University of Nottingham
UK | CHINA | MALAYSIA

A Framework for Knowledge Representation Learning-based Building Control

by

Kevin Luwemba Mugumya

Thesis Submitted to The University of Nottingham for the degree of
Doctor of Philosophy

Internal Examiner:
Dr. Tomas Maul

External Examiner:
Dr. Pieter Pauwels

Supervisor:
Dr. Wong Jing Ying

Co-Supervisor:
Prof. Andy Chan

May 24, 2024

Acknowledgements

I owe a big debt of gratitude to the numerous people who have helped me over the last few years with advice, encouragement, and direction as I worked on this thesis.

Before I get into its core, I want to thank Dr Jing Ying Wong for her guidance and helping me navigate the bureaucratic hurdles in academia and industry. Dr Wong's assistance and direction have been vital in providing structure and direction to my research. I owe a great deal of my understanding of Building Information Modelling and my research to Dr Wong's generosity of spirit, her patience, and the habitual way in which she puts things in perspective during our frequent and lengthy discussions on a wide range of issues. I consider myself extremely lucky to have received Dr Wong's counsel and to be able to count on her ongoing support.

I also want to extend my gratitude to Prof. Andy Chan, with whom I had several insightful and helpful conversations regarding my work. Prof. Andy's work ethic has been an invaluable example to my overall growth as a researcher. In addition, I had the pleasure of working with Dr Yip; his tranquillity and chilled vibes have shown me a different perspective on life. I am also really grateful to Dr Tomas Maul for agreeing to serve as my thesis's internal and second examiner.

To MES Group, you have been the cornerstone of my financial support and exposure to the industry dynamic. You have constantly challenged me to be better, do better, and strive for nothing but the best. For this, I will forever be grateful. So, this is to Ms Miranda and Ms Donna; thanks for believing in me and giving me a chance.

To my father Vincent, you not only introduced me to science and mathematics but also taught me to think creatively. I owe everything that I am to you. My siblings Susan, Maria, Joseph, Junior, Angela and the little one Cynthia - you know how grateful I am for everything you did for me throughout the years. To Uncle Dan, my mothers, Mary and Betty, there is so much that I would like to say, but I will just say thank you for always being there when I needed you.

Abstract

Current Building Automation Systems (BAS) have crucial context-awareness limitations that must be addressed before they can reach human-like levels and adapt better to the dynamic needs of modern buildings. Among others, our buildings still lack sensors, actuators and control agents that learn reliable models of the environment and plan complex action sequences. Moreover, modern Machine Learning (ML)-backed BAS, though trained on massive datasets, are usually overly specialized (trained for one task) and brittle (make stupid mistakes). In contrast, human learning is very efficient, and with only a few examples, we can find intuitive ways to complete a task while generalizing our learning to other tasks. To address the above limitations, this thesis proposes a foundational framework that advances the context-awareness capabilities of BAS using knowledge graphs and Knowledge Representation Learning (KRL). At the framework's core is the notion of using Semantic Web Technologies (SWT) to model the semantic relationships between different building components, packaging them inside a network-like data structure called a knowledge graph, and using KRL to learn the hidden patterns within the graph. During the learning phase, KRL utilizes message-passing to propagate the learnt information throughout all nodes in the graph. This research hypothesizes that building automation agents can leverage the notion of message-passing to aggregate information from all entities in a graph and use it to continuously update their understanding of a building's systems and components. The perception is that imbuing agents with holistic information about the buildings they control can support context-aware and intelligent decision-making during building automation.

To test the research hypothesis, a three-phase investigation was carried out: literature review, framework development, and evaluation. Phase one focused on *situating the research* within the scholarly discourse of Building Information Modeling (BIM), BIM-based knowledge graphs, building automation, and KRL. A bibliometric search was conducted using Google Scholar, Web of Science, and Scopus to find relevant academic publications and after applying specific selection criteria, 110 publications were identified, labelled, and clustered accordingly. The results show that since the advent of SWT in the Architecture, Engineering, Construction and Facility Management (AEC/FM) field in 2010, it has been a driving force in advancing BIM research by providing the mechanics to represent complex relationships within the built environment. Concurrently, KRL has seen significant development in domains such as bioinformatics, where it has been used to understand complex biological relationships and processes; however, despite the apparent suitability of applying KRL to the BIM field, such integration has not materialized and remains largely unexplored. To get around these research shortcomings, the next phase of this thesis was to *develop a framework* for applying KRL to BIM-based knowledge graphs using a two-step process: specification and performance analysis. The specification formalised the structural and relational patterns the KRL system had to learn from BIM-based knowledge graphs. These patterns are fundamental for understanding the connectivity and interactions between a building's entities and can be leveraged for various building automation tasks. For *performance analysis*, 3 families of KRL models (tensor decomposition, geometric, and deep learning) were chosen and experimentally tested on their ability to learn the specified patterns. Two publicly available BIM-based knowledge graph datasets were used in these experiments. The goal was not to identify the best KRL model configurations. Rather this research took a deeper look at how model performance can be affected by changes in the training step, choice of hyper-parameter optimization strategy, model parameter initialization and dataset split mechanics. From this, a framework was developed encapsulating step-by-step recommendations of using KRL with BIM-based knowledge graphs for not only building automation, but also other usecases in the AEC/FM domain. To assess the viability of the framework, an evaluation pipeline was set up consisting of a BIM model, Internet of Things (IoT) devices, and a prototype program of the framework wrapped inside an Application Programming Interface (API). The API consists of a server-side module and a client-side module. The server-side module allows a building automation system to communicate with external services such as knowledge graph databases, sensor data stores, and Message Queuing Telemetry Transport (MQTT) brokers. The client-side module consists of a Construction Operations Building Information Exchange (COBie) handler service which facilitates the curation of BIM-based knowledge graphs from COBie files in a Graphical User Interface (GUI) and an interrogation service that facilitates declarative interrogation of the server-side module.

The results indicate that RotatE consistently outperforms other models across both datasets, establishing itself as a robust baseline within the context of building automation. Notably, older models like TransE can still be competitive with optimized training and Hyper-parameter Optimization (HPO) configurations. NSSA and AdaGrad emerged as favorable training setup choices, suggesting their potential as initial benchmarks for future evaluations. This study further underscores the significance of HPO, revealing its substantial impact on model performance. Despite extensive hyper-parameter searches, there remains considerable variance among top-performing model configurations, indicating the need for nuanced parameter combinations. This complexity suggests that manual tuning may not yield optimal results, advocating for the adoption of HPO strategies. Furthermore, the disparity in hyper-parameters between the two knowledge graph datasets underscores the influence of dataset-specific parameters. Random search methods, when repeated sufficiently, yielded configurations closely comparable to more systematic approaches, albeit in less time.

This work emphasizes the critical importance of comprehensively reporting model architectures, training setups, hyperparameters, and dataset splits to enhance reproducibility in research. The insight highlights a prevalent issue in the AEC/FM field where results are often difficult to replicate due to incomplete documentation of experimental setups.

In conclusion, knowlegde graphs play an increasingly vital role in addressing the context-awareness limitations of BAS. This thesis assessed how various factors influence the performance of 3 classes of KRL models on 2 real-world BIM-oriented knowledge graphs. However, for these KRL approaches to impact building automation, they must be integrated into decision-making processes within the industry. This integration relies on establishing trust and enhancing understanding of KRL-based methods among AEC/FM stakeholders. Increased focus on foundational aspects such as transparency, reproducibility, and understanding of factors influencing different tasks and contexts is essential to unlock the potential to improve building automation efforts.

List of Figures

List of Tables

List of Publications

List of publications here

List of Abbreviations

AEC	Architecture, Engineering and Construction
AEC/FM	Architecture, Engineering, Construction and Facility Management
API	Application Programming Interface
AUC	Area under the Receive Operator Curve (ROC) Curve
BAS	Building Automation Systems
BCEL	Binary Cross-Entropy Loss
BIM	Building Information Modeling
CAD	Computer-Aided Design
CAFM	Computer-Aided Facility Management
CMSS	Computerized Maintenance Management System
CNN	Convolutional Neural Network
COBie	Construction Operations Building Information Exchange
CWA	Closed World Assumption
FM	Facility Management
GAT	Graph Attention Network
GCN	Graph Convolutional Network
GNN	Graph Neural Network
GraphSAGE	Graph Sampling and Aggregation
GUI	Graphical User Interface
HVAC	Heating, Ventilation and Air Conditioning
HPO	Hyper-parameter Optimization
IFC	Industry Foundation Classes
IoT	Internet of Things
KRL	Knowledge Representation Learning
LBD	Linked Building Data
LSTM	Long short-term memory

ML	Machine Learning
MR	Mean Rank
MRL	Margin Ranking Loss
MRR	Mean Reciprocal Rank
MQTT	Message Queuing Telemetry Transport
PDF	Portable Document Format
POC	Proof of Concept
R-GCN	Relational Graph Convolutional Network
RDF	Resource Description Framework
RNN	Recurrent Neural Network
ROC	Receive Operator Curve
SGD	Stochastic Gradient Descent
SPL	Softplus Loss
SHACL	Shape Constraints Language
SPARQL	SPARQL Protocol and RDF Query Language
SRL	Statistical Relational Learning
SWT	Semantic Web Technologies
TransE	Translating Embeddings

Contents

Chapter 1

Introduction

The Facility Management (FM) life-cycle of buildings is characterized by a continuous flow and exchange of information. The involved parties are predominantly operational building systems, sensor networks, actuators, control agents[1], and building occupants. At the foundation of each party exists heterogeneous processes that inhibit the seamless flow of *contextually rich information* needed for several downstream FM tasks, of which building automation is the focal point of the investigation herein.

This introductory chapter starts by framing the research context within the boundaries of ongoing efforts to address the issue encapsulated in the above statement. A proposal of the core research problem is then made, followed by its breakdown into more specific research questions. The aim and objectives of the study are then made explicit, and the scope of work is laid out. Finally, this chapter concludes with a summarized methodology, research outcomes, and the organizational structure for the rest of the thesis.

1.1 Research Context and Motivation

With most people spending 80–90% of their daily lives indoors, buildings have become the largest consumers of global energy due to heavy reliance on heating and air conditioning (ASHRAE, 2016). Undoubtedly, the building industry has continued to put pressure on the sustainability equilibrium of the natural environment. Notably, extremely high temperatures and prolonged heat waves have been recorded in many continents and countries (Somerville et al., 2012; Akompab et al., 2013; Hopke, 2020; Miller et al., 2021; Junk et al., 2019). Moreover, the frequency, intensity, and duration of heat waves are increasing rapidly, making adaptation to heat a priority (Peng et al., 2011; Mitchell et al., 2016; Baniassadi et al., 2018; Alam et al., 2019; Kriebel-Gasparro, 2022).

Global energy efficiency policies and regulations are quickly evolving to reverse this trend (Zhou et al., 2020; Viguié et al., 2020) and the ripple effects are being felt by building owners. They are increasingly being forced to develop buildings characterized by intricate

[1] In this thesis, the term *agent* is used to mean anything that can perceive the built environment around it, take control actions autonomously in order to achieve a specific set of goals, and may iteratively improve its performance by learning from the information around it.

automation systems and swarms of sensor networks towards optimal performance. With this ever-growing complexity of the built environment, so has the increase in the *maintenance challenge*. Moreover, the already existing stochastic factors in play, such as occupancy behavior, envelope tightness and variable weather patterns, only compound this problem. As a result, developing agents with contextually adaptive control policies has become a finicky process requiring exhaustive thought and care. Curry et al. (2012)'s investigation attributed this puzzle to difficulties in identifying and exploiting the inherent latent dependencies between the factors mentioned above.

1.1.1 The Facility Management Challenge

As soon as a building is *commissioned*, a chain of events is set into motion to ensure proper functionality of its systems and that operational efficiency targets are met in compliance with set regulations. Over the years, this FM process has steered towards *occupant-centricity*, which not only means that building occupants are getting more engaged in the operation process of the embedded building systems but also, optimization targets are not achieved at the expense of their comfort (Park and Nagy, 2018; Park et al., 2019b,a; O'Brien et al., 2020; Park et al., 2022). On that basis, FM qualifies to be a *multi-objective optimization problem* that requires a careful trade-off analysis between conflicting objectives (i.e., achieving both operational and energy efficiency while maintaining acceptable indoor air quality and thermal comfort) (Toffolo and Lazzaretto, 2002; Delgarm et al., 2016; Shaikh et al., 2018; Yong et al., 2020).

Just like any other stage of a building's life-cycle, FM is a heavily data-driven process that involves multi-disciplinary stakeholders constantly exchanging and sharing *heterogeneous* information, which is mainly attributed to their departmentalized data handling cultures. Any deficiencies that arise in managing this heterogeneity can arguably propagate to the building systems in the loop, leading to unintended and unexpected under-performing behavior.

1.1.2 Digitization of the Facility Management Process

Traditionally, FM information is collated by the design and construction team and piped to the operations team close to the handover stage of a building. At such a time when project budgets and deadlines are soon approaching their elastic limit, perhaps an important question to ask is *"how often is this information checked for completeness, accuracy or reliability?"*. The answer to this question is arguably *never*. To complicate matters further, some FM information is stored using traditional Computer-Aided Design (CAD) drawings and paper files, making its utilization cumbersome and inefficient. As a result, building owners started to embrace Computerized Maintenance Management Systems (CMSSs) and Computer-Aided Facility Management (CAFM) systems to capture FM information in a more structured and digitized way. However, even with these, typical day-to-day operational information is usually locked down in a myriad of Portable Document Format (PDF) files. All these challenges necessitate an efficient mechanism for capturing and propagating FM information from the outset of a building's design and construction to its operation.

To an extent, BIM has served in this role as the primary driver of digitization in the AEC/FM industry by providing an efficient way of handling large amounts of building information (*semantic* and *geometric*) centrally within a three-dimensional model (Borrmann et al., 2018). However, several obstructions still lie on the critical path of sharing this model information *within* and *outside* the AEC industry making it hard for other domains to become part of the BIM story (Jeroen et al., 2018; Pauwels et al., 2017b). Literature has attributed this exchange 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 2016, the IFC schema was only available in its native EXPRESS format, which is cumbersome to work with in domain applications such as building automation, geo-spatial, heritage and facility management (Pauwels and Terkaj, 2016; Pauwels and Roxin, 2016).

Specific to FM is the COBie standard, a subset of IFC which encapsulates the industry's best practices for exchanging FM information between a construction firm and a facility management team (William East et al., 2013; Teicholz, 2018). Though COBie 's adoption and interest are on the rise, its spreadsheet architecture is cumbersome to navigate (Anderson et al., 2012; Kumar and Teo, 2021a,b) and there are still many misconceptions surrounding its use, and as a result, it is under-utilized.

Meanwhile, independent of IFC and outside the Architecture, Engineering and Construction (AEC) industry, other powerful knowledge-representation techniques are trending with various disciplines able to interlink their heterogeneous datasets using SWT underpinned by principles of the world wide web (Berners-Lee et al., 2001b; Berners-Lee, 2003, 2006). Only recently, has there been increased research interest in applying this notion to the BIM ecosystem as a mechanism of integrating, managing and extracting value from its heterogeneous data sources (Barbau et al., 2012; Beetz et al., 2009; Pauwels and Roxin, 2016; Pauwels and Terkaj, 2016).

1.1.3 Building Automation in Facility Management

FM is becoming increasingly reliant on digitized workflows such as CMSSs and CAFM systems to enhance its efficiency and *building automation*. Building automation is ideally a centralized process that involves the automated control of a building's electrical equipment such as Heating, Ventilation and Air Conditioning (HVAC), lighting and access control, all driven by sensor networks, actuators and control agents, which follow a set of pre-defined or self-learnt control policies.

As mentioned earlier, FM is a multi-objective optimization problem, and *ML* is a promising solution that is being widely adopted to solve such problems (Toffolo and Lazzaretto, 2002; Asadi et al., 2012; Shaikh et al., 2018; Chen et al., 2018a; Merlet et al., 2022; Wijeratne et al., 2022). At the foundation of ML is the principle of first developing a statistically-driven mathematical model, a mechanism for ingesting data in its most raw form while subsequently learning to extract the most relevant information (typically *hidden features* and *patterns*) necessary for performing a specific downstream task. Although there are several advancements that are making it possible for ML models to extract value from a concoction of disparate information sources, many ML methods are *domain-specific* and tailored to ingest data of the same format. For example, Recurrent Neural Network (RNN) and Long short-term memory (LSTM) models (Hochreiter and Schmidhuber, 1997) are designed to handle sequence prediction problems—machine translation and speech recognition—involving sound and text data while Convolutional Neural Network (CNN) models (LeCun et al., 1998) specialize in learning from image and video data.

On the other hand, the building automation domain is complex, highly heterogeneous and fragmented. It exhibits data with multiple modalities, each with different statistical properties and levels of specificity. Therefore, a naive application of typical ML workflows in this domain would lead to models that apply deductions with low precision, efficiency and scalability. In pursuit of an integrator for heterogeneous FM domain information, several proposals anchored by SWT have been put forward in the literature (Pauwels et al., 2018; Pauwels and Terkaj, 2016; Pauwels et al., 2017b; Rasmussen et al., 2019a; Pauwels et al., 2022). The resulting semantic glue has made it easier for facility managers to link and holistically analyze data collected across multiple operational building systems. This work also envisages such an integrator to be a data fusion strategy that can be embedded in the learning pipeline of building automation ML models towards improved *collective reasoning*. However, due to the limited understanding of the peculiarities arising from linking FM data, developing ML strategies for downstream building automation tasks is still in its infancy and proving to be a major hurdle. Certain application fields such as social network analysis, drug discovery in bioinformatics, and fraud detection in e-commerce often deal with immensely interwoven and complex dataset structures. Knowledge Representation Learning (KRL), Relational Learning (RL) and Statistical Relational Learning (SRL) are areas of machine learning that have made significant strides in understanding the idiosyncrasies of these datasets (Nickel et al., 2011, 2012; Bengio et al., 2013; Nickel et al., 2016a; Lin et al., 2018; Yi et al., 2022). However, the same cannot be said for their application in the FM domain yet it exhibits similarly intricate datasets and this is the research direction this thesis is taking.

Before crafting the problem statement, it is necessary to delineate the distinction between KRL, RL and SRL. KRL, RL and SRL are related but distinct fields in Machine Learning and Artificial Intelligence. Let's start with what they all have in common, which is a mechanism for extracting knowledge from data and representing it in a structured format for downstream tasks. KRL involves learning a set of predefined concepts and relationships from data and using them to represent knowledge. This approach typically involves manually designing a set of concepts and relationships and then using ML techniques to extract instances of these concepts and relationships from data (Liu et al., 2016). RL, on the other hand, involves learning the relationships between entities in a dataset without predefined concepts or relationships. This approach typically involves using ML techniques to discover patterns in the data and infer relationships between entities based on these patterns (Singh and Gordon, 2008). Contrarily, SRL involves using probabilistic models to capture the uncertainty and dependencies between entities and their relationships in a dataset. This approach allows the model to make probabilistic predictions about the relationships between entities and to reason about the uncertainty of these predictions Ginestet (2010).

KRL and BIM-based knowledge graphs are closely related concepts. Knowledge graphs make it possible to formalize BIM data in a graph format, where nodes represent building components, systems, and other entities, and edges represent relationships between them. In the context of BIM, KRL can be used to analyze and understand the relationships between different building components, systems and other entities represented in the BIM knowledge graph. This can enable a wide range of applications, such as energy efficiency analysis, building performance prediction, and maintenance planning. One of the advantages of using BIM-based knowledge graphs and relational learning is that it allows to take into account the complex interactions and relationships between building components, systems and other entities, which can be difficult to capture using traditional ML methods. Finally, BIM knowledge graphs can also be used to enable more human-like reasoning and decision-making, by providing a way to represent and reason about the knowledge of building experts, which can be used to support the decision-making process in the design, construction, and operation of buildings.

