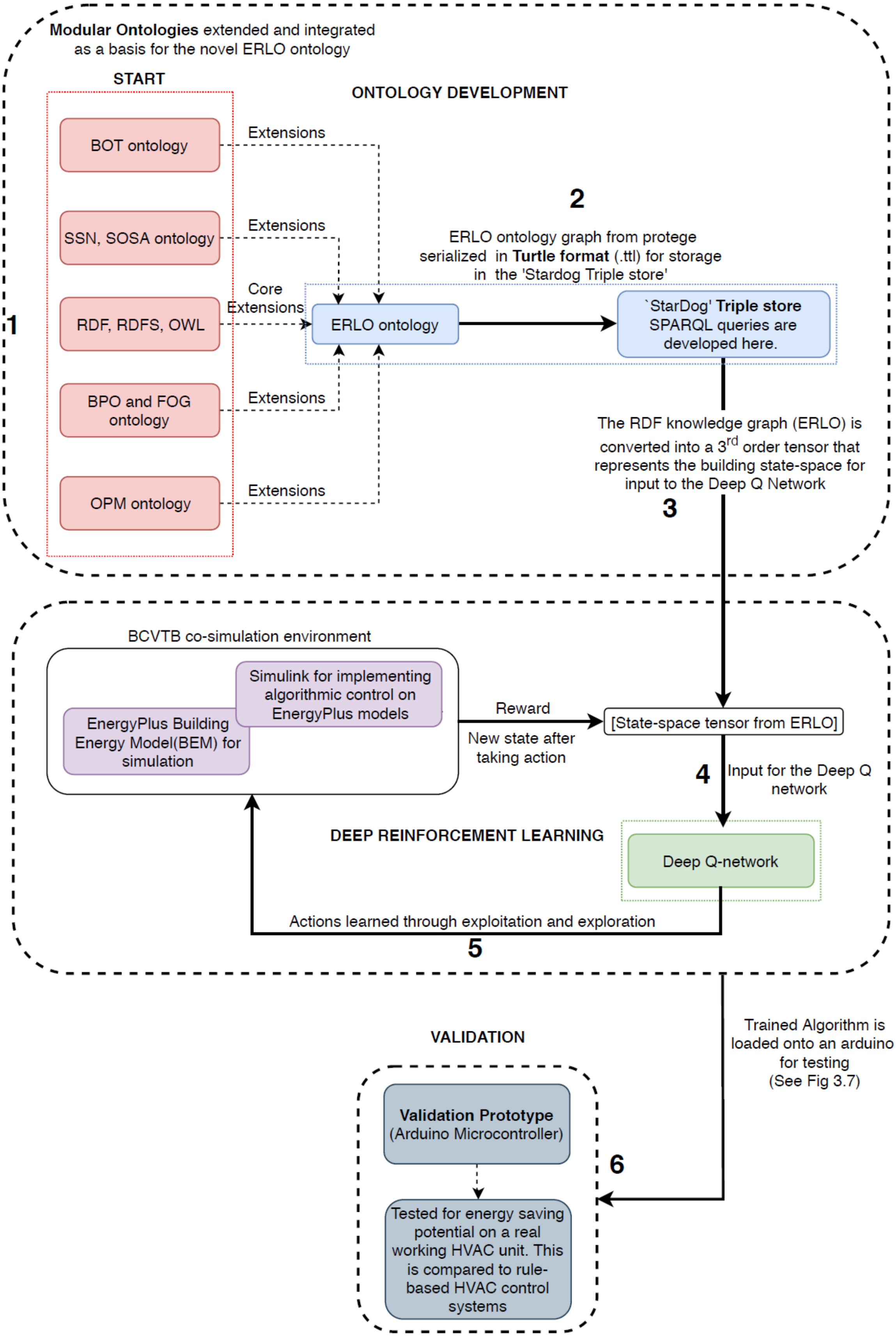

Figure 3.8: Methodology Framework

Chapter 4

Conclusion

Designing energy efficient buildings is a finicky process, requiring an exhaustive amount of thought and care, notably because of the extreme complexity of the various dynamic impact factors like occupancy behaviour, ventilation air flows patterns, outdoor and indoor temperatures, not forgetting the ever-changing weather patterns. Being a multi-objective optimization problem not only means that these impact factors are dependent with unknown relationships but also often conflicting when trying to balance reduced energy consumption with both enhanced occupant thermal comfort and acceptable indoor air quality. Static optimization systems/ building controllers are insufficient in delivering long term energy savings in such stochastic and dynamic building environments where a valid solution today might be rendered invalid in the near future due to the corresponding dynamic nature of the aforementioned impact factors. To this effect, this research intends to merge two technologies namely; 'reinforcement learning' and 'knowledge graphs (Linked Data and semantic web principles)' in development of a *' novel heuristic and iterative' energy optimization solution* for buildings capable of encapsulating the following core principles:

- Representation and utilization of building information (e.g. Sensors, geometry, product data, occupancy behaviour etc.) in *open, formalized and extensible structures using knowledge graphs and Linked Data principles*, a format in which this building information can be re-used *iteratively* in context-aware situations like machine learning for energy optimization.
- A *heuristic* machine learning system with an agent that can dynamically and continuously be trained using building information and simulation data in both 'native' and 'Linked data formats' to learn what energy optimization actions to take when interacting with an indoor building environment in different states.
- Finally, *catastrophic forgetting*[1] is a key attribute the machine learning system should avoid i.e. the prior building control actions taken by the algorithmic system should not be lost but kept and re-used heuristically and iteratively for continual improvement of the system.

[1]The tendency of an artificial neural network to completely and abruptly forget previously learning information upon learning new information.

Bibliography

Abdul-Ghafour, S., Ghodous, P., Shariat, B., and Perna, E. (2007). A Common Design-Features Ontology for Product Data Semantics Interoperability. In *IEEE/WIC/ACM International Conference on Web Intelligence*, pages 443–446. IEEE.

AEC-UK (2012). *AEC (UK) BIM Protocol - Implementing UK BIM Standards for the Architectural, Engineering and Construction Industry Version 2.0*. AEC-UK Committee.

Agostinho, C., Dutra, M., Jardim-Gonçalves, R., Ghodous, P., and Steiger-Garção, A. (2007). EXPRESS to OWL morphism: making possible to enrich ISO10303 Modules. In *14th ISPE International Conference on Concurrent Engineering*, pages 391–402, London. Springer.

Allison, P. D. (2012). Handling Missing Data by Maximum Likelihood. *SAS Global Forum 2012 Statistics and Data Analysis*, 312:1–21.

Anderson, C. W., Hittle, D. C., Katz, A. D., and Kretchmar, R. M. (1996). Reinforcement Learning, Neural Networks and PI Control Applied to a Heating Coil. In *Int. Conf. EANN*, volume 96, pages 135–142.

Anzaldi, G., Corchero, A., Wicaksono, H., Mcglinn, K., and Gerdelan, A. (2018). Knoholem: Knowledge-Based Energy Management for Public Buildings Through Holistic Information Modeling and 3D Visualization. *International Technology Robotics Applications*, 70:47–56.

Apache Jena (2009). Apache Jena - Working with RDF Streams in Apache Jena.

Arulkumaran, K., Deisenroth, M. P., Brundage, M., and Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38.

Baader, F., L. McGuinness, D., Nardi, D., and F. Patel-Schneider, P. (2003). *The Description Logic Handbook – Theory, Implementation and Applications*. Cambridge University Press, Cambridge, MA, USA.

Barbau, R., Krima, S., Rachuri, S., Narayanan, A., Fiorentini, X., Foufou, S., and Sriram, R. D. (2012). OntoSTEP: Enriching product model data using ontologies. *CAD Computer Aided Design*, 44(6):575–590.

Barrett, E. and Linder, S. (2015). Autonomous HVAC control, a reinforcement learning approach. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9286, pages 3–19. Springer Verlag.

Beckett, D. (2014). RDF 1.1 XML Syntax-W3C Recommendation 25 February 2014.

Beckett, D. and Berners-Lee, T. (2011). Turtle - Terse RDF Triple Language-W3C Team Submission 28 March 2011.

Beetz, J., J.P. Leeuwen, V., and B. Vries, D. (2005). An Ontology Web Language Notation of the Industry Foundation Classes. In *22nd CIB W78 Conference on Information Technology in Construction*, pages 193–198. Technische Universität Dresden.

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):89–101.

Bellman, R. (1957). A Markovian Decision Process. *Journal of Mathematics and Mechanics*, 6:679–684.

Berners-Lee, T. (2003). WWW Past & Future - Berners - Lee - Royal Society.

Berners-Lee, T. (2006). Linked Data - Design Issues.

Berners-Lee, T. and Connolly, D. (2011). Notation3 (N3): A readable RDF syntax-W3C Team Submission 28 March 2011.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001a). The semantic web: a brain for humankind. *IEEE Intelligent Systems*, 16(2):24–25.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001b). The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American,*, pages 1–4.

Bonino, D. and De Russis, L. (2018). DogOnt as a viable seed for semantic modeling of AEC/FM. *Semantic Web*, 9(6):763–780.

Borrmann, A., König, M., Koch, C., and Beetz, J. (2018). *Building Information Modeling Technology Foundations and Industry Practice*. Springer.

Bray, T., Hollander, D., Layman, A., Tobin, R., and S.Thompson, H. (2009). Namespaces in XML 1.0 (Third Edition)-W3C Recommendation 8 December 2009.

Brickley, D. and Guha, R. (2014). RDF Schema 1.1-W3C Recommendation 25 February 2014.

Cao, X., Dai, X., and Liu, J. (2016). Building energy-consumption status worldwide and the state-of-the-art technologies for zero-energy buildings during the past decade. *Energy and Buildings*, 128:198–213.

Chen, X., Wei, T., Chen, X., Li, X., and Zhu, Q. (2018a). Model-based and Data-driven Approaches for Building Automation and Control. *ICCAD*, pages 1–8.

Chen, Y., Norford, L. K., Samuelson, H. W., and Malkawi, A. (2018b). Optimal control of HVAC and window systems for natural ventilation through reinforcement learning. *Energy and Buildings*, 169:195–205.

Chen, Y., Tong, Z., Samuelson, H., Wu, W., and Malkawi, A. (2019a). Realizing natural ventilation potential through window control: The impact of occupant behavior. *Energy Procedia*, 158:3215–3221.

Chen, Y., Tong, Z., Wu, W., Samuelson, H., Malkawi, A., and Norford, L. (2019b). Achieving natural ventilation potential in practice: Control schemes and levels of automation. *Applied Energy*, 235:1141–1152.

Chipman, T., Liebich, T., and Weise, M. (2016). mvdXML specification 1.1, Specification of a standardized format to define and exchange Model View Definitions with Exchange Requirements and Validation Rules. By Model Support Group (MSG) of buildingSMART. *BuildingSMART Malaysia*, 1:49.

Chuck, E., Paul, T., Rafael, S., and Kathleen, L. (2011). *BIM handbook: A guide to building information modeling for owners, managers, designers, engineers, and contractors, John Wiley and Sons Inc*, volume 2. John Wiley and Sons Inc.

CIC (2015). CIC Building Information Modelling Standards (Phase One). *Construction Industry Council*, pages 1–147.

Corry, E., Pauwels, P., Hu, S., Keane, M., and O'Donnell, J. (2015). A performance assessment ontology for the environmental and energy management of buildings. *Automation in Construction*, 57:249–259.

Crawley, D. B., Lawrie, L. K., Winkelmann, F. C., Buhl, W. F., Huang, Y. J., Pedersen, C. O., Strand, R. K., Liesen, R. J., Fisher, D. E., Witte, M. J., and Glazer, J. (2001). EnergyPlus: Creating a new-generation building energy simulation program. *Energy and Buildings*, 33(4):319–331.

Cun, L., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1990). Handwritten Digit Recognition with a Back-Propagation Network. *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, 2:396—-404.

Cunningham, P. and Delany, S. J. (2007). K -Nearest Neighbour Classifiers. *Multiple Classifier Systems*, pages 1–17.

Curry, E., O'Donnell, J., and Corry, E. (2012). Building Optimisation using Scenario Modeling and Linked Data. In *First International Workshop on Linked Data in Architecture and Construction*.

Debruyne, C., McGlinn, K., McNerney, L., and O'Sullivan, D. (2017). A lightweight approach to explore, enrich and use data with a geospatial dimension with semantic web technologies. In *Proceedings of the Fourth International ACM Workshop on Managing and Mining Enriched Geo-Spatial Data*.

Dibley, M., Li, H., Miles, J., and Rezgui, Y. (2011). Towards intelligent agent based software for building related decision support. *Advanced Engineering Informatics*, 25(2):311–329.

Dibley, M., Li, H., Rezgui, Y., and Miles, J. (2012). An ontology framework for intelligent sensor-based building monitoring. *Automation in Construction*, 28:1–14.

Dolenc, M., Katranuschkov, P., Gehre, A., Kurowski, K., and Turk, Ž. (2007). The inteligrid platform for virtual organisations Interoperability. *Electronic Journal of Information Technology in Construction*, 12:459–477.

Donnell, J. O., See, R., Rose, C., Maile, T., Bazjanac, V., and Haves, P. (2011). SIMMODEL : A domain data model for whole building energy simulation. In *Proceedings of Building Simulation2011: 12th Conference of International Building Performance Simulation Association*, pages 382–389.

Du, D. and Fei, M. (2008). A two-layer networked learning control system using actor-critic neural network. *Applied Mathematics and Computation*, 205(1):26–36.