The current state-of-the-art in merging relational learning with BIM-based knowledge graphs is still in its early stages, but there have been a growing number of research efforts in this area in recent years. One of the main challenges in merging relational learning with BIM-based knowledge graphs is the integration of data from different sources. BIM models typically include a large amount of structured data, such as building components and systems, while sensor data and energy consumption data are typically unstructured. Integrating these different types of data requires the use of data integration and preprocessing techniques, such as ontology alignment (Schneider, 2017). Another challenge is the ability to handle incomplete or missing data, which is common in BIM models and sensor data. This requires the use of techniques such as data imputation or dimensionality reduction to fill in missing values or identify the most important features of the data. As building data is often temporal and can change over time, there is a need for techniques anchored by time-series analysis to identify patterns and trends in the data over time. Finally, there is a need for the development of new evaluation metrics to evaluate the performance of relational learning models on BIM-based knowledge graphs, as the traditional ML metrics may not be suitable for this type of data. Overall, there is a growing interest in the research community in merging relational learning with BIM-based knowledge graphs, and new developments are expected in the near future.

1.2 Problem statement

How can FM datasets originating from various sources in and outside of a building be efficiently integrated into the self-learning process of building automation agents?

FM datasets are inherently heterogeneous and fragmented. If *expressive* enough mechanisms are orchestrated to *represent* and *unify* these datasets, the resulting analytics have the potential to confirm known FM inefficiencies, shed light on new ones, or prove hypotheses wrong.

Whilst Semantic Web Technologies have emerged as the promising orchestrator to achieve this, thus far, their primary focus has been on achieving semantic interoperability for logical inference and complex querying. However, what is still in its infancy is investigating how to leverage the inherent relational structure of semantically inter-linked FM datasets as a mechanism for message passing and information propagation to facilitate *collective contextual reasoning*[2] in building automation agents. In an attempt to bridge this gap, this thesis builds upon the work of multiple earlier researchers to propose a *Knowledge Representation Learning-based Building Control framework* (KRL-based BC). A Proof of Concept (POC) workflow is presented to show how the proposed framework can be deployed in practice and from this, a set of requirements necessary to evaluate the framework is also delineated. A design of the following research questions is deemed appropriate to guide the process of discovering a solution to the above problem.

1.3 Research Questions

- **Research Question 1**: *How can knowledge graphs be used to represent the semantic relationships between different building components and systems?*

  This research question addresses an important data management problem in the highly fragmented and data-intensive building automation domain. The question is tackled by first analyzing the current literature for relevant theories, methods, and tools that have been developed to capture semantic relationships between different building components and systems with regard to automation and control. Specific focus is placed on the use of ontologies and SWT to formulate BIM-based knowledge graphs while investigating their fit within the boundaries KRL-based building control.

- **Research Question 2**: *How can KRL be used to learn the relationships between different building components and systems?*

  This research question investigates effective ways to integrate and use linked building data in the training and evaluation of KRL algorithms, and how the reliability and robustness of these algorithms can be ensured. A literature review is first conducted to analyze current theories, hypotheses, and tools that have been developed to automate the control of building systems based on facts represented in a knowledge graph. While making sure to address the discovered literature gaps, the 4-step framework is formulated to guide the rest of this question's investigation.

  1. Data preprocessing: This research investigates different LBD pre-processing techniques within the KRL context. This investigation entails an exploration of the

[2] Inter-linked data exhibits patterns and dependencies that occur between attributes and relationships of different entities of the dataset. ML methods that can exploit these patterns *collectively* in their learning pipeline are refered to in this thesis as exhibitants of *collective contextual reasoning*.

mechanics of these pre-processing techniques for generating a robust and good quality training LBD set that is free of irrelevant and redundant information or noisy and unreliable data for relational learning models.

2. Data splitting: It is a common practice to split the data into training, validation, and test sets. The training set is used to train the model, the validation set is used to tune the model hyperparameters, and the test set is used to evaluate the performance of the model.
3. Feature engineering: LBD may contain a large number of features, and it is important to select the most relevant and informative ones for training the model. This is done using feature selection techniques, such as selecting the most important features based on their correlation with the target variable or using dimensionality reduction techniques to reduce the number of features.
4. Model selection: There are various relational learning algorithms available, and it is important to select the one that is most suitable for the task at hand. This may involve comparing the performance of different algorithms on the training and validation sets, and selecting the one that performs the best.

- **Research Question 3**: *What are the benefits and limitations of merging BIM-based knowledge graphs and relational learning within the context of building automation and reasoning?*

  To answer this, we first perform a thorough literature analysis to learn more about the present barriers to combining (BIM)-based knowledge graphs with KRL for the purpose of automation and reasoning in buildings. We then conduct an experimental investigation to assess how the combination in question works. This experimentation entails the testing of different KRL techniques on (BIM)-based knowledge graphs. The performance of the system is then evaluated using a number of measures and compared to more conventional methods of building automation and reasoning. To evaluate the applicability and scalability of the proposed method, as well as to identify any limitations or issues that may arise when applying the method in real-world scenarios, a case study is conducted by applying the method to a campus building. Finally, the interpretation of results is done, and conclusions are drawn about the benefits and limitations of merging BIM-based knowledge graphs and KRL for building automation and reasoning while delineating any recommendations for future research in this area.

1.4 Aim

To propose and evaluate a KRL-based Building Control Framework that leverages the inherent relational structure of semantically inter-linked FM and building automation datasets to facilitate collective contextual reasoning in building automation agents.

1.5 Objectives

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

1. To explore the use of knowledge graphs to represent the semantic relationships between different building components and systems.
2. To investigate the use of KRL to learn the relationships between different building components and systems.
3. To evaluate the benefits and limitations of merging BIM-based knowledge graphs and KRL within the context of building automation and reasoning.

1.6 Research scope

This scope delineated below is not only deemed appropriate to provide a clear research direction but is also broad enough to allow for a range of possible approaches and methods to be explored by this thesis.

1.6.1 BIM-based knowledge graphs

1. Research on the use of knowledge graphs to represent the semantic relationships between different building components and systems involves an analysis of existing knowledge graph models and their suitability for representing building data.
2. A study is conducted on the development and implementation of BIM-based knowledge graphs, including the selection of suitable knowledge representation languages and data modelling techniques.
3. A performance evaluation of the knowledge graphs is done in terms of their ability to represent and query building data while comparing them with traditional data representation methods.

1.6.2 Knowledge Representation Learning on BIM-based knowledge graphs

1. Research on the use of KRL to learn the relationships between different building components and systems involves a study of existing KRL models and their suitability for learning from BIM-based knowledge graphs.
2. Experiments are conducted on the application of KRL techniques to the data represented in the BIM-based knowledge graph, including the selection of suitable algorithms and evaluation metrics.

3. Performance analysis of KRL models is done in terms of their ability to identify patterns and relationships within the building data and compare them with traditional ML methods.

1.6.3 Evaluation of benefits and limitations of integrating BIM-based knowledge graphs and KRL into building automation and reasoning systems

1. This involves the design and implementation of a framework that leverages the knowledge graphs and KRL models from the previous steps.

2. Experiments are conducted to evaluate the performance of the framework in terms of its ability to support building automation and reasoning tasks, such as energy management, and fault detection, and compare it with traditional building automation systems.

3. Analysis of the scalability and feasibility of the proposed approach is carried out using a campus building.

4. Analysis of the experimental results, case study, and literature review is done to draw conclusions about the benefits and limitations of the proposed approach and identify potential areas for future research.

1.7 Summarized methodology

The research involves three phases: literature review, system development, and evaluation. In the literature review phase, a comprehensive review is conducted to identify the current state of research in the areas of BIM-based knowledge graphs, KRL, and building automation and reasoning. In the system development phase, a BIM-based knowledge graph is developed to represent the semantic relationships between different building components and systems using ontologies and semantic web technologies. Additionally, a KRL algorithm is used to learn the relationships between different building components and systems. In the evaluation phase, the BIM-based knowledge graph and KRL algorithm are integrated into a building automation and reasoning system and its performance is evaluated using a case study of a real building. The performance is measured using metrics such as energy efficiency, comfort, and reliability.

1.8 Limitations

The proposed approach for building automation and reasoning based on BIM-based knowledge graphs and KRL is a novel approach, but it also has several limitations that need to be considered.

1.8.1 Scientific

- The proposed approach has been developed and evaluated for a specific type of building (offices in a University Campus setting). Therefore, it may not generalize well to other types of buildings or building systems, such as older buildings or buildings with different types of systems.
- The proposed approach may not be able to handle time-series building data effectively as building automation and control systems often generate time-series data that need to be re-analyzed over time for the algorithms in question to stay relevant.
- Building data is often incomplete or uncertain, which can make it difficult to represent and analyze. The proposed approach has some mitigations for this however these may not be robust enough to handle all cases of missing or uncertain data effectively.
- Building data often contains several layers of intertwined sensitive information that need to be protected however, the approach presented herein may not cater fully to these privacy issues.

1.8.2 Non-scientific

- The proposed approach requires real-world building data for experimentation and evaluation. However, access to real-world building data was difficult and time-consuming, which may have impeded the robustness of the presented approach.
- The proposed approach requires extensive expertise in building automation and control systems. However, expertise in this field was limited.
- Limited collaboration and participation from industry partners and building operators.

1.9 Contributions of this Thesis

This research is significant in terms of its theoretical and practical applications in extending the boundary of knowledge within the building automation domain.

1.9.1 Practical contributions

The proposed approach has several practical contributions to society and the building industry. The approach provides building operators and facility managers with a powerful tool for optimally automating building control and energy management tasks which can lead to improvements in energy efficiency, comfort, safety, and overall sustainability of buildings. Additionally, this research also avails facility managers 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 building control systems.

1.9.2 Theoretical contributions

In addition to the above, the proposed approach also makes several theoretical contributions to the research community. The findings of this study will provide a better understanding of how KRL can be used to leverage the relational structure of building data for building automation and control. This can be applied to other domains and used as a new direction for future research.

1.9.3 Expected deliverables

Based on the research scope and objectives outlined above, the expected deliverables of this research are:

1. A BIM-based knowledge graph model for representing building automation and control data, including the utilized data modelling techniques and knowledge representation languages.

2. A set of KRL models for learning the relationships between different building components and systems and their evaluation metrics.

3. A prototype building automation and reasoning system that integrates the BIM-based knowledge graph with one of the KRL models, including a detailed description of the system architecture and implementation details.

4. A set of evaluation metrics and a proof-of-concept (POC) workflow for evaluating the performance of the proposed approach in terms of building automation and reasoning tasks such as energy management and fault detection.

5. A case study that demonstrates the practicality and scalability of the proposed approach in a real-world building project.

6. A set of source code, scripts, and framework templates for implementing the proposed approach.

7. Any additional deliverables such as datasets, models and visualizations that were used for experimentation and evaluation.

1.10 Summary

This introduction chapter has detailed the study purpose, scope, and objectives of this thesis, which are to propose and assess a novel approach for building automation and reasoning based on BIM-based knowledge graphs and relational learning. The research will be divided into four main components: the use of knowledge graphs to represent building data, the use of relational learning to learn the relationships between building components and systems, the integration of BIM-based knowledge graphs and relational learning into a building automation and reasoning system, and the evaluation of the benefits and limitations of this approach within the context of building automation and reasoning. This research has the potential to bring several benefits to the building industry and society, such as improving energy efficiency, comfort and safety of building occupants, and overall sustainability of buildings. The study is conducted through a detailed literature review, experimental study, case study and implementation of a proof of concept (POC) workflow.

The next chapter will provide a detailed background on the research topic by reviewing the relevant literature on building automation and control systems, knowledge representation and learning, and BIM-based knowledge graphs. The literature review will help to identify the gaps in current research and the potential areas for future research, providing a solid foundation for the rest of the thesis.

Chapter 2

Literature Review

2.1 Introduction

The literature review chapter provides a detailed background on the research topic and identifies the gaps in current research and the potential areas for future research. The literature review focuses on building automation and control systems, knowledge representation and learning, and BIM-based knowledge graphs.

2.1.1 Brief Background

Building automation and control systems have been widely studied in the literature, however, the integration of semantic information with building automation and control systems to facilitate collective contextual reasoning remains an open research question. This chapter demonstrates that current approaches for building automation and control systems are inadequate for handling the complexity and heterogeneity of building data and that a more comprehensive approach is needed. Furthermore, this chapter argues that BIM-based knowledge graphs and KRL can provide a powerful framework for representing, analyzing, and integrating building data with building automation and control systems, and can facilitate collective contextual reasoning in building automation agents. Before proceeding, it is necessary to highlight what this research means by the terms *collective contextual reasoning* and *building automation agents with collective contextual reasoning* because these two concepts are at the epicenter of this chapter's argument.

2.1.2 What is Collective Contextual Reasoning?

Collective contextual reasoning in this thesis refers to the ability of building automation agents to reason about and make decisions based on the collective context of a building they are operating in. This includes not only the current state of the building but also the historical data, the current goals and objectives of the building, the current weather conditions, and the behavior of the building's inhabitants. These various pieces of information are used to understand the overall context of the building and to make decisions that will optimize the building's performance, energy efficiency, comfort, and safety for the occupants.

2.1.3 Building Automation Agents With Collective Contextual Reasoning

Building automation agents are software or hardware systems that are designed to automatically control and monitor the various systems and components within a building. These agents rely on sensors and actuators to gather data about the building and to control the various systems, such as lighting, security, heating, ventilation, and air conditioning (HVAC). Collective contextual reasoning allows agents to use this data holistically and make informed decisions that optimize the building's performance, rather than simply responding to pre-programmed rules or setpoints. Building control systems are dependent on a number of complex heterogeneous parameters ranging from indoor to outdoor. These, inherently have unknown latent dependencies which can be statistical in nature rather than deterministic. The previous chapter hypothesized that a holistic representation and interpretation of building information makes such latent dependencies accessible in the self-learning process of building automation agents toward optimality.

2.2 The Need for Linked Data in Building Automation and Control

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).

2.2.1 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., 2018c; 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

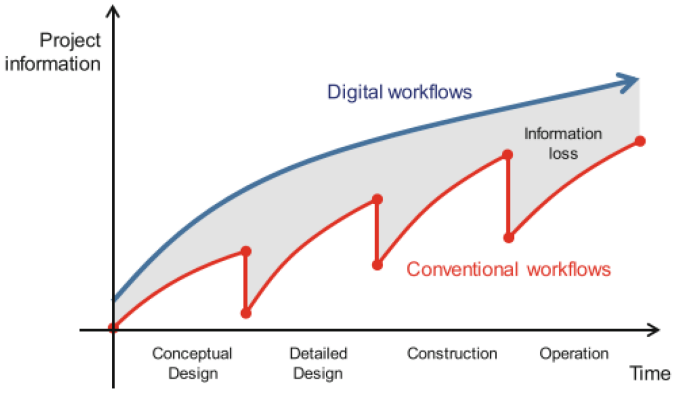


Figure 2.1: Information loss at various stages of the project lifecycle (Borrmann et al., 2018)

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 2.2.2 below.

2.2.2 Industry Foundation Classes

As delineated in the subsection 2.2.1, 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

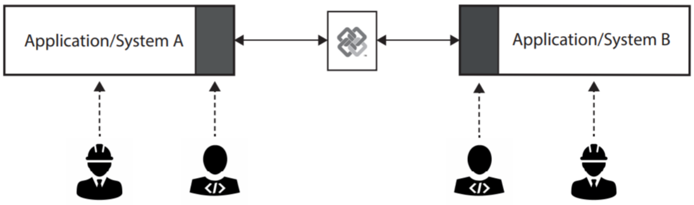


Figure 2.2: IFC exchange (which relies on end-users modelling expertise) between two BIM software via import-export routines implemented by software developers (Zhang, 2019).

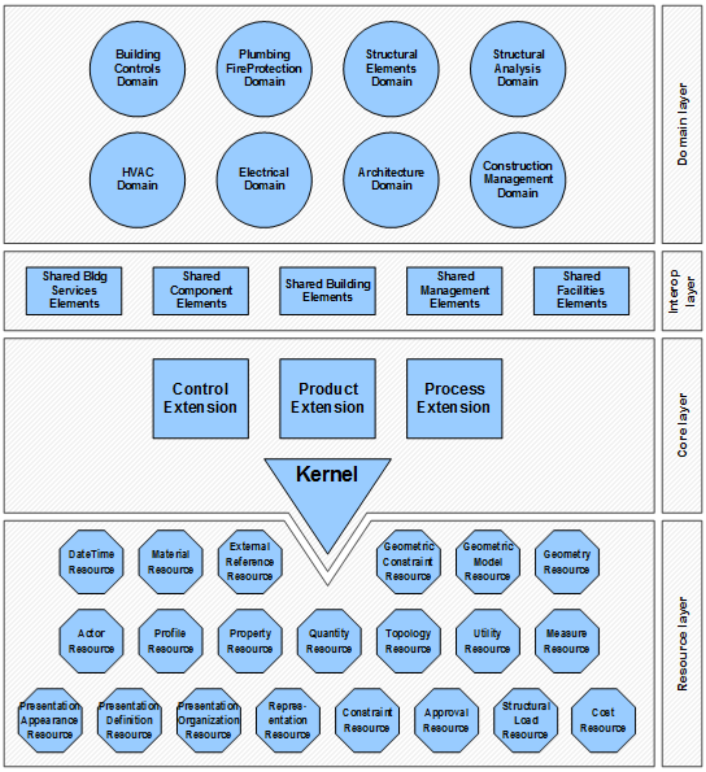


Figure 2.3: Conceptual layers of the IFC data model. Source: buildingSMART (Liebich, 2013).

model need to be implemented 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 2.2.3, but first, a brief introduction to EXPRESS, the data modelling language for the IFC standard.

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. 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).

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.

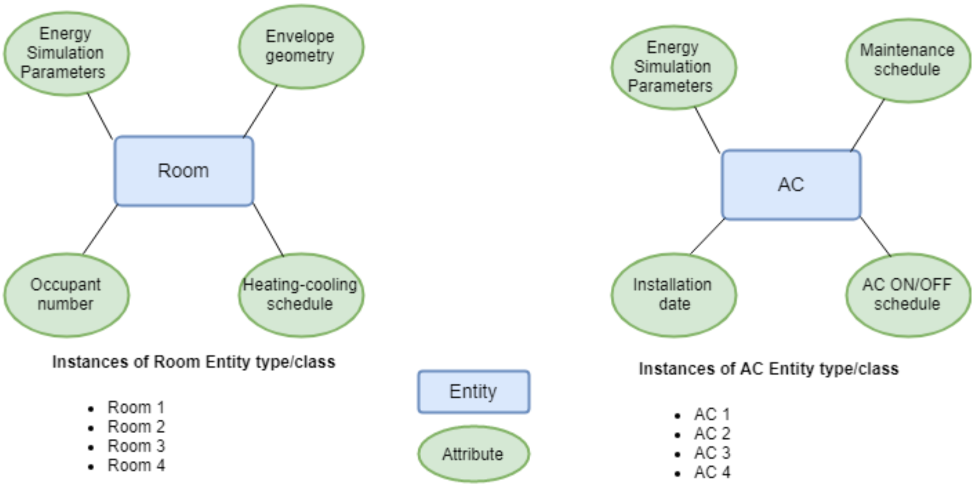
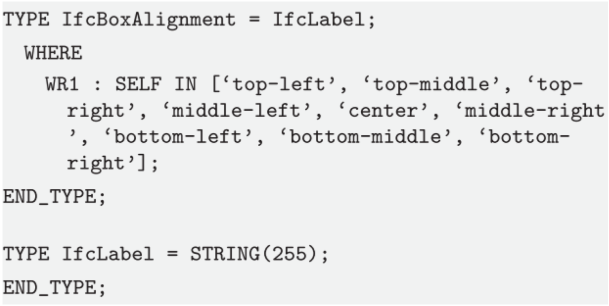