Dulac, J., Abergel, T., and Delmastro, C. (2019). Buildings.

El-Mekawy, M. (2010). *Integrating BIM and GIS for 3D City Modelling: The Case of IFC and CityGML.*

El-Mekawy, M. and Östman, A. (2010). Semantic Mapping: an Ontology Engineering Method for Integrating Building Models in IFC and CITYGML. In *Proceedings of the 3rd ISDE Digital Earth Summit.*

Elghamrawy, T. and Boukamp, F. (2008). A vision for a framework to support management and learning from construction problems. In *Proceedings of the 25th International Conference on Formation Technology in Construction: Improving the management of Construction Projects through IT adoption*, number 1517, pages 1–10.

Elghamrawy, T. and Boukamp, F. (2010). Managing construction information using RFID-based semantic contexts. *Automation in Construction*, 19(8):1056–1066.

Erhan, D., Bengio, Y., Courville, A., and Vincent, P. (2009). Visualizing Higher-Layer Features of a Deep Network. *University of Montreal*, 1341(3):1.

EUBIM Task Group (2016). Handbook for the introduction of Building Information Modelling by the European Public Sector. *EUBIM Task Group*, pages 1–84.

François-lavet, V., Henderson, P., Islam, R., Bellemare, M. G., François-lavet, V., Pineau, J., and Bellemare, M. G. (2018). An Introduction to Deep Reinforcement Learning. *Foundations and trends in machine learning*, II(3 - 4):1–140.

Gao, G., Li, J., and Wen, Y. (2019). Energy-Efficient Thermal Comfort Control in Smart Buildings via Deep Reinforcement Learning. *arXiv preprint*, pages 1–11.

Giacomo, G. D. and Lenzerini, M. (1996). TBox and ABox Reasoning in Expressive Description Logics. In *Fifth International Conference on Principles of Knowledge Representation and Reasoning (KR'96)*, number May, pages 316–327.

Gómez-Romero, J., Bobillo, F., Ros, M., Molina-Solana, M., Ruiz, M., and Martín-Bautista, M. (2015). A fuzzy extension of the semantic Building Information Model. *Automation in Construction*, 57:202–212.

Graves, A., Mohamed, A. R., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 6645–6649.

Grimm, S., Abecker, A., Völker, J., and Studer, R. (2011). Ontologies and the Semantic Web. *Handbook of Semantic Web Technologies*, pages 507–579.

Haller, A., Janowicz, K., Cox, S., Le Phuoc, D., Taylor, K., and Lefrançois, M. (2017). Semantic Sensor Network (SSN) Ontology-W3C Recommendation 19 October 2017.

Han, M., Zhang, X., Xu, L., May, R., Pan, S., and Wu, J. (2018). A review of reinforcement learning methodologies on control systems for building energy: Working papers in transport, tourism, information technology and microdata ana.

Harris, S. and Seaborne, A. (2013). SPARQL 1.1 Query Language-W3C Recommendation 21 March 2013.

Hepp, M. (2008). GoodRelations: An ontology for describing products and services offers on the web. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5268 LNAI:329–346.

Hitzler, P., Krötzsch, M., Parsia, B., F.Patel-Schneider, P., and Rudolph, S. (2012). OWL 2 Web Ontology Language Primer (Second Edition)-W3C Recommendation 11 December 2012.

ISO 10303-11 (2004). Industrial automation systems and integration – Product data representation and exchange – Part 11: Description methods: The EXPRESS language reference manual. *Geneva: International Organization for Standardization.*

ISO 10303-14 (2005). Industrial automation systems and integration - Product data representation and exchange - Part 14: Description methods: The EXPRESS-X language reference manual. *Geneva: International Organization for Standardization.*

ISO 10303-21 (2016). Industrial automation systems and integration–product data representation and exchange-Part 21: Implementation methods: Clear text encoding of the exchange structure. *ISO Central Secretariat.*

ISO 10303-22 (1999). Industrial automation systems and integration - Product data representationand exchange - Part 22: Implementation methods: Standard data access interface. *Geneva: International Organization for Standardization.*

ISO 16739:2016 (2016). Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries. *Geneva: International Organization for Standardization.*

ISO 29481-1 (2016). Building information modelling- Information delivery manual- Part 1: Methodology and format. *Geneva: International Organization for Standardization.*

Jeroen, W., Pauwels, P., and Bekers, W. (2018). *Linking Data : Semantic enrichment of the existing building geometry*. PhD thesis, Ghent University.

Jia, R., Jin, M., Sun, K., Hong, T., and Spanos, C. (2019). Advanced Building Control via Deep Reinforcement Learning. *Energy Procedia*, 158:6158–6163.

Karan, E. P. and Irizarry, J. (2015). Extending BIM interoperability to preconstruction operations using geospatial analyses and semantic web services. *Automation in Construction*, 53:1–12.

Karlshøj, J., See, R., and Davis, D. (2012). An Integrated Process for Delivering IFC Based Data Exchange. *buildingSMART International*, (1):53.

Kellogg, G. and Champin, P.-A. (2019). JSON-LD 1.1-A JSON-based Serialization for Linked Data.

Kipf, T. N. and Welling, M. (2016). Semi-Supervised Classification with Graph Convolutional Networks. *arXiv preprint*, pages 1–14.

Kofler, M. J., Reinisch, C., and Kastner, W. (2012). A semantic representation of energy-related information in future smart homes. *Energy and Buildings*, 47:169–179.

Kohl, N. (2010). Artificial intelligence - Role of Bias in Neural Networks - Stack Overflow.

Kohl, N. and Stone, P. (2004). Policy gradient reinforcement learning for fast quadrupedal locomotion. *Proceedings - IEEE International Conference on Robotics and Automation*, 2004(3):2619–2624.

Krima, S., Barbau, R., Fiorentini, X., Sudarsan, R., and Sriram, R. D. (2009). OntoSTEP : OWL-DL Ontology for OntoSTEP : OWL-DL Ontology for. In *2009 International Conference on Product Lifecycle Management*, pages 770–780.

Kris, M., Lawton, W., Wicaksono, H., Lawton, W., Weise, M., Nikolas, K., Ioanna, P., and Dimitrios, T. (2016). Identifying Use Cases and Data Requirements for BIM Based Energy Management Processes. In *CIBSE Technical Symposium*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems 25 (NIPS)*, pages 1097—-1105.

Kröse, B. J. (1995). Learning from delayed rewards. *Robotics and Autonomous Systems*, 15(4):233–235.

L. McGuiness, D. and van Harmelen, F. (2004). OWL Web Ontology Language Overview-W3C Recommendation 10 February 2004.

Le, Q. V. (2013). Building high-level features using large scale unsupervised learning. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 8595–8598.

Le, T. and David Jeong, H. (2016). Interlinking life-cycle data spaces to support decision making in highway asset management. *Automation in Construction*, 64:54–64.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Liebich, T. (2013). What's new in IFC4 ? *buildingSMART International*, pages 1–25.

Liu, S. and Henze, G. P. (2006). Experimental analysis of simulated reinforcement learning control for active and passive building thermal storage inventory: Part 1. Theoretical foundation. *Energy and Buildings*, 38(2):142–147.

Lösch, U., Bloehdorn, S., and Rettinger, A. (2012). Graph kernels for RDF data. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 7295 LNCS, pages 134–148.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157.

Lu, S., Wang, S., Hameen, E., Shi, J., and Zou, Y. (2019a). Comfort-based Integrative HVAC System With Non-intrusive Sensing In Office Buildings. In *Annual Conference of the Association for Computer-Aided Architectural Design Research in Asia-CAADRIA*, volume 1, pages 785–794.

Lu, S., Wang, W., Lin, C., and Hameen, E. C. (2019b). Data-driven simulation of a thermal comfort-based temperature set-point control with ASHRAE RP884. *Building and Environment*, 156:137–146.

Manola, F., Miller, E., and McBride, B. (2014). RDF 1.1 Primer-W3C Working Group Note 24 June 2014.

Mason, K. and Grijalva, S. (2019). A Review of Reinforcement Learning for Autonomous Building Energy Management. *arXiv e-prints*, pages 1–20.

McGlinn, K., Debruyne, C., McNerney, L., and O'Sullivan, D. (2017). Integrating building information models with authoritative Irish geospatial information. *CEUR Workshop Proceedings*, 1963:1–4.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.

Mohd Nawi, M. N., Baluch, N., and Bahauddin, A. Y. (2014). Impact of Fragmentation Issue in Construction Industry: An Overview. In *MATEC Web of Conferences*, volume 15, page 1009.

Motik, B., Cuenca Grau, B., Horrocks, I., Wu, Z., Fokoue, A., and Lutz, C. (2012). OWL 2 Web Ontology Language Profiles (Second Edition)-W3C Recommendation 11 December 2012.

MT Højgaard (2016). CAD-BIM Manual - General Part. *MT Højgaard*, 3:1–58.

NBIMS (2007). *National building information model standard version 1.0-part 1: Overview, principles, and methodologies.* National Institute of Building Sciences (NIBS).