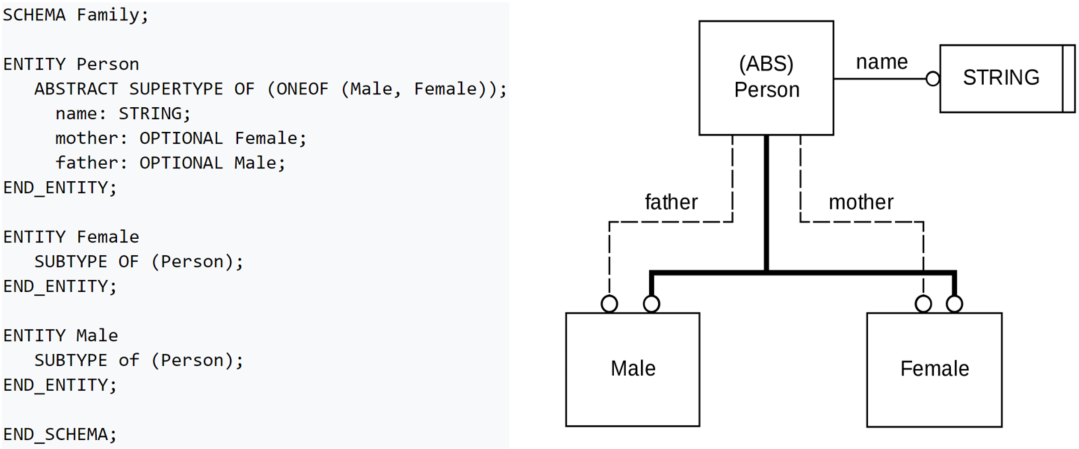
Figure 2.4: Entity relationship diagram showing both the Room (left) and AC (right) Entity types with their respective attributes and instances (below). Instances of the room entity type are indirectly associated with those of AC type via the shared property *Energy Simulation Parameters.*



```
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).

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).

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 2.3) 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.

2.2.3 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).

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 (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 2.2.3.

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).

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 (?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.

Therefore more flexible and extensible ways of representing shared knowledge in the

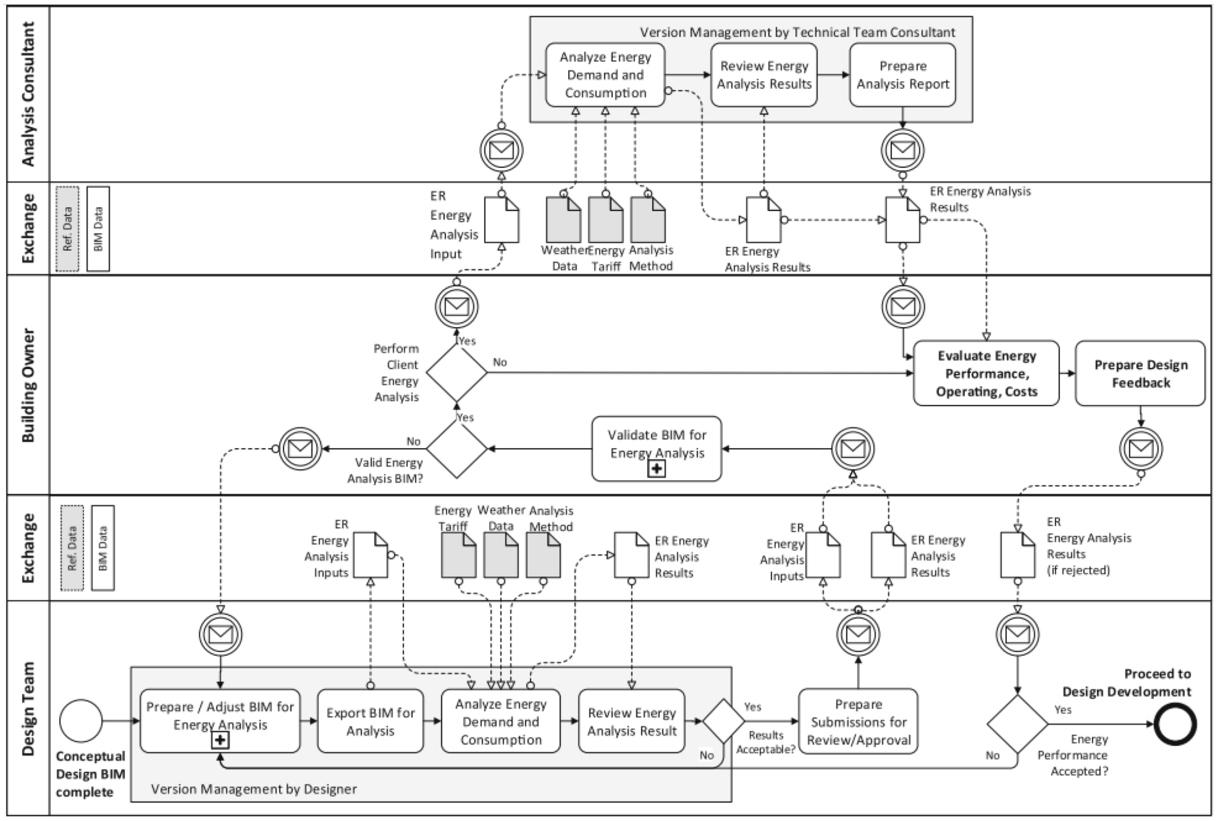

Analysis Consultant
Exchange
Building Owner
Exchange
Design Team
Version Management by Technical Team Consultant
Analyze Energy Demand and Consumption
Review Energy Analysis Results
Prepare Analysis Report
Ref. Data
BIM Data
ER Energy Analysis Input
Weather Data
Energy Tariff
Analysis Method
ER Energy Analysis Results
ER Energy Analysis Results
Perform Client Energy Analysis
Yes
No
Evaluate Energy Performance, Operating, Costs
Prepare Design Feedback
Yes
No
Valid Energy Analysis BIM?
Validate BIM for Energy Analysis
Ref. Data
BIM Data
ER Energy Analysis Inputs
Energy Tariff
Weather Data
Analysis Method
ER Energy Analysis Results
ER Energy Analysis Inputs
ER Energy Analysis Results
ER Energy Analysis Results (if rejected)
Conceptual Design BIM complete
Prepare / Adjust BIM for Energy Analysis
Export BIM for Analysis
Analyze Energy Demand and Consumption
Review Energy Analysis Result
Yes
No
Results Acceptable?
Prepare Submissions for Review/Approval
Proceed to Design Development
Yes
No
Energy Performance Accepted?
Version Management by Designer

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).

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.

2.3 Semantic Web Technologies in the BIM ecosystem

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 2.3.1). 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 Zhang, 2006).

It is not possible nor desirable to give a full overview on the 'Semantic Web' as the concepts

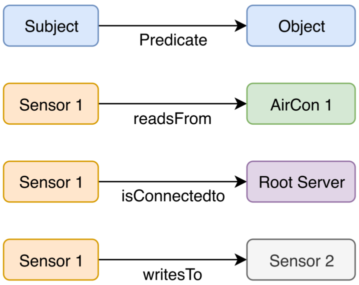


Figure 2.8: RDF triples in the form *subject-predicate-object*. The arrows implies directionality of the relationship.

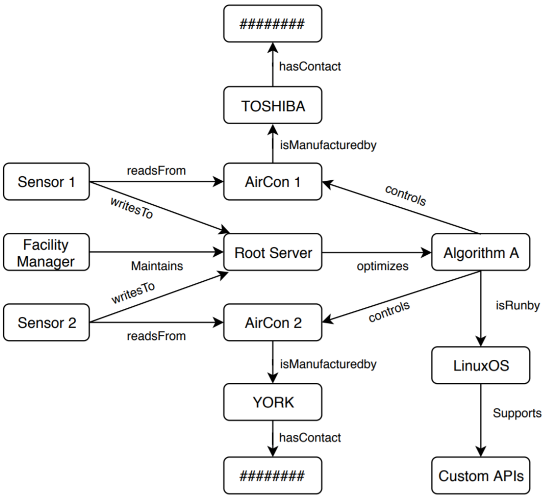


Figure 2.9: An example of an RDF graph (combination of triples) describing some information about sensors in a building connected to different air conditioning units and managed by a root server.

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.

2.3.1 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.

2.3.2 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;
'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.

2.3.3 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

[1] Process of translating the RDF ontology graph structures into a simple format that can be stored.

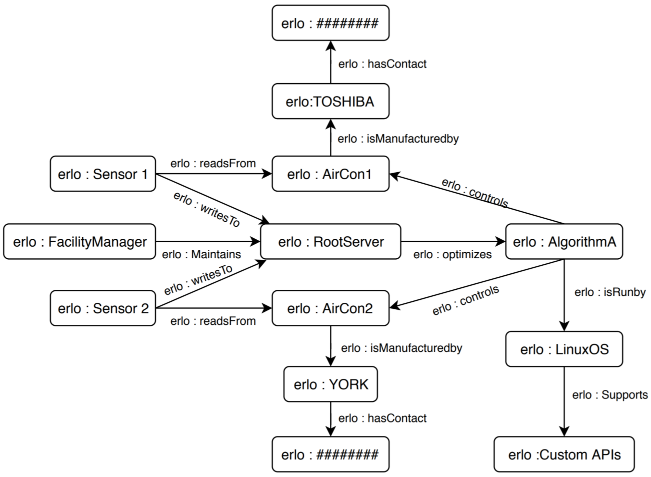


Figure 2.10: The RDF graph with resources labelled using QNames.

such documents require parsers[2] (W3C, 2006) that can read the used serializations and convert them back to RDF triples. Parsing 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).

2.3.4 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 2.3.1) 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 2.3.4).

2. assert domain and range constraints on the use of properties (see subsubsection 2.3.4).

3. introduce class and property hierarchies using the subClassOf and subPropertyOf constructs (subsubsection 2.3.4)

[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

[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**.

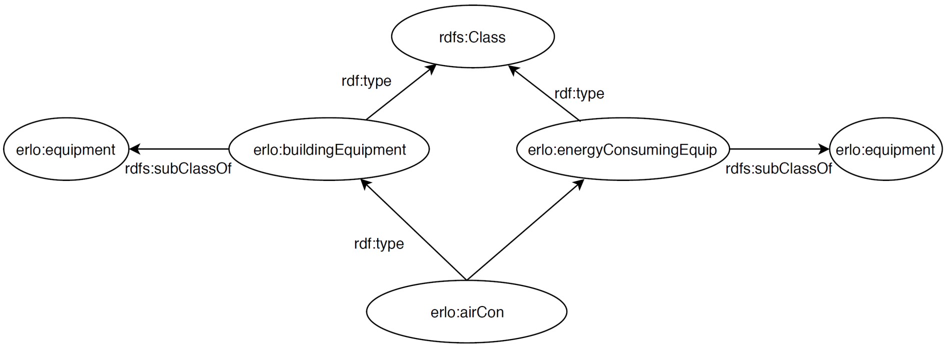


Figure 2.11: Graphical Representation of table 2.1

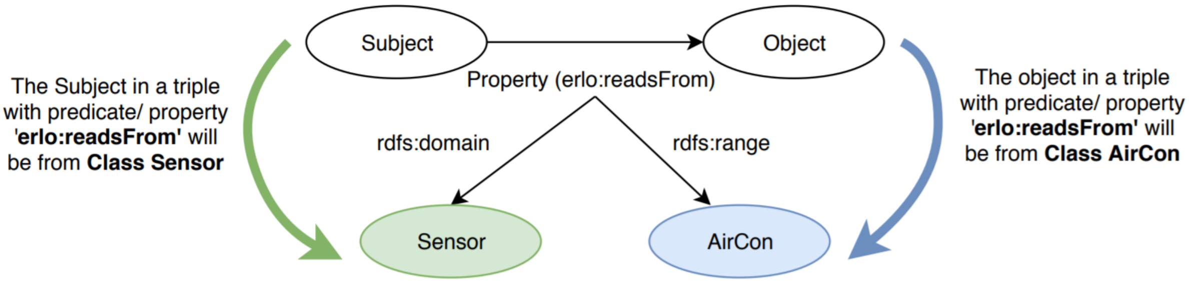

Subject
Object
Property (erlo:readsFrom)
rdfs:domain
rdfs:range
Sensor
AirCon
The Subject in a triple with predicate/ property 'erlo:readsFrom' will be from Class Sensor
The object in a triple with predicate/ property 'erlo:readsFrom' will be from Class AirCon

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

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**.

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

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.

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

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'*.

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 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).

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).

[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.

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 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).

2.3.5 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
```

[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

```
WHERE {
?erlo:ACunits rdf:type ?erlo:BuildingEquipment .
?erlo:erlo:Sensor erlo:readsFrom ?erlo:ACunits .
?erlo:Sensor erlo:hasValue ?erlo:AboveThresholdSensorValue .
}
```

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.

2.3.6 Industry track towards semantic inter-operability of BIMs

Linked Data principles have 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). 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 2.2.2) 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).

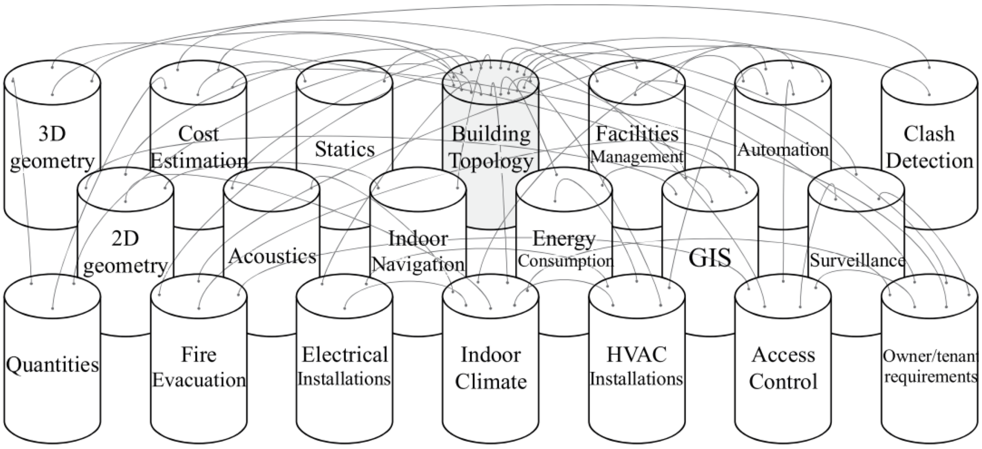


Figure 2.13: The BOT ontology extending to other domain data sets with inter-dependencies. (Extracted from Rasmussen et al. (2017b).

2.3.7 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.

2.3.8 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, 2019b) 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 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.

2.3.9 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. (2014b,a) 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).

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). Much as there have been several research efforts to semantically enrich building information using RDF, very little work has been done to assess how well this data model performs in a reinforcement learning-based building control setting. RL is a mathematical framework for autonomous experience-driven learning and although it has had numerous successful attempts in crafting responsive and context-aware control systems (Han et al., 2018; Kohl and Stone, 2004; Mason and Grijalva, 2019; Yang et al., 2015; Yu and Dexter, 2010), earlier approaches lacked scalability to high dimensional problems due to memory and computational complexity (Strehl et al., 2006). The advent of *Deep Reinforcement Learning* (DRL) has provided the mechanics to overcome these challenges using neural network function approximation, making it possible to deal with high dimensional state-action environments (Mason and Grijalva, 2019). RL, in general, consists of an *agent* interacting in an *environment*,

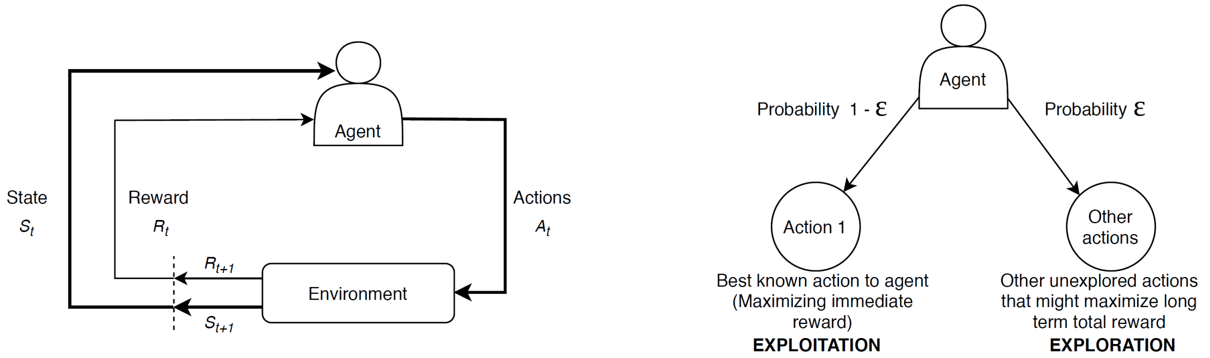


Figure 2.14: *(Left)* Agent Environment Interaction. *(Right)* If $\varepsilon = 0.2$, this means that for 80% of the time the agent chooses to take the best known action (1) while trying to maximize immediate reward, and 20% of the time trying to explore other actions (with equal chance) with the hope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

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 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).

2.3.10 The need for LBD Augmentation in 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 3rd 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. (2016b) 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. (2016b)'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.

2.3.11 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[12] 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. (2018c, 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. (2018b); 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)

2.4 The need for KRL in LBD domains

Certain application fields such as social network analysis, drug discovery in bio-informatics, and fraud detection in e-commerce often deal with immensely interwoven and complex dataset structures. KRL is one aspect of machine learning that has made significant strides in understanding these datasets, however the same cannot be said for its application in building automation. This domain exhibits similarly intricate datasets, and the recent proliferation of connected devices in buildings only compounds the issue. These difficulties arise from a limited understanding of how building automation data is entangled and its resulting peculiarities. Using KRL, patterns in building automation data can be discovered, exploited, and reasoned about to provide transparent and accessible outputs. LBD necessitates the use

[12] 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.

of effective ML methods that can capture the intrinsic underlying structure and patterns. Traditional Statistical Relational Learning (SRL) methods, such as logic programming, inductive logic programming, rule mining, and graphical models, have been widely used for learning from graphs. However, these methods suffer from scalability issues and limited modeling power, and they rely on intractable non-differentiable approaches meaning that neural networks and some well-known methods such as Stochastic Gradient Descent (SGD) cannot be used. One of the most difficult aspects LBD graphs is that they lack spatial locality, which implies that the graph's structure cannot be effectively represented as a fixed grid. Furthermore, the graph isomorphism problem, which refers to the difficulty of determining whether two graphs are structurally identical, adds to the difficulty of learning from LBD graphs. This is a known problem that is neither NP-complete nor solvable in polynomial time, rendering it computationally difficult for traditional SRL methods that rely on manual feature engineering. Also, because LBD graphs can represent multimodal data such as text, numbers, and timestamps, traditional approaches with limited expressivity struggle to model and learn from such complex representations. To overcome these challenges, KRL methods have gained a lot of traction in recent years. These approaches aim to learn representations of nodes and edges that capture the structural patterns of the graph data without the need for manual feature engineering. By doing so, these methods can better capture the global structure of the graph, and handle the multimodality and graph isomorphism problem. In section 2.4.1, we will define relational data the LBD context and discuss the implications of its properties, a foundational basis for highlighting the current gaps this research aims to fill.

2.4.1 A Mathematical Perspective to Learning in LBD Domains

Several definitions of relational data exist in the KRL domain, most of which are similar and are based on either relational database entries or first-order logic ground predicates (Friedman et al., 1999; Heckerman et al., 2007; Richardson and Domingos, 2003). For the purposes of this thesis, a mathematical explanation from both set theory and first-order logic is not only deemed appropriate to define relational data but also highlights the relevance of exploiting the intrinsic relational structure of Resource Description Framework (RDF) in downstream LBD tasks. Relations, in general, define connections between entities, such as whether two rooms have a wall that connects them, whether a person has a specific indoor comfort preference, or whether a sensor is found in a particular space of a building. More precisely, in the domain of set theory and first-order logic, an $\boldsymbol{n}$-ary relation $\mathcal{R}$ over sets $\mathcal{A}_1, \cdots, \mathcal{A}_\mathrm{n}$ is defined as a set of ordered $\boldsymbol{n}$-tuples[13] $\langle a_1, \cdots, a_n \rangle$ where a_i is an element of $\mathcal{A}_\mathrm{i}$ $\forall$ i, $1 \leqslant i \leqslant n$. More intuitively, an $\boldsymbol{n}$-ary relation $\mathcal{R}$ is a subset of the Cartesian product of n sets (Halmos, 1998, Chapter 7) $\mathcal{A}_1, \cdots, \mathcal{A}_\mathrm{n}$, formally expressed as:

$$\mathcal{R} \subseteq \mathcal{A}_1 \times \cdots \times \mathcal{A}_\mathrm{n} \tag{2.1}$$

The relation $\mathcal{R}$ is interpreted as the set of all *existing* relationships, while the Cartesian product is interpreted as the set of all *possible* relationships over the entities in the domains $\mathcal{A}_1, \cdots, \mathcal{A}_\mathrm{n}$.

[13] A tuple is useful for aggregating data that is needed to be considered as a single unit.