Nguyen, T. H. and Grishman, R. (2015). Relation Extraction: Perspective from Convolutional Neural Networks. In *NAACL-HLT*, pages 39–48. Association for Computational Linguistics (ACL).

Nickel, M., Murphy, K., Tresp, V., and Gabrilovich, E. (2016). A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33.

Niles, I. and Pease, A. (2001). Towards a standard upper ontology. *Proceedings of the international conference on Formal Ontology in Information Systems - FOIS '01*, pages 2–9.

Olah, C., Mordvintsev, A., and Schubert, L. (2017). Feature Visualization. *Distill*, 2(11).

Pan, J. and Ren, Z. (2004). Potential Application of the Semantic Web. In *20th Annual Conference of the Association of Researchers in Construction Management (ARCOM), Heriot Watt University EdinBurgh*, volume 2, pages 923–929.

Pauwels, P., Corry, E., and O'Donnell, J. (2014a). Making SimModel information available as RDF graphs. *eWork and eBusiness in Architecture, Engineering and Construction*, pages 439–445.

Pauwels, P., Corry, E., and O'Donnell, J. (2014b). Representing SimModel in the Web Ontology Language. In *Computing in Civil and Building Engineering (2014)*, pages 2271–2278, Reston, VA. American Society of Civil Engineers.

Pauwels, P., De Meyer, R., Van Campenhout, J., Meyer, R. D., and Campenhout, J. V. (2010). Interoperability for the Design and Construction Industry through Semantic Web Technology. In *International Conference on Semantic and Digital Media Technologies*, pages 143–158.

Pauwels, P., Krijnen, T., Terkaj, W., and Beetz, J. (2017a). Enhancing the ifcOWL ontology with an alternative representation for geometric data. *Automation in Construction*, 80:77–94.

Pauwels, P., Kris, M., Seppo, T., and Jakob, B. (2018). Linked Data. In *Building Information Modeling Technology Foundations and Industry Practice*, pages 181–197. Springer.

Pauwels, P. and Roxin, A. (2016). SimpleBIM: From full ifcOWL graphs to simplified building graphs Building Topology Ontology (BOT) View project SemanticGIS View project. In *ECPPM 2016 (11 European Conference on Product and Process Modelling)*.

Pauwels, P. and Terkaj, W. (2016). EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. *Automation in Construction*, 63:100–133.

Pauwels, P., Terkaj, W., Krijnen, T., and Beetz, J. (2015). Coping with lists in the ifcOWL ontology. *22nd EG-ICE International Workshop*, pages 113–122.

Pauwels, P., Zhang, S., and Lee, Y. C. (2017b). Semantic web technologies in AEC industry: A literature overview. *Automation in Construction*, 73:145–165.

Petrova, E. A., Romanska, I., Stamenov, M., Svidt, K., and Jensen, R. L. (2017). Development of an Information Delivery Manual for Early Stage BIM-based Energy Performance Assessment and Code Compliance as a Part of DGNB Pre-Certification. In *The 15th International Conference of International Building Performance Simulation Association*, volume 15, pages 2100–2109.

Port of Portland (2015). CAD & BIM Standards Manual 2015. *Port of Portland*, page 106.

Pratt, M. J. (2001). Introduction to ISO 10303—the STEP standard for product data exchange. *Journal of Computing and Information Science in Engineering*, 1(1):102–103.

Priya, D., Sivaraj, R., Assistant, R., and Gr, S. (2015). A Review of Missing Data Handling Methods. *International Journal On Engineering Technology and Sciences – IJETS™ ISSN*, 2(2):2349–3968.

Radulovic, F., Poveda-Villalón, M., Vila-Suero, D., Rodríguez-Doncel, V., García-Castro, R., and Gómez-Pérez, A. (2015). Guidelines for Linked Data generation and publication: An example in building energy consumption. *Automation in Construction*, 57:178–187.

Rasmussen, M., Pauwels, P., Lefrançois, M., Ferdinand, G., Hviid, C., Karlshøj, J., Rasmussen, M., Pauwels, P., Lefrançois, M., Schneider, G. F., and Hviid, C. (2017a). Recent changes in the Building Topology Ontology. In *5th Linked Data in Architecture and Construction Workshop*.

Rasmussen, M. H., Lefrançois, M., Bonduel, M., Hviid, C. A., and Karlshø, J. (2018). OPM: An ontology for describing properties that evolve over time. In *CEUR Workshop Proceedings*, volume 2159, pages 23–33.

Rasmussen, M. H., Pauwels, P., Hviid, C. A., and Karlshøj, J. (2017b). Proposing a Central AEC Ontology That Allows for Domain Specific Extensions. In *Joint Conference on Computing in Construction*, volume 1, pages 237–244.

Rasmussen, M. H., Pauwels, P., Lefrançois, M., and Schneider, G. F. (2019). Building Topology Ontology (BOT)-Draft Community Group Report 07 January 2019.

Reinisch, C., Kofler, M. J., Iglesias, F., and Kastner, W. (2011). Thinkhome energy efficiency in future smart homes. *Eurasip Journal on Embedded Systems*, 2011.

Ricquebourg, V., Durand, D., Menga, D., Marhic, B., Delahoche, L., Logé, C., and Jolly-Desodt, A. M. (2007). Context inferring in the smart home: An SWRL approach. *Proceedings - 21st International Conference on Advanced Information Networking and Applications Workshops/Symposia, AINAW'07*, 1(iv):290–295.

Ristoski, P. and Paulheim, H. (2016). RDF2Vec: RDF graph embeddings for data mining. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9981 LNCS:498–514.

Rossello-Busquet, A., Brewka, L. J., Soler, J., and Dittmann, L. (2011). OWL Ontologies and SWRL Rules Applied to Energy Management. In *2011 UkSim 13th International Conference on Computer Modelling and Simulation*, pages 446–450. IEEE.

Ruelens, F., Iacovella, S., Claessens, B. J., and Belmans, R. (2015). Learning agent for a heat-pump thermostat with a set-back strategy using model-free reinforcement learning. *Energies*, 8(8):8300–8318.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (2013). Learning Internal Representations by Error Propagation. *Readings in Cognitive Science: A Perspective from Psychology and Artificial Intelligence*, 5(3):399–421.

Rummery, G. A. and Niranjan, M. (1994). On-Line Q-Learning Using Connectionist Systems. *Technical Report CUED/F-INFENG/TR 166*.

Russomanno, D. J., Kothari, C. R., and Thomas, O. A. (2005). Building a Sensor Ontology: A Practical Approach Leveraging ISO and open geospatial consortium (OGC) models. In *The 2005 International Conference on Artificial Intelligence, Las Vegas, NV*, pages 637–643.

Scherer, R., Katranuschkov, P., Kadolsky, M., and Laine, T. (2012). Ontology-based building information model for integrated lifecycle energy management. In *eWork and eBusiness in Architecture, Engineering and Construction*, pages 951–956. CRC Press.

Schevers, H. and Drogemuller, R. (2005). Converting the Industry Foundation Classes to the Web Ontology Language. In *2005 First International Conference on Semantics, Knowledge and Grid*, pages 73–73. IEEE.

Schlichtkrull, M., Kipf, T. N., Bloem, P., van den Berg, R., Titov, I., and Welling, M. (2017). Modeling Relational Data with Graph Convolutional Networks. *European Semantic Web Conference*, 1:593–607.

Schneider, G. F. (2017). Towards Aligning Domain Ontologies with the Building Topology Ontology. In *5th LDAC workshop, 13-15 November*.

Shah, N., Chao, K. M., Zlamaniec, T., and Matei, A. (2011). Ontology for home energy management domain. *Communications in Computer and Information Science*, 167 CCIS(PART 2):337–347.

Sommaruga, L., Perri, A., and Furfari, F. (2005). DomoML-env: An ontology for human home interaction. In *CEUR Workshop Proceedings*, volume 166, pages 1–8.

Statsbygg (2013). Statsbygg Building Information Modelling Manual Version 1.2.1 (SBM1.2.1) –. *Norwegian Directorate of Public Construction and Property*, 1:1–98.

Sterne, J. A., White, I. R., Carlin, J. B., Spratt, M., Royston, P., Kenward, M. G., Wood, A. M., and Carpenter, J. R. (2009). Multiple imputation for missing data in epidemiological and clinical research: Potential and pitfalls. *BMJ (Online)*, 339(7713):157–160.

Strehl, A. L., Lihong, L., Wiewiora, E., Langford, J., and Littman, M. L. (2006). PAC model-free reinforcement learning. *ACM International Conference Proceeding Series*, 148:881–888.

Studer, R., Grimm, S., and Abecker, A. (2007). *Semantic web services: Concepts, technologies, and applications*.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Leaning: An Introduction*. MIT Press.

Tah, J. H. and Abanda, H. F. (2011). Sustainable building technology knowledge representation: Using Semantic Web techniques. *Advanced Engineering Informatics*, 25(3):547–558.

Terkaj, W. and Šojić, A. (2015). Ontology-based representation of IFC EXPRESS rules: An enhancement of the ifcOWL ontology. *Automation in Construction*, 57:188–201.