A single $\boldsymbol{n}$-tuple $\langle a_1, \cdots, a_n\rangle$ therefore represents a possible relationship between the entities $a_1, \cdots, a_n$, which we simply denote by $\mathcal{R}\langle a_1, \cdots, a_n\rangle$. With this background, it is evident that the RDF data modelling structure adopts *binary or dyadic relations* of the form:

$$\mathcal{R} \subseteq \mathcal{A}_1 \times \mathcal{A}_2 \tag{2.2}$$

There are situations in RDF which require the modeling of $\boldsymbol{n}$-ary relations involving more than two sets of entities. These can be handled efficiently using blank nodes that intrinsically force back a dyadic relational structure. Assuming that *entities* of a particular type, for instance, sensors, rooms, walls and windows are encapsulated within a set, $\mathcal{E}_\mathrm{m}$. Similarly, let a set $\mathcal{L}_\mathrm{n}$ hold possible *literals* values associated with the datatype property of an entity, for instance, a sensor reading, last calibration date of a sensor, U-value of a window glass. Then, any relation $\mathcal{R} \subseteq \mathcal{E}_\mathrm{i} \times \mathcal{E}_\mathrm{j}$ is an object property while $\mathcal{R} \subseteq \mathcal{E}_\mathrm{i} \times \mathcal{L}_\mathrm{j}$ is a datatype property. Typical non-relational machine learning (NRML) settings utilize data that is literal valued and spanning over a single type of entity i.e. consisting relations that take the form $\mathcal{E} \times \mathcal{L}_\mathrm{j}$, with $\mathcal{E}$ denoting the set of all entities of the same type and the sets $\mathcal{L}_\mathrm{j}$ corresponding to the different datatype properties of these entities. Intuitively, $\mathcal{E}$ could contain all sensors in a building and the sets $\mathcal{L}_\mathrm{j}$ could reflect the datatype properties of those sensors like reading, calibration date, location in the building, maintenance date, accuracy etc. NRML makes an independence assumption between the literal values of different entities. For instance the accuracy of a sensor $\mathtt{s}_1 \in \mathcal{E}$ might depend on its other datatype properties like the calibration date, but it is assumed to be independent from the datatype properties of another sensor $\mathtt{s}_2 \in \mathcal{E}$ if $\mathtt{s}_1 \neq \mathtt{s}_1$. However, in a machine learning setting adopting the RDF dyadic relational structure, different entity types can not only exist but also have relationships between them taking the form, $\mathcal{E}_\mathrm{i} \times \mathcal{E}_\mathrm{j}$. These entity-entity relationships introduce rich patterns like homophily[14], stochastic equivalence[15] and global dependencies[16], which patterns can be exploited for collective reasoning in self-learning building control systems towards improved context-aware behaviour. To put this in context, the previous set of sensors $\mathcal{E}_\mathrm{i}$ together with their datatype properties could be complemented by a set of actuators $\mathcal{E}_\mathrm{j}$ and a relation $\mathtt{isConnectedTo} \subseteq \mathcal{E}_\mathrm{i} \times \mathcal{E}_\mathrm{j}$ which indicates which sensor is connected to which actuator. Take, for instance, a sensor observing a certain feature of interest in a building. If this sensor fails and the connected actuator starts deriving wrong control actions, one could implicitly assign credibility of the actuation error to the failed sensor using the existential relation between the two. Because climate control of the indoor environment is a highly dimensional problem (dependent on many parameters that are directly or indirectly related), and often suffers from uncertain data that is incomplete, noisy or even false, necessitates the need for an approach that can expose rich patterns in such data for exploitation by other machine learning methods.

[14] In humans, homophily is the tendency of individuals to associate and bond with similar others.

[15] The fact that entities in a data set can be partitioned into groups such that the observed relationships can be explained via relationships between these groups

[16] Dependencies that affect different types of relations and can possibly range over chains of multiple relationships.

2.4.2 LBD Patterns That are Exploitable for Building Automation

- **Stochastic Equivalence**: In Nowicki and Snijders (2001)'s model, each node in a graph is assumed to belong to an unobserved latent class, and a probability distribution characterizes the associations between each pair of classes. Such a model captures *stochastic equivalence*, a characteristic typically seen in network data in which class members have comparable association patterns. Kemp et al. (2004) made some refinements to the above model. First, the authors made a fundamental distinction between relations and attributes by delineating relations as the notion that defines how different nodes in a network graph interact and attributes being the characteristics of a single node in a network. From this, a generative infinite block model was derived with the assumption that latent classes are associated with the patterns exhibited by node attributes.

- **Homophily**: Social networks are known to be characterised by homophily, the tendency for people from similar backgrounds to connect with one another. This pattern, also known as *autocorrelation*, has been seen in numerous relational data sets and is an important concept in the field of KRL (Jensen and Neville, 2002). Within the building automation context, considering an LBD graph, relationships between nodes with similar characteristics will be stronger than the relationships between nodes having different characteristics.

2.4.3 Comparative Study of KRL Models

Within the context of building control, KRL aims to enable automation agents to understand and reason about complex building data in a manner analogous to that of humans. In KRL, knowledge is typically represented as a set of entities and their relationships. The goal of KRL algorithms is to learn a compact, meaningful representation of this knowledge that can be applied to tasks such as prediction, classification, and recommendation. This section will analyze how Graph Neural Networks (GNNs), translational embedding models, bilinear, tensor, and complex vector models compare and contrast with each other. A synthesis of their strengths and weaknesses will be done with regard to building control and automation.

Graph Neural Networks

A GNN is a neural model that is designed to learn from graph-structured data. At its core, is the concept of message-passing, which allows nodes to communicate with each other by sending and receiving messages along the edges of the graph. Each node receives messages from its neighboring nodes, aggregates them, and combines them with its own features to generate a new representation. GNNs are often used to solve three types of problems,

1. **Node-level problems**: Here, the focus is on node problems such as node classification, regression, and clustering. Node classification attempts to classify nodes into different

groups (classifying sensors based on their type, location, or function). Node regression predicts a continuous value for each node (predicting the energy consumption of an HVAC system in a building). Node clustering attempts to divide nodes into numerous distinct groups, with similar nodes placed in the same group (group together sensors that share similar characteristics or are located in the same area of the building). By applying GNNs to these node-level problems, it is possible to obtain insights into the behavior and performance of various building elements and optimize their control and automation strategies for enhanced energy efficiency and occupant comfort.

2. **Edge-level problems**: GNNs can perform edge-level inferences such as edge classification and link prediction. For example, edge classification can be used to classify the type of relationship between building elements, such as the relationship between a specific sensor and a system. Similarly, link prediction can be used to forecast the existence of links between building elements, such as a light switch and a lighting system.

3. **Graph-level problems**: In graph-level tasks, the goal is to classify entire graphs into different categories based on their topological properties. An example could be determining whether a sensor network comprises motion sensors, temperature sensors, or air quality sensors. Graph-level tasks include graph matching, graph classification, and graph regression. These have several applications in the building automation domain. For graph classification, take for example fault detection in HVAC systems. HVAC systems are essential components of building automation and can account for a considerable amount of a building's energy consumption. Defects in HVAC systems can result in inefficiency, energy waste, and increased maintenance expenses. Therefore, early defect detection is crucial for ensuring optimal system performance. In this case, the HVAC system can be represented as a graph, where each node represents a component (e.g., compressor, evaporator, condenser) and the edges represent their interconnections. By analyzing the structural properties of the graph, we can identify system anomalies and categorize the graph according to the type of defect. Occupancy detection is another application of graph classification. Predicting occupancy is crucial for energy optimization, enhancing comfort levels, and reducing maintenance costs. In this scenario, the building can be represented as a graph, where each node represents a room and the edges represent their interconnections. We can then examine the graph's topology and properties to forecast the occupancy levels of the rooms and classify the graph into distinct categories based on the occupancy patterns.

Graph Convolutional Network (GCN): A common GNN variant is the *GCN*, which uses convolutional operations on graphs to capture structure information. GCNs employ a localized filter that combines information from surrounding nodes, and the filter is applied to nodes in the graph recursively. This method is computationally efficient and can learn representations of local structural information however it may not be appropriate for capturing long-range dependencies or global structural information in the graph typical in BIM-based knowledge graphs. GCN has the ability to manage large-scale graphs with millions of nodes and edges, which are frequent in building automation however, it is prone to over-smoothing, which can result in the loss of essential structural information. Furthermore, due to the vanishing gradient problem, the number of convolutional layers that can be used is limited. Perhaps another limitation of the vanilla GCN architecture is its inability to handle different edge types in a graph. It assumes a single type of edge and treats all edges equally during the message passing and aggregation process. In a graph with multiple edge types, a vanilla GCN would not be able to distinguish between different types of relationships.

For the mathematical intuition behind GCNs, consider an undirected graph $G = (V, E)$, where V is the set of nodes (vertices) and E is the set of edges, and a node feature matrix $X \in \mathbb{R}^{N \times D}$, where N is the number of nodes and D is the number of input features for each node, the GCN aims to learn a node representation matrix $H \in \mathbb{R}^{N \times F}$, where F is the number of output features for each node. The GCN layer can be defined as:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$

where $H^{(l)}$ represents the input node features at layer l, $\sigma(\cdot)$ is the activation function (e.g., ReLU), $W^{(l)}$ is the trainable weight matrix at layer l, and $\tilde{A}$ and $\tilde{D}$ are derived from the adjacency matrix A of the graph. Below is a breakdown of the GCN layer equation:

1. $\tilde{A} = A + I$ is the adjacency matrix of the graph G augmented with self-loops, where I is the identity matrix. The self-loops ensure that the node's own feature is considered during the aggregation process.

2. $\tilde{D}$ is a diagonal matrix with diagonal elements $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$, which represents the degree matrix of the augmented graph.

3. $\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ is a normalization term that symmetrically normalizes the adjacency matrix.

4. $H^{(l)}$ represents the node features at layer l of the GCN.

5. $W^{(l)}$ is the weight matrix that maps the input features $H^{(l)}$ to the output features $H^{(l+1)}$ at layer $l + 1$.

6. $\sigma(\cdot)$ is the activation function applied element-wise to the aggregated neighborhood information.

Graph Attention Network (GAT): Another GNN variant is the *GAT*. It uses attention mechanisms to learn node representations from a graph. GAT weights each node's neighbors based on their significance to the node and aggregates their representations to generate the node's new representation. Attention mechanisms allow the model to focus on the most important relationships and components, making the predictions more accurate and interpretable. Attention methods, on the other hand, can be computationally expensive as they require additional computations to determine the significance of each node or edge. To circumvent this, sparse attention has been proposed in the literature, where only a subset of the graph's nodes or edges is considered for attention computations. Unlike GCNs, GATs are effective at learning representations that capture both local and global structural information. Mathematically, the GAT layer is defined as:

$$H_i^{(l+1)} = \sigma\left(\sum_{j \in N_i} \alpha_{ij} \cdot \left(W^{(l)} \cdot h_j^{(l)}\right)\right)$$

where $H_i^{(l)}$ represents the input node features at layer l for node i, $\sigma(\cdot)$ is the activation function (e.g., LeakyReLU), $W^{(l)}$ is the trainable weight matrix at layer l, and α_{ij} is the attention coefficient. The equation is broken down as follows:

1. For each node i, the GAT layer computes the attention coefficients α_{ij} between node i and its neighbors $j \in N_i$, where N_i represents the set of neighbors of node i.

2. The attention coefficients are computed using a shared attention mechanism, such as a multi-layer perceptron (MLP), followed by a softmax operation to obtain normalized coefficients.

3. $h_j^{(l)}$ represents the output features at layer l for neighbor node j.

4. $W^{(l)}$ is the weight matrix that maps the input features $h_j^{(l)}$ to the output features $H_i^{(l+1)}$ for node i at layer $l + 1$.

5. The weighted sum of the neighbor features is computed using the attention coefficients, and the result is passed through the activation function $\sigma(\cdot)$ to obtain the final output features $H_i^{(l+1)}$ for node i at layer $l + 1$.

Note that the equation defined represents a single GAT layer. In practice, multiple GAT layers can be stacked to capture deeper and more complex graph representations.

Graph Sampling and Aggregation (GraphSAGE): *GraphSAGE* is another GNN variation that is widely used for learning node embeddings in large-scale graphs. Hamilton et al. presented the approach in 2017 and it builds further on the concept of message passing. The core idea underlying GraphSAGE is to learn node representations by sampling and aggregating node neighbors' representations. The approach, in particular, samples a specified number of neighbors for each node in the graph and aggregates their representations using a neural network. The aggregated representation is then utilized to generate a new representation for the node, together with the node's own features. This procedure is performed several times to learn representations at various layers of the graph. Scalability is one of GraphSAGE's key features. The approach can learn node embeddings for very large graphs rapidly by pooling information from a small number of neighbors. Furthermore, GraphSAGE can handle a variety of input features, such as occupancy data and multiple sensor readings, making it suitable for a wide range of building control and automation tasks and it can sit at the core of many recommender systems in the building domain. GraphSAGE has progressively been improving to handle more complex graph structures, such as heterogeneous graphs with different node and edge types. Unlike GCN and GAT, GraphSAGE is a good choice for applications that involve large-scale graphs and where scalability is a concern. Mathematically, the GraphSAGE layer is defined as:

$$h_i^{(l+1)} = \sigma\left(W \cdot \text{CONCAT}(\text{AGGREGATE}(h_j^{(l)}, \forall j \in N_i))\right)$$

where $h_i^{(l)}$ represents the input node features at layer l for node i, $\sigma(\cdot)$ is the activation function (e.g., ReLU), W is the trainable weight matrix, AGGREGATE$(\cdot)$ denotes the aggregation function that combines the features of neighboring nodes, and CONCAT$(\cdot)$ concatenates the aggregated features. The above equation is broken down as follows

1. For each node i, the GraphSAGE layer aggregates the features of its neighboring nodes $j \in N_i$, where N_i represents the set of neighbors of node i.

2. The aggregation function AGGREGATE$(\cdot)$ computes a summary representation of the neighboring node features.

3. The aggregated features are concatenated using the CONCAT$(\cdot)$ operation to obtain a combined representation.

4. The combined representation is multiplied by the trainable weight matrix W and passed through the activation function $\sigma(\cdot)$ to obtain the final output features $h_i^{(l+1)}$ for node i at layer $l + 1$.

Note that the equation defined represents a single GraphSAGE layer. In practice, multiple GraphSAGE layers can be stacked to capture deeper and more complex graph representations

Relational Graph Convolutional Network (R-GCN): *R-GCN* is a form of GCN that can learn node embeddings in heterogeneous graphs with diverse types of nodes and edges. Schlichtkrull et al. proposed R-GCN in 2018 and it is based on the concept of message passing in GCNs. Nodes in heterogeneous networks can have diverse types, as can the relationships between them. In a building automation system, for example, nodes can represent various building components such as rooms, HVAC systems, and lighting systems, and edges can represent different interactions between these components such as airflow, temperature, and power usage. R-GCN can learn embeddings for these various types of nodes and edges by applying varying weights to message-passing operations based on the node and edge types. R-GCN represents the relationship between different types of nodes and edges using a shared weight matrix. The weight matrix is learned during training and can capture the relationship between different types of nodes and edges. R-GCN performs message-passing operations on the graph by aggregating and updating the embeddings of surrounding nodes based on the type of edge that connects them. Because of its ability to handle various types of input types, it is a viable tool for developing building automation applications. Given a graph $G = (V, E, R)$, where V is the set of nodes (vertices), E is the set of edges, and R is the set of edge types, and a node feature matrix $X \in \mathbb{R}^{N \times D}$, where N is the number of nodes and D is the number of input features for each node, the R-GCN aims to learn a node representation matrix $H \in \mathbb{R}^{N \times F}$, where F is the number of output features for each node. The R-GCN layer can be defined as:

$$h_i^{(l+1)} = \sigma \left(\sum_{r \in R} \sum_{j \in N_i^r} \frac{1}{c_{ir}} \cdot W_r^{(l)} \cdot h_j^{(l)} \right)$$

where $h_i^{(l)}$ represents the input node features at layer l for node i, $\sigma(\cdot)$ is the activation function (e.g., ReLU), N_i^r is the set of neighbors of node i connected by edges of type r, c_{ir} is the normalization constant for the number of neighbors connected by edge type r, and $W_r^{(l)}$ is the trainable weight matrix associated with edge type r. Here's the breakdown of the equation:

1. For each node i, the R-GCN layer computes the aggregation of features from its neighbors j connected by edges of type r ($j \in N_i^r$).

2. The sum is taken over all edge types r in the graph.

3. c_{ir} is the normalization constant that accounts for the number of neighbors connected by edge type r to node i.

4. $W_r^{(l)}$ is the weight matrix associated with edge type r at layer l. Each edge type has its own weight matrix to capture different transformations for different relationships.

5. The weighted sum of the neighbor features is computed and passed through the activation function $\sigma(\cdot)$ to obtain the final output features $h_i^{(l+1)}$ for node i at layer $l + 1$.

Note: This equation represents a single R-GCN layer. In practice, multiple R-GCN layers can be stacked to capture deeper and more complex graph representations.

Translation embedding models

Translational embedding models are designed to capture the ”translation” process between entities and relationships in a knowledge graph, where the output entity is related to the input entity through a specific relationship. There are different kinds of translational embedding models, such as TransE, TransH, and TransR. Translating Embeddings (TransE) is a translational embedding model that learns vector representations of knowledge graph entities and relations. Each entity and relation in this paradigm is represented by a low-dimensional vector in a continuous vector space.

TransE: TransE, proposed by Antoine Bordes et al.,2013, is a simple yet effective model for knowledge representation learning. It represents entities and relationships as vectors in a low-dimensional embedding space. The key idea of TransE is to interpret relationships as translations between entities. The scoring function of TransE measures the plausibility of a triple (h, r, t) by computing the distance between the head entity h, the relationship r, and the tail entity t in the embedding space. The scoring function for TransE is defined as:

$$f(h, r, t) = ||h + r - t||_2$$

To train TransE, a margin-based ranking loss function, such as the hinge loss or the softmax loss, is commonly used to optimize the embeddings. TransE is efficient and easy to implement, making it a popular choice for many KRL tasks. However, it has certain limitations that make it less suitable for capturing the complexity and heterogeneity of relationships in BIM-based knowledge graphs. Let's delve deeper into these limitations and explore how models like TransH and TransR address them more effectively.

One limitation of TransE is its assumption that relationships can be represented as simple translations in the embedding space. While this assumption may work well for certain knowledge graphs with straightforward relationships, it fails to capture the intricacies and diversity of heterogenous relationships in BIM-based knowledge graphs. In building automation and control, relationships between entities are often multi-faceted, involving various factors such as containment, control, measurement, and more. Treating all relationships as translations oversimplifies the true nature of these relationships and can lead to learning malformed embeddings.

Another limitation of TransE is its lack of differentiation between entities in different relationship contexts. In TransE, entities are represented uniformly across all relationships, which limits its ability to capture the heterogeneity of relationships. For example, an entity's role in one relationship may be entirely different from its role in another relationship.

TransH: TransH addresses some of the limitations of TransE by considering the heterogeneity and complexity of relationships in knowledge graphs. TransH introduces relationship-specific hyperplanes to capture the diverse transformations associated with different relationships. It models the interaction between entities and relationships on these hyperplanes. The scoring function for TransH is defined as:

$$f(h, r, t) = ||h_{\perp r} + r - t_{\perp r}||_2$$

Here, $h_{\perp r}$ and $t_{\perp r}$ denote the projected representations of the head and tail entities onto the hyperplane associated with the relationship r. By using relationship-specific hyperplanes, TransH provides a more flexible and expressive approach for capturing complex relationships in BIM-based knowledge graphs.

TransR TransR, proposed by Yankai Lin et al. in 2015, extends TransE by introducing separate mapping matrices for entities and relationships. It allows entities to have different representations under different relationships. TransR decouples the entity and relationship embeddings, enabling more diverse and accurate modeling of relationships. The scoring function for TransR is defined as:

$$f(h, r, t) = ||M_r h + R_r t - M_r t||_2$$

In this equation, M_r and R_r are the relationship-specific mapping matrices, and $M_r t$ represents the projected representation of the tail entity under relationship r. By utilizing these mapping matrices, TransR captures the varying roles and characteristics of entities across different relationships, enhancing its ability to represent complex relationships in BIM-based knowledge graphs.

2.4.4 Summary of Research Gaps Identified

The literature review has shown that in recent years, the advent of SWT has been a driving force in advancing the field of BIM, leading to a significant maturation of BIM-based knowledge graphs. SWTs have provided the capability to represent complex relationships within BIM data, effectively transforming this data into structured knowledge graphs. This transformation has paved the way for intelligent decision-making processes in various building and infrastructure projects.