Tomic, S., Fensel, A., and Pellegrini, T. (2010). SESAME Demonstrator: Ontologies, Services and Policies for Energy Efficiency. *6th International Conference on Semantic Systems, I-SEMANTICS '10*, pages 1–4.

Urieli, D. and Stone, P. (2013). A learning agent for heat-pump thermostat control. *12th International Conference on Autonomous Agents and Multiagent Systems 2013, AAMAS 2013*, 2(May):1093–1100.

Venugopal, M., Eastman, C. M., and Teizer, J. (2015). An ontology-based analysis of the industry foundation class schema for building information model exchanges. *Advanced Engineering Informatics*, 29(4):940–957.

Villalón, M. P. and Castro, R. G. (2017). SAREF4BLDG-SAREF extension for building devices This version.

W3C (2006). W3C RDF Validation Service.

W3C (2013). Prefix.cc - Semantic Web Standards.

W3C (2014). Linked Building Data Community Group.

W3C-Linked Data Community Group (2018a). GEOMETRY: Ontology for 3D Geometry.

W3C-Linked Data Community Group (2018b). Product Ontology (PRODUCT).

W3C OWL Working Group (2012). OWL 2 Web Ontology Language Document Overview (Second Edition)-W3C Recommendation 11 December 2012.

W3C SPARQL Working Group (2013). SPARQL 1.1 Overview-W3C Recommendation 21 March 2013.

Wagner, A., Moeller, L. K., Leifgen, C., and Eller, C. (2019). Building Product Ontology.

Wei, T., Wang, Y., and Zhu, Q. (2017). Deep Reinforcement Learning for Building HVAC Control. In *Proceedings of the 54th Annual Design Automation Conference 2017, Association for Computing Machinery (ACM),*, page 22.

Weise, M., Nisbet, N., Liebich, T., and Benghi, C. (2016). IFC model checking based on mvdXML 1.1. In *11th European Conference on Product & Process Modeling, CPPM,At: Limassol, Cyprus.*, number September, pages 19–26.

Wetherill, M., Rezgui, Y., Lima, C., and Zarli, A. (2002). Knowledge management for the construction industry: The e-COGNOS project. *Electronic Journal of Information Technology in Construction*, 7:183–196.

Wetter, M. (2008). Building Controls Virtual Test Bed.

Wetter, M. (2011). Co-simulation of building energy and control systems with the Building Controls Virtual Test Bed. *Journal of Building Performance Simulation*, 4(3):185–203.

Wilcke, X., Bloem, P., and de Boer, V. (2017). The knowledge graph as the default data model for learning on heterogeneous knowledge. *Data Science*, 0:1–19.

Wix, J. and Karlshøj, J. (2010). Information Delivery Manual Guide to Components and Development Methods. *buildingSMART International*, 5(12):10.

Yang, L., Nagy, Z., Goffin, P., and Schlueter, A. (2015). Reinforcement learning for optimal control of low exergy buildings. *Applied Energy*, 156:577–586.

Yang, Q. and Zhang, Y. (2006). Semantic interoperability in building design: Methods and tools. *Computer-Aided Design*, 38(10):1099–1112.

Yu, Z. and Dexter, A. (2010). Online tuning of a supervisory fuzzy controller for low-energy building system using reinforcement learning. *Control Engineering Practice*, 18(5):532–539.

Zhang, C. (2019). *Requirement checking in the building industry : enabling modularized and extensible requirement checking systems based on semantic web technologies.* PhD thesis, Technische Universiteit Eindhoven.

Zhang, C., Beetz, J., and Weise, M. (2014). Model view checking: automated validation for IFC building models. *eWork and eBusiness in Architecture, Engineering and Construction*, 0:123–128.

Zhang, C., Beetz, J., and Weise, M. (2015a). Interoperable validation for IFC building models using open standards. Technical report.

Zhang, J., Seet, B.-C., and Lie, T. (2015b). Building Information Modelling for Smart Built Environments. *Buildings*, 5(1):100–115.

Zhang, Z., Chong, A., Zhang, C., Lu, S., Pan, Y., and Lam, K. P. (2018). A Deep Reinforcement Learning Approach to Using Whole Building Energy Model For HVAC Optimal Control. *2018 Building Performance Modeling Conference and SimBuild*, (July).

Zhang, Z. and Lam, K. P. (2018). Practical implementation and evaluation of deep reinforcement learning control for a radiant heating system. In *Proceedings of the 5th Conference on Systems for Built Environments*, number November, pages 148–157. ACM.

Zhao, W. and Liu, J. (2008). OWL/SWRL representation methodology for EXPRESS-driven product information model: Part I. Implementation methodology. *Computers in Industry*, 59(6):580–589.