Concurrently, KRL has seen significant development in various other domains. It has shown immense promise in fields like bioinformatics, where it has been used to map and understand complex biological relationships and processes. KRL employs sophisticated algorithms to capture and learn from the patterns and relationships inherent in knowledge graphs. Despite its success in other domains, the application of KRL to BIM-based knowledge graphs has remained largely unexplored, presenting a potential research gap among many others as delineated below:

1. **Developing a Robust Methodology for Applying KRL to BIM**: There's a significant need for establishing a robust methodology that applies KRL techniques to BIM-based knowledge graphs, and any new methods developed would need rigorous validation. This could involve testing the new methods in a variety of real-world scenarios, such as different building types or sizes, to ensure they perform effectively under diverse conditions.

2. **Real-time Adaptation and Learning**: In the dynamic world of building control and automation, real-time adaptation is critical and KRL could be instrumental. Let's unpack this with a closer look at an intelligent HVAC system that has been represented as knowledge graph where nodes represent different components of the system, and the edges denote the relationships between these components. Further, this graph could be enriched with additional nodes representing external factors like environmental

conditions, energy prices, or occupant preferences. A KRL model could be used to learn embeddings for all these nodes and their relationships. The key feature of these embeddings is their ability to capture and quantify both local and global, or distant, information. Local information involves direct interactions, such as the relationship between the outside temperature and the performance of the HVAC system. On the other hand, global information refers to indirect interactions, such as the impact of a series of hot days (recorded as a trend in environmental conditions) on the overall energy usage of the HVAC system. In the context of real-time adaptation, as new data streams in - say, a sudden drop in outdoor temperature or a change in energy tariff - the KRL model updates the relevant node embeddings in real-time. These updated embeddings reflect the new state of the building and its environment, influencing subsequent predictions or control actions. This dynamic updating of embeddings allows the HVAC system to learn and adapt continuously. For instance, if the model learns that the building occupants often prefer a slightly cooler temperature in the afternoons, it could start pre-cooling the building just before this period, optimizing comfort while being energy-efficient.

3. **Integration with Other Data Sources**: BIM creates data-rich models that represent the physical and functional characteristics of buildings. These models, when transformed into knowledge graphs, provide a structured representation of the relationships and interactions within a building system. While these BIM-based knowledge graphs are highly effective at handling structured data, they often struggle to incorporate unstructured data. Unstructured data in the building industry can take many forms, such as maintenance logs, email communications, user feedback, sensor data, and more. This type of data may contain valuable insights and information, but its lack of structure makes it challenging to integrate within a BIM-based knowledge graph. KRL, on the other hand, offers an innovative approach to dealing with this challenge. In essence, KRL models learn the underlying patterns and structures within data, effectively capturing the rich semantics of both entities and their relationships. These learned patterns are represented as embeddings or vectors in a continuous space. Consider a scenario where we have maintenance logs in a text format. The logs could contain important details about recurring issues, the time taken to resolve them, specific parts that were replaced, etc. A KRL model could parse this text and learn embeddings that capture the relationships between different components, issues, and their resolutions. These embeddings could then be integrated with theBIM-based knowledge graph, effectively enriching it with insights gleaned from the maintenance logs. The integration of these additional data sources could lead to a more comprehensive knowledge base, benefiting various aspects of building control and automation. For example, a system that has access to a comprehensive knowledge base could predict potential failures based on historical maintenance logs, automate alerts for maintenance checks, or even optimize resource allocation for repairs and maintenance. The integration of unstructured data sources with BIM-based knowledge

graphs, made possible by KRL techniques, represents an important research opportunity with significant potential benefits for the building industry.

4. **Scalability**: Scalability in KRL refers to the ability of the model to handle an increase in data size without sacrificing performance or efficiency. In the context of BIM, data scales with the size and complexity of the building or infrastructure project. Large-scale projects could produce vast amounts of data, making scalability a crucial factor when applying KRL to BIM data. KRL techniques essentially learn and generate a continuous representation or embedding for every entity (nodes) and relationship (edges) in a knowledge graph. The model then uses these embeddings for tasks such as link prediction, entity resolution, and knowledge discovery. As the number of entities and relationships in the graph increases (which is typical for large-scale BIM projects), the model needs to generate and manage a corresponding increase in embeddings. Now, consider an example of a large residential complex or a smart city infrastructure project. The BIM data from such a project could include multiple buildings, each with its own set of systems like HVAC, lighting, security, etc., along with their respective components. Further, there could be city-level entities such as traffic systems, energy grids, public facilities, etc. Creating a knowledge graph from such comprehensive BIM data would result in a large number of entities and relationships, thereby necessitating a KRL model capable of efficiently learning and managing a substantial amount of embeddings. However, as the size of the knowledge graph grows, many KRL models face challenges, such as increased computational requirements and longer training times. Additionally, maintaining the quality of the learned embeddings can also become challenging with increased scale. Therefore, the development of scalable KRL methods that can handle large-scale BIM data effectively is a significant area of research. One possible direction of research could be exploring distributed learning methods or parallel computing techniques, which can leverage multiple computing resources to handle large datasets. Alternatively, research could investigate more efficient embedding techniques that can generate compact yet informative representations for entities and relationships, thereby reducing the computational load.

5. **Privacy and Security**: Applying KRL to BIM-based knowledge graphs involves learning from a vast amount of data. This data may include sensitive information related to building systems, operational details, or even personal data associated with building occupants in certain contexts, like smart homes or offices. As such, the application of KRL in this context presents a need for careful consideration of data privacy and security. Data privacy concerns arise when the data used or generated by the KRL models can lead to the disclosure of information that should remain confidential. For instance, a model trained on HVAC usage data might infer patterns related to building occupancy. If this inferred information is disclosed or misused, it could potentially infringe on the privacy of the building's occupants. Furthermore, as KRL models learn to represent knowledge graph entities and relationships as embeddings, there's a risk

that these embeddings could inadvertently capture and reveal sensitive information. A third party with access to the embeddings might be able to reverse-engineer them to extract sensitive data. Security concerns, on the other hand, relate to the potential for data breaches or unauthorized access to the data or the KRL models themselves. For example, an attacker gaining access to a building's BIM-based knowledge graph could manipulate the data or the KRL model, leading to incorrect predictions or even system failures. Addressing these privacy and security concerns is critical. Potential approaches could include:

- **Data anonymization**: Before applying KRL, the data could be anonymized to remove or obfuscate any identifying information. This could help protect individual privacy while still allowing the model to learn from the underlying patterns in the data.
- **Differential privacy**: This technique adds carefully calibrated noise to the data or the model's outputs, helping to prevent the leakage of sensitive information without significantly affecting the accuracy of the model's predictions.
- **Encryption**: Data could be encrypted while at rest and in transit, protecting it from unauthorized access. There are even emerging techniques, such as homomorphic encryption, which allow computations to be performed directly on encrypted data.
- **Access controls and auditing**: Implementing strict access controls on the data and the KRL models can help prevent unauthorized access, while auditing can help track and respond to any potential security incidents.

6. **Validation of KRL Methods**: Applying KRL to BIM-based knowledge graphs within the realm of building control and automation is a nascent field and any new methods developed for this application will require rigorous validation to ensure that they perform effectively in real-world scenarios. This validation process could be multi-faceted, testing the methods under a variety of conditions, such as diverse building types (commercial, residential, industrial, etc.), different sizes of buildings (single-family homes, apartment complexes, high-rises, etc.), and various operational scenarios (peak occupancy, minimal occupancy, different seasons, etc.). Additionally, validating these KRL methods would involve assessing their performance in terms of both accuracy and generalizability. Accuracy measures how well the methods can predict or infer missing information in the BIM-based knowledge graph, while generalizability assesses how well the methods perform when applied to new, unseen data. Typically, the performance of KRL methods is evaluated using a range of metrics such as ac Mean Rank (MR), Mean Reciprocal Rank (MRR), Hits@N, ROC, and Area under the ROC Curve (AUC). However, these metrics may not work "out of the box" when it comes to evaluating KRL methods in a building context. Let's delve deeper into some of these metrics:

- **MR**: This metric calculates the average rank of the true entities or relations among all the candidate entities or relations. However, in the building context, the notion of ”rank” might be less meaningful because the relationship between entities may be more complex and multi-faceted than can be captured by a single ranking.
- **MRR**: This metric is often used for link prediction tasks and calculates the mean of the reciprocals of ranks for true entities or relations. Like Mean Rank, this metric may not fully capture the complexity of relationships in a building context.
- **Hits@N**: This metric measures the percentage of true entities or relations that appear in the top N of the ranking. However, given the complexity of building systems, a strict top N ranking might not always be meaningful or useful.
- **ROC and AUC**: These metrics measure the trade-off between true positive rate and false positive rate for different thresholds. However, they are typically used for binary classification tasks, whereas many tasks in building control and automation could be multi-label or regression tasks.

Given the unique context of buildings and their operation, traditional KRL evaluation metrics may need to be tailored or even combined to provide a more meaningful assessment. For example, accuracy metrics might need to be combined with measures of energy efficiency, comfort, or other building-specific objectives. Additionally, new, domain-specific evaluation metrics might need to be developed to better reflect the complexity and multi-objective nature of building control and automation.

Much as this review has identified several gaps, this thesis focuses on gap 1, gap 3 and gap 6 while providing necessary recommendations to closing other gaps.

Chapter 3

Research Methods

3.1 Linked Building Data Modeling

In this study, a refined methodology for Linked Building Data Modeling is proposed, developed with a primary focus on maintaining relevance and tractability for the domain-specific challenges in building control. The approach leverages the potential of Semantic Web and the Resource Description Framework (RDF), recognized for their flexibility and openness, but also mindful of the complexities that can make the data modeling process daunting, if not properly scoped. A crucial facet of the methodology is the delineation of competency questions, which provide explicit guidance on the specific objectives the data model needs to satisfy. These questions are critical to ensure the relevance of the developed model for the problem at hand. For instance, a question such as "What is the current power consumption of all HVAC systems in a given building?" specifies the necessary components of the data model pertaining to HVAC systems and their power consumption. Another aspect of the methodology involves the efficient use of extant ontologies, particularly those managed by the World Wide Web Consortium (W3C). The adoption of such ontologies affords standardized vocabularies and structures that aid in the representation of information. A specific ontology of interest within this context is the ifcOWL ontology. This ontology provides an exhaustive representation of building information in RDF format. However, due to the sheer scale of ifcOWL, implementing it in its entirety would risk an overly convoluted model. Consequently, where necessary, the methodology prescribes the adoption, extension, and integration of smaller, more manageable modules from ifcOWL. This nuanced approach facilitates the use of vocabulary that suits the specific requirements of the study. To summarize, the LBD modeling methodology followed is outlined in the following steps:

1. **ifcOWL Analysis and Modular Ontology Identification**: Conducting a shallow-depth analysis of the structural, vocabularic, and relational aspects of the ifcOWL ontology with the goal of identifying suitable ifcOWL sub-modules that offer concepts and relationships that align with building automation, with a preference for W3C-managed ontologies due to their extensibility and standardization.

2. **Competency Questions Definition and Data Modeling:** Defining precise questions

that the data model must address, which serves as a guide for maintaining focus and relevance throughout the modeling process. This is followed by utilizing the modules identified in Step 1 for data modeling, extending them where necessary, and integrating them while ensuring that they work in cohesion to address the defined competency questions.

3. **Data Model Validation, Model Iteration, and Refinement**: SPARQL queries are defined to validate the data model by verifying the presence of expected data, the integrity of relationships, and the adherence to established rules and constraints. Based on the validation outcomes, the data model is iteratively refined by revising modules where necessary, introducing new ones or modifying integration methods.

3.1.1 ifcOWL Analysis and Modular Ontology Identification

The ifcOWL ontology is the semantic web rendition of the Industry Foundation Classes (IFC), an open data model that promotes interoperability in the building industry. Developed by the International Alliance for Interoperability (IAI), now buildingSMART International, the IFC has been widely used for many years in the construction and facility management domains. It represents a comprehensive and complex schema for describing building and construction data. The IFC data model has been formalized in the Web Ontology Language (OWL) as ifcOWL, facilitating its integration into the semantic web ecosystem. ifcOWL comprises a rich set of classes, properties, and relationships, covering a broad range of building and construction concepts, including architectural, structural, and MEP (Mechanical, Electrical, Plumbing) elements, spaces, materials, geometric representations, project organization, and more. The ifcOWL ontology is organized around key classes like IfcProduct (representing building objects), `IfcRelationship` (denoting relationships between objects), `IfcPropertyDefinition` (describing properties of objects), among others. The complexity of the ontology is underpinned by an intricate network of relationships, which allows for modeling intricate interdependencies and interactions between different building components. The ontology includes more than just the physical components of a building. It also comprises a temporal aspect (through IfcProcess and IfcEvent classes), enabling the modeling of building lifecycle events such as construction, maintenance, renovation, and demolition activities. As per the IFC4 Add2 release, ifcOWL has about 770 classes, approximately 1190 object properties, and around 60 datatype properties. This granularity is one of ifcOWL's main strengths, offering a level of detail that surpasses many other building-related ontologies. It caters to a wide array of applications from architectural design, structural analysis, and energy performance assessment, to facility management. However, this granularity can also pose challenges. The large number of classes and properties, coupled with its complex relationships, can make it unwieldy for specific use cases, especially those not requiring such extensive detail. Additionally, the complexity of the ontology could lead to higher computational demands for querying and inference, potentially hindering performance in real-time applications like building control and automation. Given this, smaller and more focused ontologies are deemed more suitable when developing an LBD Model for building control and automation, and these are delineated below.

1. **BOT (Building Topology Ontology)**: BOT provides a basic vocabulary for building data with a particular focus on the topological aspects, covering core concepts like Site, Building, Storey, Space, and Element. Its simpler structure makes it easier to manage and extend for building control automation tasks such as energy usage analysis at the level of a single building or across a group of buildings.

2. **SSN (Semantic Sensor Network) and SOSA (Sensor, Observation, Sample, and Actuator)**: SSN and its lightweight version SOSA offer a framework to describe sensors and their observations, which is essential for a building control system. The ontologies can be used to describe the variety of sensors in a building (temperature, light, motion, etc.) and the data they generate, providing essential information for automating tasks like HVAC control or lighting adjustment.

3. **SEAS (Smart Appliances REFerence)**: SEAS offers an ontology to describe smart appliances and their communication with the grid, making it particularly useful for modeling smart devices in building automation. It could help in managing and analyzing data from smart appliances, which is crucial for tasks like demand response or energy efficiency optimization.

4. **OPM (Ontology for Property Management)**: OPM provides a schema for representing property and property value related information. This could be very helpful for managing and querying metadata of a building or its parts (like a sensor's accuracy or a device's power rating) in building control and automation scenarios.

5. **SAREF (Smart Appliances REFerence)**: SAREF is a shared model of consensus that facilitates the matching of existing assets in the smart appliances domain. It provides a standard to ensure interoperability and seamless communication between different devices, a key requirement for any modern building control system.

3.1.2 Competency Questions Definition and Data Modeling

1. **Competency Question 1**: *How to semantically describe the high-level concepts of the building topology in a way that formulates a semantic extension baseline for describing low-level building information relevant for indoor environment monitoring and control?*

   To keep the Linked Building Data Model modular and extensible, this work adopts to use several domain-specific ontologies that provide semantic vocabulary necessary to describe the building information of interest. Specific to this competency question, the **Building Topology Ontology (BOT)**[1] (Rasmussen et al., 2017b, 2019b) is deemed appropriate for encoding meaningful relationships between the main sub-components

[1] https://w3c-lbd-cg.github.io/bot/

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

of a building (site, building, storey and space) using a highly modular and simplistic set of semantic blocks. 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. Figure 3.1 provides an intuitive description of how BOT is used to describe a site, `<UNM>`, having a building, `<Block_B>`, containing a storey, `<Floor_3>` with a certain space, `<Office_204>`. The respective entity connections are made via the object properties; `bot:hasBuilding`, `bot:hasStorey` and `bot:hasSpace`. Explicitly asserted relationships/properties are shown by the solid line arrows while those that are automatically inferred, by the dotted line arrows. The back-end inference rules at play here, are defined in BOT via the ranges and domains of the aforementioned object properties (see table 3.1 of report 1). A corresponding machine-readable serialization of the data model is shown in listing 3.1. With a basis of BOT, semantic extensions can be provided to describe low-level details about specific features of interest contained in the space, `<Office_204>` such as sensors and walls via another object property, `bot:containsElement` as the extension point (see competency question 2).

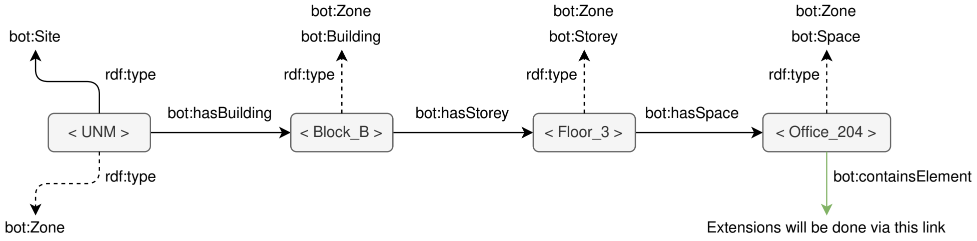


Figure 3.1: Using BOT classes and properties to describe the high level semantic topological details of a building

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix bot: <https://w3id.org/bot#> .
```

```

# Site location (UNM) for some building of interest (Block_B)
<UNM> a bot:Site ;
  bot:hasBuilding <Block_B> .

# Block_B has some storey of interest Floor_3
<Block_B> bot:hasStorey <Floor_3> .

# Floor_3 has some space(zone) of interest Office_204
<Floor_3> bot:hasSpace <Office_204> .
```

listing 3.1: Turtle serialization of the information modelled in figure above

2. **Competency Question 2**: *How to semantically describe a feature of interest within an indoor thermal zone while encoding its parametric properties, corresponding property values and property states in a way that allows tracking changes, deletions and revisions?*

   The **Semantic Sensor Network (SSN)**[2] ontology is chosen for describing relational information about sensors. At its core, exists a lightweight but self-contained ontology **(Sensor, Observation, Sample and Actuator (SOSA))** encapsulating elementary classes necessary for the semantic description of features of interest and their properties, sensor observations, and feature sampling procedures to describe tractable sensor observations and actuation behaviour. Furthermore, we avail semantic extensions to SSN using the **Smart Energy Aware Reference Ontology (SEAS)**[3]. SEAS is an ecosystem of data models/ modules that together, provide semantic vocabulary to describe physical systems, their interrelations, and specific to this work, their energy consumption behaviour and corresponding optimization strategies. Among these, is the `seas:FeatureOfInterestOntology`[4] and `seas:EvaluationOntology`[5] core modules, which are used for parametric property management. However, these natively have no semantics to encode property states in a way that can be tracked over time. For this, the **Ontology for Property Management (OPM)**[6] is deemed relevant, and the specific classes and properties it provides for this work are provided in the table 3.3. To illustrate the usage of these ontologies for modeling the information requirements that satisfy competency question 2, a direct semantic extension is made to the `bot:Space`, `<Office_204>`. First, it is defined as a `sosa:FeatureOfInterest` having two properties; `<Office_204#temperature>` and `<Office_204#humidity>`, both defined via the relation `ssn:hasProperty`. To stay complaint with OPM and satisfy competency question 2, both properties are required to have atleast one `has:propertyState` relation which provides an encoding of property state that can be tracked for any changes over time. The OPM ontology also specifies that as a minimum,

[2] https://www.w3.org/TR/vocab-ssn/
[3] https://ci.mines-stetienne.fr/seas/index.html
[4] https://w3id.org/seas/FeatureOfInterestOntology
[5] https://w3id.org/seas/EvaluationOntology
[6] https://w3c-lbd-cg.github.io/opm/

a property state should have a value and preferably, a generation time, an assignment that is respectively done via the properties; `schema:value`, from `schema.org` and `prov:generatedAtTime`, from the Provenance Ontology. `<Office_204>` is further defined to have two elements which are both walls. One wall, `<Office_204/east>`, is located on the eastern side of the room and the other wall, `<Office_204/south>`, on the southern side. Each of them is described as both a `sosa:FeatureOfInterest` and a `sosa:Platform`. The latter simply means that each of the walls hosts another entity, in this case, a `<NodeMCU>` which is a Printed Circuit Board (PCB) that is also a `sosa:Platform` meant to host a DHT22 temperature and humidity sensor. Because the temperature and humidity properties are defined on the `<Office_204>` entity, but are implicitly being measured directly on the inside walls via the embedded sensors, it is necessary to describe each wall as a `sosa:Sample`. A more intuitive graphical description of this data modelling process is provided in figure 3.2 together with its serialization in listing 3.2. Another ontology that this work might adopt for the explicit definition of complex functionality of smart appliances and their controllability, is the **Smart Appliances REFerence Ontology (SAREF)**[7]. The starting point of the SAREF data model is a device. Currently, majority of the semantic vocabulary for describing device controllability has been availed by the SEAS, SSN, SOSA and BOT ontologies however, should the need arise for more explicit description of HVAC systems and their energy consumption behaviour, SAREF extensions are adopted.

Classes	Properties
`seas:ElectricPowerSystem`	`seas:isPoweredBy`
`seas:TemperatureEvaluation`	`seas:optimizes`
`seas:AgentComfortEvaluation`	`seas:thermalTransmittance`
`seas:MaximumComfortableEvaluation`	`seas:relativeToAgent`
`seas:MinimumComfortableEvaluation`	`seas:evaluatedSimpleValue`
`seas:Battery`	`seas:hasTemporalContext`

Table 3.2: SEAS classes and properties adopted for this work's data model

Classes	Properties
`opm:Assumed`	`opm:hasPropertyState`
`opm:CurrentPropertyState`	
`opm:PropertyState`	
`opm:Confirmed`	
`opm:OutdatedPropertyState`	
`opm:Deleted`	

Table 3.3: OPM classes and properties adopted for this work's data model

[7] https://sites.google.com/site/smartappliancesproject/ontologies/reference-ontology

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix bot: <https://w3id.org/bot#> .
@prefix xsd:  <http://www.w3.org/2001/XMLSchema#> .
@prefix cdt:   <http://w3id.org/lindt/custom_datatypes#> .
@prefix schema: <http://schema.org/>.
@prefix sosa: <http://www.w3.org/ns/sosa/> .
@prefix ssn: <http://www.w3.org/ns/ssn/> .
@prefix seas: <https://w3id.org/seas/> .
@prefix opm: <https://w3id.org/opm#> .
@prefix prov:  <http://www.w3.org/ns/prov#> .

# Office_204 (FOI) hosts some 2 walls at the east and south that will host
# some sensors. The space also has two properties temperature and humidity
<Office_204> a sosa:FeatureOfInterest ;
  bot:containsElement <Office_204/east>, <Office_204/south> ;
  ssn:hasProperty <Office_204#temperature> , <Office_204#humidity> .

# Office_204 east side wall to host a NodeMCU board with
# a DHT22 temp and  hum sensor.
<Office_204/east> a sosa:FeatureOfInterest , sosa:Sample , sosa:Platform ;
  sosa:hosts <NodeMCU_1> .

# Office_204 south side wall to host a NodeMCU board with a DHT22 temp and
# hum sensor.
<Office_204/south> a sosa:FeatureOfInterest , sosa:Sample , sosa:Platform ;
  sosa:hosts <NodeMCU_2> ;

# DESCRIPTION OF PCB BOARDS HOSTING THE SENSORS
##############################################################

# NodeMCU 1 board hosted by the office_204 east side wall.
<NodeMCU_1> a ssn:System , sosa:Platform ;
  sosa:hosts <DHT22/01> ;
  ssn:hasSubSystem <DHT22/01> .

# NodeMCU 2 board hosted by the office_204 south side wall.
<NodeMCU_2> a ssn:System , sosa:Platform ;
  sosa:hosts <DHT22/02> ;
  ssn:hasSubSystem <DHT22/02> .

# Assigning a state to the temperature property of Office #204
<Office_204#temperature>
  opm:hasPropertyState <Office_204#temperature_state_48906948_er8t78> .

# Assigning semantics to Office_204#temperature_state_48906948_er8t78 state
<Office_204#temperature_state_48906948_er8t78>
  a opm:Confirmed ,
    opm:CurrentPropertyState ;
  schema:value "30.5 Cel"^^cdt:temperature ;
  prov:generatedAtTime "2020-07-28T16:41:17.711+02:00"^^xsd:dateTime.

# Assigning a state to the humidity property of Office #204
<Office_204#humidity>
  opm:hasPropertyState <Office_204#humidity_state_40039548_gktiy8> .

# Assigning semantics to Office_204#humidity_state_40039548_gktiy8 state
<Office_204#humidity_state_40039548_gktiy8>
  a opm:Confirmed ,
    opm:CurrentPropertyState ;
  schema:value "85.0 %"^^cdt:ucum ;
  prov:generatedAtTime "2020-07-28T16:41:17.711+02:00"^^xsd:dateTime.
```

listing 3.2: Turtle serialization of the information modelled in figure 3.2.

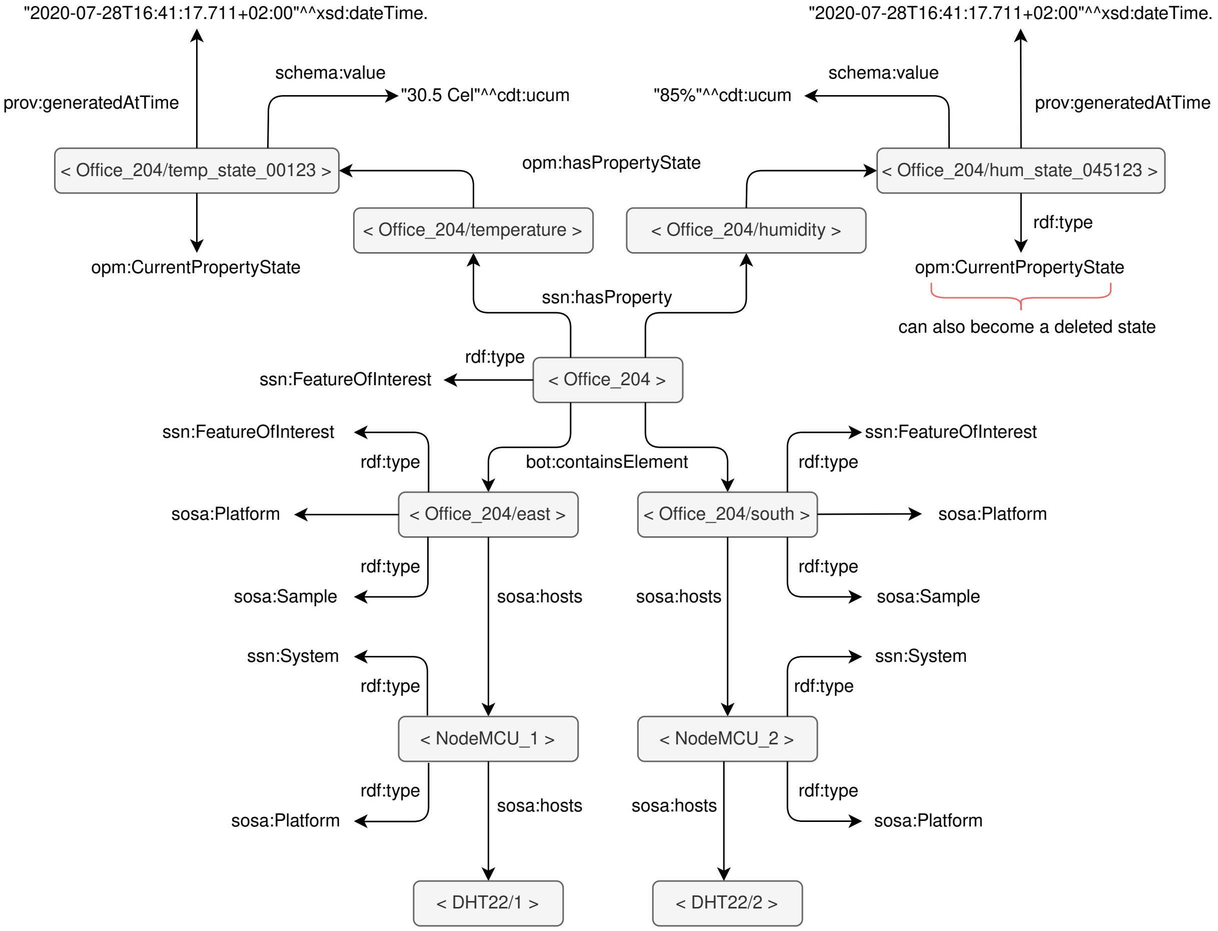


Figure 3.2: Using SSN/SOSA, OPM and SEAS ontologies to extend the semantic details of a building to encapsulate indoor zone information about sensors, elements contained within, their properties and state management (simplistic view).

3.1.3 Data Model Validation, Model Iteration, and Refinement

The potency of a KRL model is tightly bound to the quality of the knowledge graph at its disposal. Factors such as data integrity, consistency, density, noise level, completeness, redundancy, and structural regularities in the knowledge graph can significantly influence the resultant embeddings' quality leading to suboptimal control actions, potentially affecting a building's operational efficiency, occupant comfort, or even safety. For this research, SHACL is adopted to validate the curated knowledge graph against specific quality criteria. In parallel, SPARQL Protocol and RDF Query Language (SPARQL) is adopted for its data extraction and manipulation capabilities, allowing for data cleanup, quality assessment, and transformation operations. Together, SHACL and SPARQL are collaboratively used to mitigate the knowledge graph quality issues delineated below.

1. **Structural Consistency**: Structural irregularities in a knowledge graph can disrupt the learning process and impact the quality of the resulting embeddings. Using SHACL, we can define shape constraints that stipulate the acceptable structure for the nodes. This enforces structural consistency within the knowledge graph. Additionally, we can leverage SPARQL queries to identify and resolve these irregularities by updating or correcting the inconsistent data points.

2. **Data Completeness**: Incomplete data affects the density of the knowledge graph and impedes the effective training of KRL models. SHACL can be employed to define mandatory properties for specific classes in the knowledge graph. This helps ensure that essential data is not missing. Simultaneously, SPARQL can be used to find instances where mandatory properties are not present, enabling targeted data completion efforts.

3. **Redundancy**: Data redundancy in the knowledge graph can complicate the learning process and result in less accurate embeddings. SHACL allows for the definition of unique constraints, ensuring that specific property values are unique across the KG. Meanwhile, SPARQL queries can be constructed to identify duplicate nodes or properties, facilitating the elimination of redundant data.

4. **Data Accuracy**: Incorrect or inaccurate data in the knowledge graph can lead to erroneous learning and flawed decision-making. SHACL's datatype and value constraints can be leveraged to maintain data accuracy. Complementary to this, SPARQL can aid in spotting outliers or erroneous data points.

5. **Class-Property Validity**: The use of inappropriate properties for certain classes can distort the knowledge graph and affect the KRL process. SHACL's property constraints can mandate that specific properties only be used with certain classes. With SPARQL, we can inspect the use of properties across classes to ensure their appropriateness.

6. **Entity Resolution**: Distinguishing similar entities is a common issue in knowledge graphs. SHACL's unique constraints can support the prevention of entity duplication. Concurrently, SPARQL can aid in entity resolution by helping identify and reconcile duplicate entities.

7. **Consistent Property Usage**: Inconsistent usage of properties across entities can cause confusion and impede the learning process. With SHACL, constraints can be defined to

guide consistent property usage. SPARQL queries can be employed to pinpoint instances where property usage may be inconsistent or deviating from the norm.

To illustrate how the data model curated in subsection 3.1.2 is validated, consider the excerpt in listing 3.2 for which we may want to ensure that:

1. Every `sosa:FeatureOfInterest` hosts at least one `ssn:System`.
2. Every `ssn:System` hosts at least one sensor.
3. Every `ssn:Property` has an associated state.
4. Every `opm:CurrentPropertyState` has a defined value (`schema:value`) and a timestamp (`prov:generatedAtTime`).

To check these conditions, validations that are rooted in SPARQL and SHACL are adopted. First, SPARQL `ASK` queries are provided for each condition, which return either true if the condition is violated, or false if the data meets the condition:

1. `sosa:FeatureOfInterest` hosts at least one `ssn:System`:

```
PREFIX ssn: <http://www.w3.org/ns/ssn/>.
PREFIX sosa: <http://www.w3.org/ns/sosa/>.

ASK WHERE {
  ?foi a sosa:FeatureOfInterest .
  FILTER NOT EXISTS {
      ?foi sosa:hosts ?system .
      ?system a ssn:System .
  }
}
```

listing 3.3: Every `sosa:FeatureOfInterest` hosts at least one `ssn:System`

2. Every `ssn:System` hosts at least one sensor:

```
PREFIX ssn: <http://www.w3.org/ns/ssn/>.
PREFIX sosa: <http://www.w3.org/ns/sosa/>.

ASK WHERE {
  ?system a ssn:System .
  FILTER NOT EXISTS {
    ?system sosa:hosts ?sensor
  }
}
```

listing 3.4: Every `ssn:System` hosts at least one sensor

3. Every `ssn:Property` has an associated state:

```
PREFIX ssn: <http://www.w3.org/ns/ssn/>.
PREFIX sosa: <http://www.w3.org/ns/sosa/>.
PREFIX opm: <https://w3id.org/opm#>.

ASK WHERE {
  ?property a ssn:Property .
  FILTER NOT EXISTS {
    ?property opm:hasPropertyState ?state
  }
}
```

listing 3.5: Every ssn:Property has an associated state

4. Every opm:CurrentPropertyState has a defined value and a timestamp:

```
PREFIX opm: <https://w3id.org/opm#>.
PREFIX schema: <http://schema.org/>.
PREFIX prov:  <http://www.w3.org/ns/prov#>.

ASK WHERE {
  ?state a opm:CurrentPropertyState .
  FILTER NOT EXISTS { ?state schema:value ?value }
  FILTER NOT EXISTS { ?state prov:generatedAtTime ?time }
}
```

listing 3.6: Every opm:CurrentPropertyState has a defined value and a timestamp

SHACL is more expressive and capable of defining more complex constraints than SPARQL alone.

```
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd:  <http://www.w3.org/2001/XMLSchema#> .
@prefix schema: <http://schema.org/>.
@prefix sosa: <http://www.w3.org/ns/sosa/> .
@prefix ssn: <http://www.w3.org/ns/ssn/> .
@prefix opm: <https://w3id.org/opm#> .

# Shape for FeatureOfInterest
:FeatureOfInterestShape
    a sh:NodeShape ;
    sh:targetClass sosa:FeatureOfInterest ;
    sh:property [
        sh:path sosa:hosts ;
        sh:class ssn:System ;
        sh:minCount 1 ;
        sh:message "A FeatureOfInterest must host at least one System."
    ] .

# Shape for System
:SystemShape
    a sh:NodeShape ;
    sh:targetClass ssn:System ;
    sh:property [
        sh:path sosa:hosts ;
        sh:minCount 1 ;
        sh:message "A System must host at least one sensor."
    ] .

# Shape for Property
:PropertyShape
    a sh:NodeShape ;
    sh:targetClass ssn:Property ;
    sh:property [
        sh:path opm:hasPropertyState ;
        sh:minCount 1 ;
        sh:message "A Property must have an associated state."
    ] .

# Shape for CurrentPropertyState
:CurrentPropertyStateShape
    a sh:NodeShape ;
    sh:targetClass opm:CurrentPropertyState ;
    sh:property [
        sh:path schema:value ;
        sh:minCount 1 ;
        sh:message "A CurrentPropertyState must have a defined value."
    ] ;
    sh:property [
        sh:path prov:generatedAtTime ;
        sh:datatype xsd:dateTime ;
        sh:minCount 1 ;
        sh:message "A CurrentPropertyState must have a timestamp."
    ] .
```

listing 3.7: SHACL shapes to express the same constraints defined earlier

In the above SHACL shapes graph, each sh:NodeShape defines a shape for a specific class of nodes, e.g., sosa:FeatureOfInterest, ssn:System, ssn:Property, and opm:CurrentPropertyState. For each shape, sh:property is used to specify constraints for properties of the nodes that conform to the shape. For instance, in :FeatureOfInterestShape, the sh:property construct specifies that each sosa:FeatureOfInterest must host (sosa:hosts) at least one (sh:minCount 1) ssn:System. Similarly, :SystemShape specifies that each ssn:System must host at least one sensor, :PropertyShape specifies that each ssn:Property must have an associated state (opm:hasPropertyState), and :CurrentPropertyStateShape specifies that each opm:CurrentPropertyState must have a defined value (schema:value) and a timestamp (prov:generatedAtTime). The sh:message constructs are used to provide human-readable error messages that will be displayed when the constraint is violated.

SHACL validation is then performed using the pySHACL library, a Python library developed by `RDFlib`. The steps outlined below involve the configuration of the validation environment, the loading of RDF data and SHACL shapes, the execution of the validation process, and the analysis of the validation results.

1. **Configuration of `pySHACL` Environment**: The initial step in the validation process is configuring the validation environment. This involves installing and setting up the `pySHACL` library. `PySHACL` can be easily installed using pip (`pip3 install pyshacl`), the package installer for Python.

2. **Loading of RDF data and SHACL Shapes**: Once the `pySHACL` environment is configured, the next step is loading the RDF data and SHACL shapes. The RDF data encapsulates the building data model knowledge graph, and the SHACL shapes define the constraints against which the data is validated. Loading involves reading the RDF and SHACL files into Python objects:

```python
import rdflib
from pyshacl import validate

# Load RDF Data
data_graph = rdflib.Graph()
data_graph.parse("path_to_the_kg_data", format='turtle')

# Load SHACL Shapes
shapes_graph = rdflib.Graph()
shapes_graph.parse("path_to_your_shacl_shapes", format='turtle')
```

listing 3.8: Python script for loading RDF data and SHACL shapes

3. **SHACL Validation Execution and Results Analysis**: With the RDF data and SHACL shapes successfully loaded, the validation process is initiated. This involves comparing the RDF data against the SHACL shapes using the validate function provided by the pySHACL library:

```python
# Run the validation
val = validate(data_graph, shacl_graph=shapes_graph)
conforms, results_graph, results_text = val

# Check if the data passed the SHACL validation
if conforms:
    print("The data graph passed SHACL validation!")
else:
    print("The data graph failed SHACL validation.")
    print(results_text)
```

listing 3.9: Python script for performing the SHACL validation

Upon successful validation execution, the validate function returns a boolean value (`conforms`) indicating whether the RDF data conforms to the SHACL shapes, and a validation report graph (`results_graph`) and text (`results_text`). The validation report provides a detailed account of each violation if any exist. With this analysis, the data model is refined iteratively until the defined competency questions are satisfied.

3.2 Knowledge Representation Learning on Linked Building Data

The overarching aim of this research is to devise a methodology for KRL on BIM-based knowledge graphs that would facilitate more robust, adaptable, and efficient building automation and control systems. This research will concentrate on an assortment of KRL models known for their demonstrated efficacy in processing a wide array of relational datasets. The models delineated below have been selected specifically for their notable ability to reflect the intricacy, richness, and diversity of relationships within the building control domain, and capture diverse types of interactions within BIM-based knowledge graphs.

1. **Graph Neural Networks**: GNNs offer a distinct edge by being explicitly designed for graph-structured data, making them highly compatible with knowledge graph embeddings in the context of building automation and control systems. Specifically, we're looking at the GCN and GAT variants as they excel at capturing both local and global graph structures, which is paramount when representing complex interdependencies of building systems and controls and the environment that they're operating in.

2. **Translational Embedding Models:** These models, such as TransE, are incorporated into our selection for their unique capability to model relationships as translations in the embedding space. This simple yet effective methodology has shown remarkable results in many tasks outside the domain of discourse, making it a compelling candidate for building automation and control.

3. **Bilinear Models**: From this category, we're focusing on DistMult and ComplEx, owing to their expressivity and proficiency at modeling diverse relations, including symmetric, antisymmetric, and reflexive ones. Given that BIM-based knowledge graphs often contain symmetric relations (e.g., the mutual relationship between HVAC and occupancy systems for adaptive control), these models provide substantial value for our investigations.

4. **Tensor Models**: We've included RESCAL for its remarkable expressivity and ability to learn a unique latent representation of entities and relations in a new multi-linear subspace, whose extracted features can provide access to strong collective relational reasoning in downstream building automation tasks.

Similar to section section 3.1, competency questions are framed herein to guide the tensor decomposition method choice-process. These clearly delineate a criterion of essential learning tasks that the method has to satisfy while learning on BIM-based knowledge graph such that a generalizable feature extraction baseline is set for downstream tasks.

Competency Question 3: *Link Prediction: How to discover the existence of unknown relationships or interactions between different entities inside and/or outside of a building's thermal zone such that hidden correlations having an effect on indoor thermal comfort control and the corresponding energy consumption can be explained and exploited for control optimization?*

Because the indoor built environment is very dynamic and stochastic in nature, it is often daunting if not impossible to infer hidden dependencies between two entities in a thermal zone based solely on their individual attributes as is the case with many current building automation systems. These systems generate inefficient insights into the optimization of the cooling load because they rely on sensory data and building information that is scattered around disparate 'Revit families' of the as-built model. Also, during the operation life-cycle stage of a building, control digital twins (as-built models) evolve mainly due to the retrofitting of indoor thermal zones with new materials, sensor additions, removals or alterations in placement, and changes in space utilization requirements. With each new change, new corresponding latent relationships are introduced which necessitates an efficient mechanism for mining them.

Link prediction is an important task in link mining that allows inference about unknown interactions between two entities based not only on their attribute information, but also on the observed existential link information. Having such relational causal explanations for the different phenomena of interest in the controlled indoor environment provides insight into optimization strategies that can be exploited by reinforcement learning control agents. Other relational learning tasks can also be regarded as link prediction problems are shown in the following paragraphs.

Competency Question 4: *Collective Classification: How to use the inherent relational structure of linked building data as a source of valuable information for identification and assignment of related types of information into pre-defined classes?*

Already highlighted in question 3, is the evolvement of relational information as new instances are added to the linked building data model. It is important that the relational learning approach is efficient in exploiting mined latent links between these new instances to compute their relative similarity and infer their target classes based on the pre-defined classes of the known instances they are connected to. For instance, if we need to allocate rooms to those *with* or *without* high cooling load. This can be done using general inference about the heat gain of nearby rooms based on their spatial location relative to the sun's direction and the number of supply air ducts they have. When classes are introduced as entities of the data during relational learning via a new relation rdfs:subClassOf, collective classification is cast as a link prediction problem that infers the probability of occurrence of a relationship `rdfs:subClassOf(i-th entity,j-th class)`.

Competency Question 5: *Entity Resolution: How to use relationships in linked building data as means of propagating deduplication decision information useful to reasonably disambiguate repeated entities in data?*

As the volume and velocity of linked building data grows, so does the duplication of entity definitions across different domain ontologies. For example a sensor entity is addressed differently in both the SSN and SEAS ontologies, and with thousands of such duplications, sensible link prediction inferences necessary for collective classification become a challenge. To alleviate this, authors of LBD ontologies have defined ontological alignments providing an instance matching/record linkage mechanism to break such data disparities. Information necessary for the disambiguation of entities that are assumed to be identical can propagate via the relationships between these entities, and it is therefore important for the relational learning approach to exploit this identity information for collective rather than individual decision making over such identical entities. As in the curation of ontologies, during relational learning, introducing a new relation `owl:sameAs`, and inferring the probability of occurrence of the relationship `owl:sameAs(i-th entity,j-th entity)`, entity resolution can be cast as a link prediction problem.

Competency Question 6: *Link-Based Clustering: How to use relationships in linked building data as a means of propagating entity similarity information useful in grouping entities based on the similarity of both their attributes and relationships?*

A relational approach to modelling building information exposes inherent community structures where entities tend to cluster based on the similarity of not only their attributes but also their relationships. The formulated communities provide holistic insight that is exploitable by self-learning agents and facility managers who need a clear understanding of how the different or related parts of their buildings consume energy. The computational complexity of link-based clustering is hampered by both the size of the graph network and its continuously evolving nature.

Chapter 4

Results and discussion

This chapter presents the experimental results obtained from the performance analysis that is delineated in chapter 3. For an impartial assessment of all tested models during training, a random 10% holdout set of test triples was used. The holdout set it not seen by the models during training or validation. This strategy ensures fair evaluation of each model's generalization capabilities to new, unseen data. To lay a solid framework for repeatability in future research experiments, throughout the analysis, a consistent set of training setup choices and hyper-parameters was maintained, as detailed in table 4.1.

4.1 A study on training setup choices

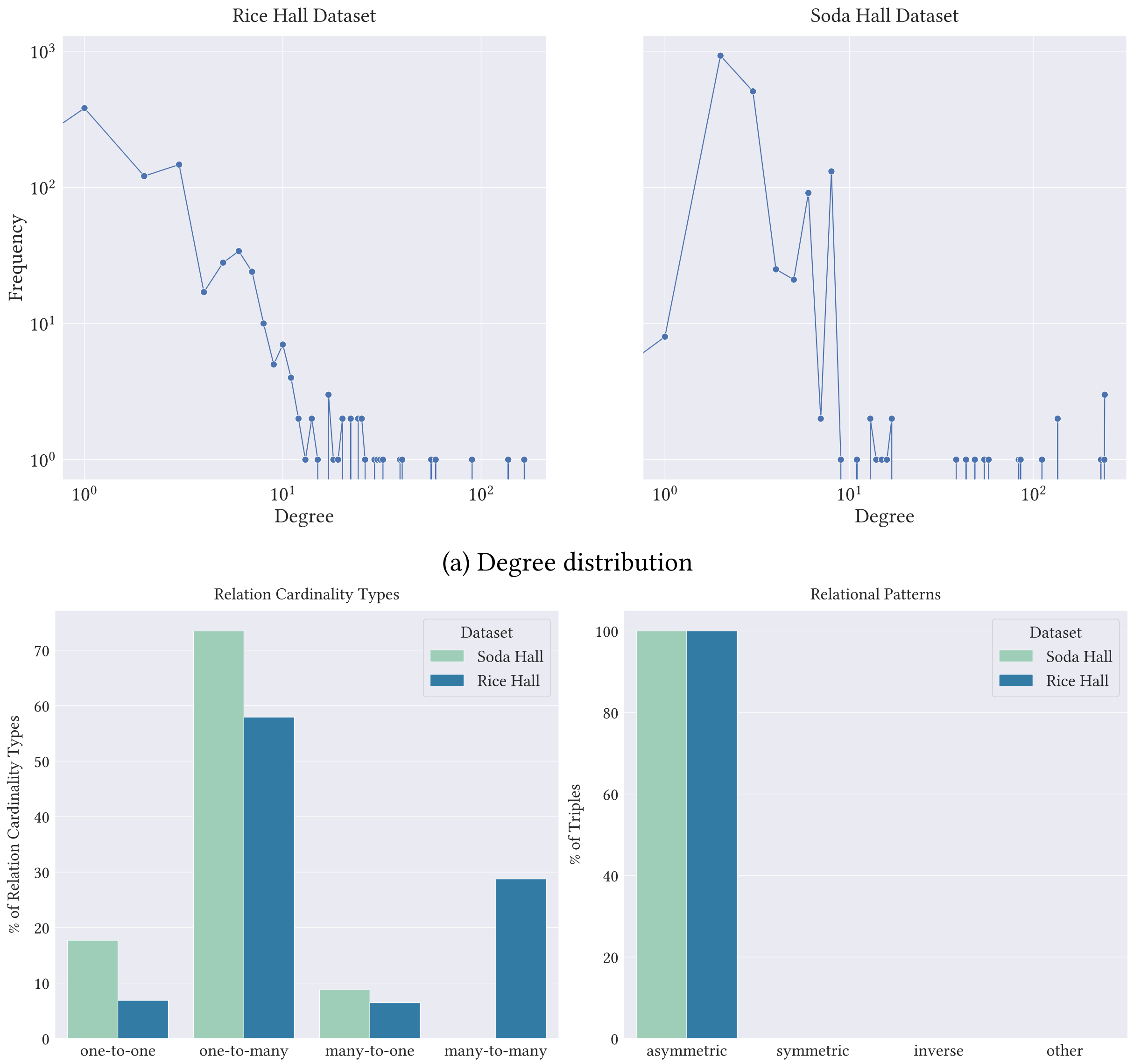
The initial series of experiments aims to evaluate the influence of different categorical decisions related to the training configuration of KRL models. In particular, this method varies the optimizer (selecting from a choice of Adam, AdaGrad and Stochastic Gradient Descent (SGD)), training objective function (selecting from a choice of Binary Cross-Entropy Loss (BCEL), Softplus Loss (SPL), Margin Ranking Loss (MRL), and the self-adversarial loss (NSSA)), and finally considering the exclusion or inclusion of inverse relationships in the graph, a process that involves adding a copy of each triple during training but with an inverse relation. A summary of the above categorical choices is presented in table 4.3.

Table 4.1: Default Training Setup Choices and Hyperparameters

Parameter	**Value By Approach**				
	ComplEx	DistMult	RotatE	TransE	TransH
Embedding Dim	50	50	200	50	50
Num Epochs	500				
Learning Rate	0.02				
Num Negatives	1				
Optimizer	Adagrad				
Inverse Relations	False				
Loss Function	Margin Ranking Loss (Margin 1.0)				

Dataset	Density	Entity Heterogeneity	Average Degree
Rice Hall	0.002	65	3.664
Soda Hall	0.001	36	4.342

Table 4.2: Training dataset properties and structural patterns



(a) Degree distribution

(b) Relation cardinality types and relation patterns

Figure 4.1: Training dataset degree distributions, relation cardinality types and relation patterns

Figure 4.2 depicts the Hits@10 scores distribution on the test set for different training setup options for all models and datasets. It offers detailed insight into how the models react to variations in the training setup. In the case of the Rice Hall dataset, all models show a comparable performance range, with TransE and RotatE standing out as the highest performers, whereas DistMult and ComplexE consistently show subpar performance no matter the training setup, a trend that can also be seen on the Soda Hall dataset, albeit happening more aggressively. DistMult inherently assumes symmetric relations due to its

Table 4.3: Training Setup Choice Matrix

Model	Optimizer	Loss Function	Inverse Relationship
All models	Adagrad	Binary Cross Entropy Loss (BCEL)	False
			True
		Softplus Loss (SPL)	False
			True
		Margin Ranking Loss (MRL)	False
			True
	Adam	Binary Cross Entropy Loss (BCEL)	False
			True
		Softplus Loss (SPL)	False
			True
		Margin Ranking (MRL)	False
			True
	SGD	Binary Cross Entropy Loss (BCEL)	False
			True
		Softplus Loss (SPL)	False
			True
		Margin Ranking Loss (MRL)	False
			True

bilinear scoring function. However, many relationships in both datasets, such as `feeds` or `isPartOf`, are inherently directional (asymmetric). ComplEx attempts to address this by extending DistMult to complex numbers, allowing it to capture both symmetric and asymmetric relations. Despite this, the real challenge lies in the nuanced, hierarchical, and interdependent relationships typical of BIM-based knowledge graphs, which may not be fully captured by the algebraic nature of ComplEx due to its inability to infer composition patterns. Composition patterns allow a building's multi-faceted relationships to be represented in knowledge graph embeddings. For example, the temperature in a room might be influenced

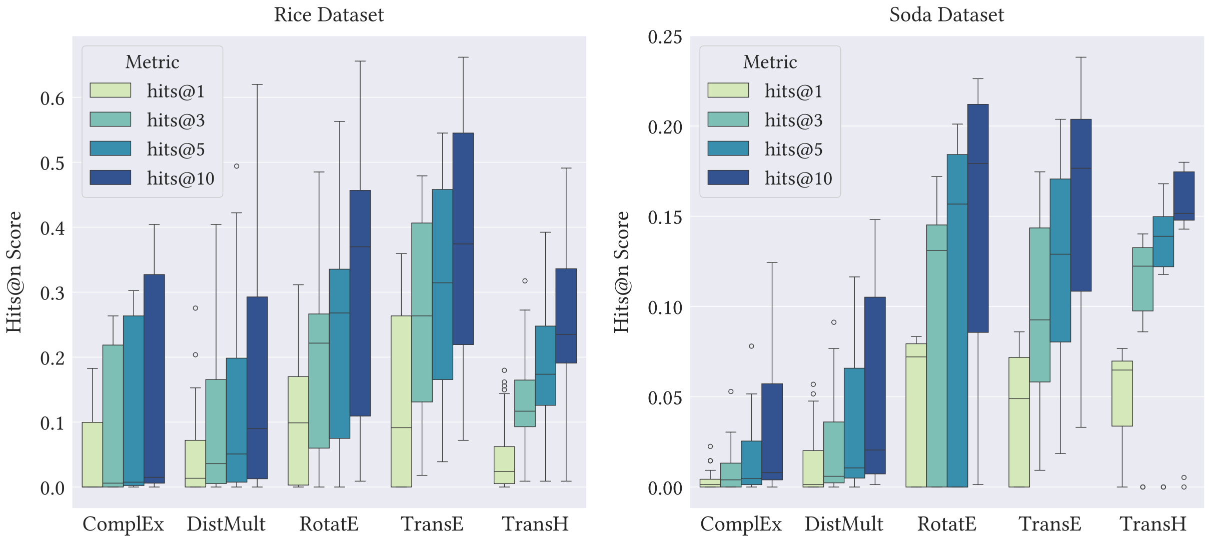
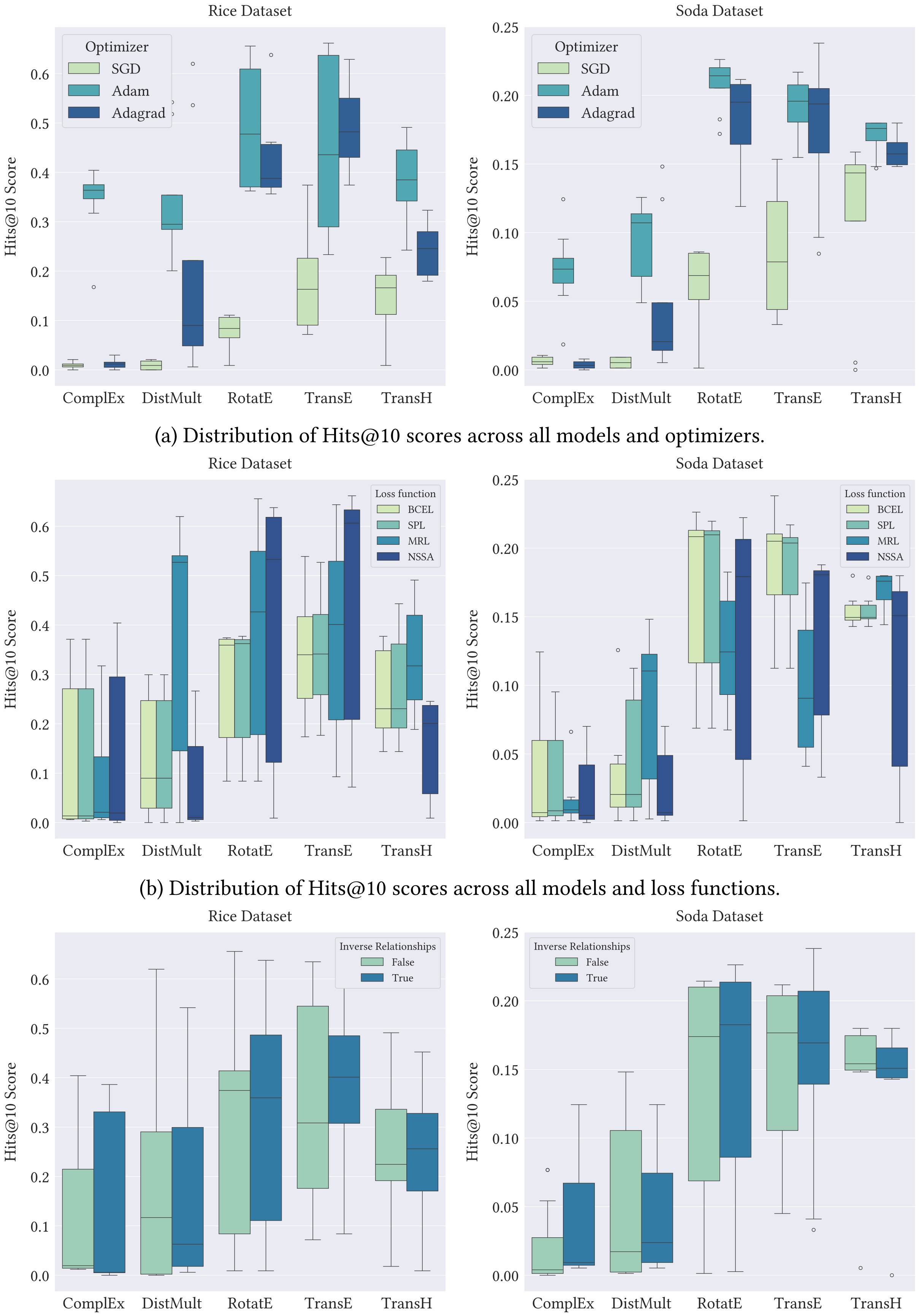


Figure 4.2: Distribution of Hits@n (where $n \in \{1, 3, 5, 10\}$) scores across all categorial training setup choices.



(a) Distribution of Hits@10 scores across all models and optimizers.

(b) Distribution of Hits@10 scores across all models and loss functions.

(c) Distribution of Hits@10 scores across all models with inverse relations present or absent.

Figure 4.3: The effect of different training setup choices across all models and both datasets.

by the operation of HVAC systems, the number of inhabiting occupants, time of day, and even external weather conditions. Expressive capture of such patterns can enable an automation agent to predict the impact of adjusting the configurations of one system (like the HVAC settings) on various related metrics (such as energy consumption or occupant comfort). Also, buildings often have a hierarchical structure: composed of floors, floors are composed of rooms, and rooms can contain various sensors and actuators. Composition patterns in embeddings can reflect this hierarchy, allowing automation agents to aggregate or disaggregate information at different levels. For instance, understanding the aggregated energy use at the building level while also being able to drill down into specific floors or rooms. It is also important to note that BIM-based knowledge graphs are often characterized by sparse data, with many potential but unobserved relationships between entities. This sparsity challenges models to generalize well from observed to unobserved relationships. TransE and RotatE are potentially less susceptible to overfitting in sparse environments because they embody lower model complexity through their geometric operations—translations and rotations, respectively. These operations inherently require fewer parameters: TransE needs a single vector to represent each relation as a translation in the embedding space, while RotatE requires a complex number to represent each relation as a rotation. The lower number of parameters reduces the models' capacity to fit noise, a common pitfall in sparse datasets where the signal-to-noise ratio [1] can be low. DistMult and ComplEx can capture complex patterns and interactions through their use of dot products and complex space embeddings, respectively. However, this increased expressiveness comes at the cost of higher susceptibility to overfitting when data are sparse. Perhaps the most striking observation is that RotatE demonstrates superior performance across both datasets, however, as seen in the Rice Hall dataset, older methods, such as TransE, can outperform it if given an optimized training setup.

In order to delve deeper into the effects of various training configurations, other distributions of model performance (Hits@10) for both datasets are presented in figure 4.3a through figure 4.3c, revealing some intriguing patterns based on the different choices made. Figure 4.3a indicates that setups utilizing the Adam optimizer consistently outperform those using Adagrad and SGD. Adam's adaptive learning rate mechanism and momentum updates likely contribute to its ability to converge faster and escape local minima more effectively. Adagrad also adopts adaptive learning rates, performing smaller updates for parameters associated with frequently occurring features, and larger updates for parameters associated with infrequent features. Adam and Adagrad's adaptive learning rate mechanisms make them particularly well suited for tasks with sparse data or highly variable feature importance. However, the monotonic decreasing learning rate of Adagrad can pose challenges in certain scenarios. As the algorithm accumulates squared gradients over time, the learning rates for all parameters continuously decrease. While this ensures stable and well-scaled updates, it may also cause the algorithm to prematurely and excessively reduce the learning rate. The poor performance of SGD indicates that its simplistic updating mechanism faces challenges

[1] Signal-to-noise ratio is defined as the ratio of the power of a signal (meaningful input) to the power of background noise (meaningless or unwanted input)

in effectively exploring the complex parameter space of KRL models. Also SGD does not incorporate adaptive learning rates which means that it treats all parameters equally, applying the same update magnitude across the board. This uniform approach does not account for individual characteristics or the importance of different features within the data. In scenarios where certain features are more sparse or informative than others, a uniform update size can lead to suboptimal convergence rates. Parameters associated with infrequent features may not be adjusted adequately, potentially causing the model to miss out on learning from these critical but rare data points. Its important to note that, due to the *no-free-lunch*[2] theorem, there is no one-size-fits-all optimizer; in reality, an optimizer's efficiency is highly reliant on the training setup and unique characteristics of the underlying dataset. This is evident in figure 4.3a (Soda Hall dataset), where for ComplEx, Adagrad performs worse than SGD.

[2] There is no single optimizer to that will always do better than any other optimizer

Chapter 5

Conclusion

This thesis has examined several aspects of KRL with respect to building automation and control, which will be summarized in this chapter while discussing interesting directions for future research.

5.1 Summary

In this thesis, we proposed a framework for building control based on KRL and discussed its properties, applications, and evaluation criteria. This framework presents the mechanics to capture both local and global dependencies in relational building data to facilitate contextual reasoning in building automation agents. The models encapsulated in the framework do not require deep domain knowledge to function, which facilitates the learning process significantly. At the framework's core is a 4-step process that involves data preprocessing, data splitting, feature engineering, and model selection and this thesis has recommended the following strategies for each of these steps.

1. Data Preprocessing

    - **Duplicate and inconsistency removal**: This thesis discussed how it is not uncommon for semantically interlinked building data to have a lot of duplicates and inconsistencies. These mainly arise from automated knowledge graph curation, knowledge graph stitching, or just data extraction errors. Eliminating duplicates helped reduce data unnecessary noise while improving the integrity of the data representations being learnt from. Deduplication is very complicated, especially when carried out on very large datasets. It is on this basis that this research recommends adopting schema alignment at the earliest stage of knowledge graph curation together with SHACL rules that validate and ensure that the knowledge graph has no duplicates. Degree-based and frequency-based entity and relation pruning techniques were also found to be efficient at removing irrelevant entities and relations that can affect the downstream KRL process.
    - **Data Transformation**: Different learning models in the framework require a knowledge graph to be transformed into a suitable format. Four representations

were investigated including vector-based, matrix-based, graph-based, and tensor-based which was found to be the most effective for modeling high-order interactions and capturing the inherent dependencies between nodes in a graph. Being able to model complex relationships is key for efficient message passing and propagation to exploit both local and distant information in a knowledge graph.

2. Data Splitting

   In this framework, it was deemend appropriate to use a *non-cold start* train/test split strategy such that the test set contains only entities and relations that are seen during training. This approach provides an accurate estimate of a model's performance on data that it is likely to encounter in practice however, this strategy may not capture the model's ability to generalize to new entities or relations that were not observed during training. Cold-start splitting can lead to better generalization on unseen data but it can be challenging to obtain reliable performance estimates with limited test data, which is usually the case in the context of building control and automation. Buildings are complex systems with numerous interacting components, and it is challenging to accurately simulate all of the potential scenarios and interactions that may occur in real-world environments. Using a non-cold start approach that incorporates all entities and relations in the knowledge graph in both the training and test sets increases the likelihood of a more realistic estimation. Domain-based splitting is another strategy that is useful in evaluating a model's ability to transfer knowledge across domains. At the core, this approach ensures that the training set contains data from one domain and the test set contains data from another domain. However, in the context of building control and reasoning, it can be challenging to identify distinct domain boundaries between the training and test sets. As buildings are highly interconnected systems, it may not be possible to divide entities and relationships into distinct domains.

3. Feature Engineering, Model Selection, Hyper-Parameter Tuning and Evaluation

   This research relies on embedding entities and relationships into a high-dimensional vector space to extract relevant inference features from a knowledge graph. The findings show that GNNs, bilinear and tensor models are good at learning embeddings that capture both local and distant information in a knowledge graph, a mechanism that allows exploiting both local and global dependencies in knowledge graphs. However, these models require large amounts of data to jump out of their cold starts and are resource intensive when trained on large distributed graphs such as those found in smart-city applications. They are also difficult to interpret and if used naively, they can produce misleading inferences. On the other hand, translational embedding models were simpler to interpret and they showed better computational efficiency evident by their ability to learn meaningful representations with relatively small amounts of data however, they struggled with sparse knowledge graphs. Generally, this research concludes that choosing the right model requires a careful trade-off analysis mainly between explainability and computational efficiency. Compromises have to be made depending on the application at hand. The framework herein favors computational efficiency over explainability which makes sense in a building automation use case because the density of sensor networks and actuators in buildings is ever-growing and so is their generated data. As discussed in 4, a known peculiarity of real-life building control knowledge graphs is that they only contain positive real facts (positive examples for model training), and false facts (negative examples) have to be curated under the Local Closed World Assumption (CWA). This thesis showed that there is no one-size-fits-all answer to which negative sampling strategy is most effective. Downstream inference tasks, computational resources as well as time-constraints dictate the choice of method. Regarding hyperparameter tuning, grid search was found to be straightforward and efficient with small search spaces but as a knowledge graph grows, so do the resources required to satisfy the search. Early stopping was explored to speed up the grid search, however, this led to premature convergence on some occasions, especially if the hyperparameters were sensitive to the learning rate or the number of epochs. Combining Quasi-random search and Bayesian optimization is more sophisticated but significantly reduces the number of hyperparameter samples needed to obtain acceptable results. To effectively evaluate the framework's models, this research casts KRL-based building control as a *learning to rank* problem and uses a combination of MR, MRR and Hits@n to assess how well positive facts are ranked against synthetic negatives built under the Local CWA.

5.2 Outlook

The KRL-based Building Control Framework has shown good results both in its ability to leverage the inherent relational structure of BIM-based knowledge graphs and in its scalability with the evergrowing complexity of building automation systems. This section will briefly outline interesting directions for future research that can improve the framework's capabilities, explainability, and computational efficiency within the building automation domain. This section will also delineate how the framework can be applied to other BIM domains such as costing and evacuation.

5.2.1 Enforcing onset SHACL validations and schema conformity

Instead of attempting to address duplicates and inconsistencies later on in the KRL pipeline, using SHACL restrictions at the outset of BIM knowledge graph curation might be a proactive method to ensure the uniqueness and consistency of the knowledge graph from the beginning. Duplicates in the knowledge graph substantially increase the computational complexity of the KRL Pipeline. For instance, if multiple entities in the knowledge graph represent the same physical object in the building, this can result in redundant computations and a model that is both larger and more complicated. Likewise, if the relationships between entities are inconsistent or contradictory, this can result in low precision and efficacy of the learned representations. In addition to onset SHACL validations, starting with a clear and consistent schema that specifies the kinds of entities and relationships that will be included in the knowledge graph can improve the quality of the training data used to learn representations resulting in more accurate and effective models. In its present form, the research's framework does not explicitly account for the erroneous nature of many BIM-based knowledge graphs, which can potentially diminish the computational efficiency of the KRL pipeline and the accuracy of the learnt representations. An interesting extension of the framework could therefore be the inclusion of the mechanics for enforcing schema conformity and deduplication SHACL constraints for curated knowledge graphs.

5.2.2 Learning from multi-modal BIM-based knowlegde graphs

Multi-modal knowledge graphs have the capacity to represent different types of building information, such as structural, thermal, electrical, and spatial data, which are usually of different formats and frequently maintained in separate data sources. Integrating these modalities into a single knowledge graph can provide a more comprehensive understanding of a building, allowing for more sophisticated reasoning by building control agents. Learning from multi-modal knowledge graphs poses new challenges for KRL. First, the embeddings must be capable of capturing the interactions between multiple modalities, which may necessitate the development of new embedding methods or fusion techniques. Second, different modalities may have varying degrees of sparsity or noise, necessitating the use of specialized pre-processing or regularisation procedures. Third, when integrating multiple modalities, the knowledge graph may become very large, necessitating the use of scalable learning algorithms or distributed computing platforms. Despite these challenges, learning from multi-modal BIM-based knowledge graphs is a promising direction for future research and has the potential to avail unprecedented optimization opportunities within the boundaries of building automation and other BIM domains.

5.2.3 Explainability improvements

The ability to comprehend and interpret the outcomes of the framework's results is referred to as ”explainability”. It may not always be obvious how the learnt representations were derived or what factors influenced them. To overcome this constraint, future research may focus on methods for visualizing and interpreting the learned embeddings. This could entail producing heatmaps or saliency maps that draw attention to the most crucial features or relationships within the embeddings, making it simpler for building automation specialists to comprehend the framework's decision-making processes. Including rule-based systems or logical reasoning in the framework is another strategy for enhancing explainability. By introducing explicitly defined rules into the framework, it would be possible to make the decision-making process more transparent and interpretable as the reasoning behind the decisions can be traced back to the specific rules being used. This would help building automation specialists better understand and trust the decisions being made in instances where the framework's choices directly affect the physical environment and the occupants in it. Some example rules are provided together with the pseudocode for a hypothetical algorithm incorporating them into the research framework. The algorithm takes as input a Knowledge Graph G and outputs predictions for building control and automation tasks. It first initializes a KRL model M and a set of logical rules R. The model is trained and is then used to make predictions for each building control task. These predictions are checked against the logical rules to ensure that they are within acceptable bounds. Finally, the algorithm performs the appropriate action based on the predictions. Being able to exploit these kinds of rules in KRL could therefore be a very valuable extension to the research framework developed herein.

1. If the outside temperature is above 25℃ and the CO_2 levels inside the building exceed 1000 ppm, then increase the airflow to improve indoor air quality.
   $T_{out} > 25 \wedge CO_2 > 1000 \Rightarrow Airflow_{in} + +$

2. If the occupancy level in a room is zero, then turn off all lights and HVAC systems to conserve energy.
   $Occupancy_{room} = 0 \Rightarrow Lights_{room} = 0 \wedge HVAC_{room} = 0$

3. If the temperature in a room is below the setpoint and the blinds are closed, then open the blinds to allow for passive solar heating.
   $T_{room} < Setpoint \wedge Blinds_{room} = closed \Rightarrow Blinds_{room} = open$

4. If the temperature in a room is above the setpoint and the occupancy level is high, then increase the airflow to improve thermal comfort.
   $T_{room} > Setpoint \wedge Occupancy_{room} = high \Rightarrow Airflow_{room} + +$

Algorithm 1 A hypothetical integration of logical rules with the framework

```
Input: Knowledge Graph $G$
Output: Predictions for building control and automation
Initialize the KRL model $M$
Initialize the logical rules $R$
Train the KRL model
for each building control task do
    Use the KRL model $M$ to make predictions based on the current state of $G$
    Use the logical rules $R$ to impose constraints and ensure predictions are within bounds
    Perform action based on predictions
end for
```

5.2.4 A need for agreed-upon fair evaluation protocols and novel datasets

A major obstacle to the development and assessment of KRL-based building control systems in the AEC/FM domain is the absence of agreed-upon evaluation protocols and benchmark datasets. To address this issue, it is essential to develop fair and objective evaluation protocols for comparing the performance of various KRL-based building control systems. These protocols should take into account a variety of metrics such as accuracy, efficiency, scalability, and interpretability. Similarly, creating new benchmark datasets that reflect real-world building control scenarios can aid in evaluating the performance of KRL-based building control systems. These datasets should contain a wide range of diverse and representative instances of building control tasks, such as temperature and lighting control, occupancy detection, and energy management. Furthermore, it is crucial to make sure that the benchmark datasets are available and open to the research community.

5.2.5 Security issues

KRL models are sensitive to a variety of security vulnerabilities, including adversarial alterations, which can have serious implications for building safety and efficiency. Adversarial examples are inputs to machine learning models that are deliberately designed to induce inaccurate predictions or classifications. Due to the fact that KRL models rely on accurately representing and comprehending complex data patterns, they are susceptible to minor perturbations in the input data that can result in substantial alterations to the output. An attacker, for example, could try to change sensor readings or other inputs to the KRL model, forcing it to make inaccurate or dangerous conclusions about building control and automation. To address these security concerns, several defense mechanisms can be developed such as robust training methods, input preprocessing approaches, and model-based defenses. In the case of input preprocessing approaches, an IoT device may encrypt data before delivering it to the KRL model, preventing attackers from intercepting and altering the data. Furthermore, the KRL model could be trained using a mix of clean and adversarial data to improve its resistance to attacks. These are all open questions that can be explored to extend the framework presented herein.

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.

Akompab, D. A., Bi, P., Williams, S., Grant, J., Walker, I. A., and Augoustinos, M. (2013). Awareness of and attitudes towards heat waves within the context of climate change among a cohort of residents in adelaide, australia. *International Journal of Environmental Research and Public Health*, 10(1).

Alam, M., Sanjayan, J., and Zou, P. X. (2019). Balancing energy efficiency and heat wave resilience in building design. In *Climate Adaptation Engineering: Risks and Economics for Infrastructure Decision-Making*.

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

Anderson, A., Marsters, A., Dossick, C. S., and Neff, G. (2012). Construction to operations exchange: Challenges of implementing COBie and BIM in a large owner organization. In *Construction Research Congress 2012: Construction Challenges in a Flat World, Proceedings of the 2012 Construction Research Congress*.

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.

Asadi, E., Da Silva, M. G., Antunes, C. H., and Dias, L. (2012). Multi-objective optimization for building retrofit strategies: A model and an application. *Energy and Buildings*, 44(1).

ASHRAE, G. (2016). Guideline 10-2016-Interactions Affecting the Achievement of Acceptable 1 Indoor Environments. *American Society of Heating, Refrigerating and Air-Conditioning*, 2.

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.

Baniassadi, A., Heusinger, J., and Sailor, D. J. (2018). Energy efficiency vs resiliency to extreme heat and power outages: The role of evolving building energy codes. *Building and Environment*, 139.

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.

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8).

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.

Chen, K. W., Janssen, P., and Schlueter, A. (2018a). Multi-objective optimisation of building form, envelope and cooling system for improved building energy performance. *Automation in Construction*, 94:449–457.

Chen, X., Wei, T., Chen, X., Li, X., and Zhu, Q. (2018b). 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. (2018c). 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. *ACM*, (May):1–6.

Delgarm, N., Sajadi, B., Kowsary, F., and Delgarm, S. (2016). Multi-objective optimization of the building energy performance: A simulation-based approach by means of particle swarm optimization (PSO). *Applied Energy*, 170:293–303.

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, Z. (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.

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.

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.

Ginestet, C. (2010). Introduction to Statistical Relational Learning. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 173(4).

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.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Hopke, J. E. (2020). Connecting Extreme Heat Events to Climate Change: Media Coverage of Heat Waves and Wildfires. *Environmental Communication*, 14(4).

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.

Junk, J., Goergen, K., and Krein, A. (2019). Future heat waves in different european capitals based on climate change indicators. *International Journal of Environmental Research and Public Health*, 16(20).

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.

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

Kriebel-Gasparro, A. (2022). Climate Change: Effects on the Older Adult. *Journal for Nurse Practitioners*, 18(4).

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. *CIBSE Technical Symposium*, (April).

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.

Kumar, V. and Teo, A. L. E. (2021a). Development of a rule-based system to enhance the data consistency and usability of COBie datasheets. *Journal of Computational Design and Engineering*, 8(1).

Kumar, V. and Teo, E. A. L. E. (2021b). Exploring the application of property graph model in visualizing COBie data. *Journal of Facilities Management*, 19(4).

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.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11).

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

Lin, Y., Han, X., Xie, R., Liu, Z., and Sun, M. (2018). Knowledge Representation Learning: A Quantitative Review.

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.

Liu, Z., Sun, M., Lin, Y., and Xie, R. (2016). Knowledge representation learning: A review. *Jisuanji Yanjiu yu Fazhan/Computer Research and Development*, 53(2):247–261.

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*, (March):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.

Merlet, Y., Rouchier, S., Jay, A., Cellier, N., and Woloszyn, M. (2022). Integration of phasing on multi-objective optimization of building stock energy retrofit. *Energy and Buildings*, 257.

Miller, S., Chua, K., Coggins, J., and Mohtadi, H. (2021). Heat waves, climate change, and economic output.

Mitchell, D., Heaviside, C., Vardoulakis, S., Huntingford, C., Masato, G., P Guillod, B., Frumhoff, P., Bowery, A., Wallom, D., and Allen, M. (2016). Attributing human mortality during extreme heat waves to anthropogenic climate change. *Environmental Research Letters*, 11(7).

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.

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. (2016a). A review of relational machine learning for knowledge graphs.

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

Nickel, M., Tresp, V., and Kriegel, H.-P. (2011). A Three-Way Model for Collective Learning on Multi-Relational Data. Technical report.

Nickel, M., Tresp, V., and Kriegel, H.-P. (2012). Factorizing YAGO.

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.

O'Brien, W., Tahmasebi, F., Andersen, R. K., Azar, E., Barthelmes, V., Belafi, Z. D., Berger, C., Chen, D., De Simone, M., Simona d'Oca, Hong, T., Jin, Q., Khovalyg, D., Lamberts, R., Novakovic, V., Park, J. Y., Plagmann, M., Rajus, V. S., Vellei, M., Verbruggen, S., Wagner, A., Willems, E., Yan, D., and Zhou, J. (2020). An international review of occupant-related aspects of building energy codes and standards. *Building and Environment*, 179.

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.

Park, J. Y., Dougherty, T., Fritz, H., and Nagy, Z. (2019a). LightLearn: An adaptive and occupant centered controller for lighting based on reinforcement learning. *Building and Environment*, 147.

Park, J. Y., Mistur, E., Kim, D., Mo, Y., and Hoefer, R. (2022). Toward human-centric urban infrastructure: Text mining for social media data to identify the public perception of COVID-19 policy in transportation hubs. *Sustainable Cities and Society*, 76.

Park, J. Y. and Nagy, Z. (2018). Comprehensive analysis of the relationship between thermal comfort and building control research - A data-driven literature review.

Park, J. Y., Ouf, M. M., Gunay, B., Peng, Y., O'Brien, W., Kjærgaard, M. B., and Nagy, Z. (2019b). A critical review of field implementations of occupant-centric building controls.

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

Pauwels, P., Costin, A., and Rasmussen, M. H. (2022). Knowledge Graphs and Linked Data for the Built Environment. In *Structural Integrity*, volume 20.

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.

Peng, R. D., Bobb, J. F., Tebaldi, C., McDaniel, L., Bell, M. L., and Dominici, F. (2011). Toward a quantitative estimate of future heat wave mortality under global climate change. *Environmental Health Perspectives*, 119(5).

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., Lefrançois, M., Schneider, G. F., and Pauwels, P. (2019a). BOT: the Building Topology Ontology of the W3C Linked Building Data Group. *Semantic Web Journal*, 0(0).

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. (2019b). 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.

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., Berg, R. v. d., 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.

Shaikh, P. H., Nor, N. B. M., Nallagownden, P., and Elamvazuthi, I. (2018). Intelligent multi-objective optimization for building energy and comfort management. *Journal of King Saud University - Engineering Sciences*, 30(2):195–204.

Singh, A. P. and Gordon, G. J. (2008). Relational learning via collective matrix factorization. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 650–658.

Somerville, R., Trenberth, K., Meehl, J., and Masters, J. (2012). Heat Waves and Climate Change. *Climate Communications Science & OUtrecha.*

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.

Teicholz, P. (2018). *BIM for facility managers.*

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.

Toffolo, A. and Lazzaretto, A. (2002). Evolutionary algorithms for multi-objective energetic and economic optimization in thermal system design. *Energy*, 27(6):549–567.

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.

Viguié, V., Lemonsu, A., Hallegatte, S., Beaulant, A. L., Marchadier, C., Masson, V., Pigeon, G., and Salagnac, J. L. (2020). Early adaptation to heat waves and future reduction of air-conditioning energy use in Paris. *Environmental Research Letters*, 15(7).

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.

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.

Wijeratne, W. M. U., Samarasinghalage, T. I., Yang, R. J., and Wakefield, R. (2022). Multi-objective optimisation for building integrated photovoltaics (BIPV) roof projects in early design phase. *Applied Energy*, 309.

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.

William East, E., Nisbet, N., and Liebich, T. (2013). Facility Management Handover Model View. *Journal of Computing in Civil Engineering*, 27(1).

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.

Yi, H. C., You, Z. H., Huang, D. S., and Kwoh, C. K. (2022). Graph representation learning in bioinformatics: Trends, methods and applications.

Yong, Z., Li-juan, Y., Qian, Z., and Xiao-yan, S. (2020). Multi-objective optimization of building energy performance using a particle swarm optimizer with less control parameters. *Journal of Building Engineering*, 32.

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.

Zhou, X., Carmeliet, J., Sulzer, M., and Derome, D. (2020). Energy-efficient mitigation measures for improving indoor thermal comfort during heat waves. *Applied Energy*, 278.