School of Design and the Built Environment

Improving data management through automatic information extraction model in ontology for road asset management

Xiang Lei
0000-0003-2838-984X

This thesis is presented for the Degree of
Doctor of Philosophy
of
Curtin University

December 2023

Declaration

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgment has been made.

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

(Include where applicable)
Human Ethics (For projects involving human participants/tissue, etc) The research presented and reported in this thesis was conducted in accordance with the National Health and Medical Research Council National Statement on Ethical Conduct in Human Research (2007) – updated March 2014. The proposed research study received human research ethics approval from the Curtin University Human Research Ethics Committee (EC00262), Approval Number #HRE2021-0639

Signature: Xiang Lei

Date: 24/12/2023

Acknowledgements

I would like to express my heartfelt gratitude to all those who have contributed to the completion of this thesis. First and foremost, I would like to thank my main supervisor, Professor Peng Wu, for his invaluable guidance and wise advice throughout my research journey. His constant encouragement has been instrumental in overcoming any obstacles I encountered.

I am also incredibly fortunate to have had Dr. Junxiang Zhu and Dr. Wenchi shou as my co-supervisors. Their valuable suggestions and mentorship have deepened my understanding of research and helped me grow as a researcher. Especially Dr. Junxiang, he dedicated significant time and effort to meticulously review and revise every academic paper I drafted, and his support was indispensable in getting these papers published.

I am immensely grateful for the opportunity to work alongside my research colleagues in Curtin University: Dr. Chengke Wu, Dr. Rui Jiang, Dr. Shuyuan Xu, Dr. Yongze Song, and Dr Heap-Yih Chong have been instrumental in this study. I extend special thanks to Dr Jun Wang for his technical advice on hybrid deep learning model training and ontology development.

Most data used in this research were collected from a public online database. However, it is still important to collaborate with the industry to conduct the focus group. Thus, I must thank all participants who provided their valuable knowledge, despite that I cannot mention their identities due to the ethic issue.

The financial support provided by the Australian Research Council discovery project (grant number P180104026.) has been crucial in the realization of this study, and I am sincerely grateful for their funding. I am also indebted to the colleagues in the School of Design and Built Environment, thanks for assisting me with my study and my life at Curtin.

Finally, never to be forgotten are my warmest family and the dearest Mrs Rui Jiang. I fall short of words to properly express my gratitude for their love and support. I wish to proudly dedicate this piece of work to them all.

Abstract

Roads are a critical component of transportation infrastructure, and their effective maintenance is paramount in ensuring their continued functionality and safety. Road asset management (RAM) encompasses various activities, including inspection, condition evaluation, decision-making, and rehabilitation. In the digital age, information technology plays a vital role in streamlining bridge maintenance procedures. Notably, the success of RAM, a significant part of maintenance, significantly influences the overall success of road maintenance programmes. However, managing bridge rehabilitation projects is a complex undertaking owing to intricate constraints, including various materials, equipment, approvals, and design drawings involved. Road agencies, such as Main Roads Western Australia, have historically formulated strategic maintenance plans for their road networks primarily considering pavement condition and maintenance budgets. However, considering the growing importance of sustainability, developing pavement maintenance plans that aim for the highest sustainable benefits have become crucial.

Modern computer-based management methodologies, such as the recently developed ontology, excel in handling constraints within intricate projects, such as road construction. It was originally rooted in philosophy, and then redefined in computer science and information science as a framework describing the types, structures, objects, properties, processes, and relations. With regard to information integration, relational databases are the predominant tools utilised. However, they are limited in their ability to effectively model the interconnections among constraint entities, especially entity–relation–entity triples. In contrast, graph databases, such as ontologies, excel at managing unstructured information, while supporting essential management functions grounded in logical reasoning. Nevertheless, current ontologies face syntax limitations, rendering them unable to support complex computations and updates, including enumeration, iteration, and temporal computations. This study makes several significant contributions to the existing body of literature in this context. On a theoretical level, it delivers a comprehensive elucidation of the fundamental concepts that underlie ontologies, resource description frameworks (RDF), and labelled property graphs (LPGs). Ontologies function as semantic data models that delineate the types of entities within a given domain and the properties employed for their description. Conversely, RDF is focused on crafting an optimised data schema and description logic, with the goal of striking a balance amongst expressiveness, computational efficiency, and reasoning soundness. LPGs, which exhibit a closer alignment with graph theory, emphasise the capture of relationships among entities, facilitating efficient information retrieval and reasoning.

Environmental impact assessment (EIA) has evolved into a crucial component of RAM, particularly in the context of routine monitoring and major construction projects. However, EIA-related knowledge and information are often fragmented, and conventional management methods are deemed inefficient in collecting and providing project managers with access to such dispersed information. Ontology contributes to enhancing data management efficiency

and can serve as a platform for stakeholders with diverse backgrounds to share their knowledge and better comprehend asset management concepts. Moreover, the majority of knowledge bases (KBs) are inherently incomplete, thereby impacting the completeness of ontology, which can be measured by the time spent searching for project information and the usefulness of the information retrieved (e.g., its accuracy). However, previous studies have fallen short of establishing a compact and best-practice solution, and RDF-based ontologies are encumbered by limitations, including the capacity to handle large volumes of data and visual performance. In response to these challenges, this study seeks to explore a framework for organising, transferring, and visualising the flow of EIA information. It introduces LPGs, based on the Neo4j graph database to formalise critical knowledge related to EIA management. The EIA ontology (EIAO) is subjected to validation through the application of three real-life scenarios, demonstrating improved information retrieval and implicit reasoning capabilities. The findings indicate that this framework can offer robust support for smart decision-making systems, reducing the time required for organising and analysing EIA information by project managers. EIAO extends current knowledge in three key ways: 1) it broadens the application of ontology to EIA within the RAM domain; 2) it employs an innovative data model for storing and presenting the ontological information, thereby enhancing storage efficiency and visualisation of environmental management data; and 3) it provides enhanced search and reasoning capabilities, mitigating the need for manual intervention in road and environmental management projects, and subsequently reducing time and budget costs.

Documents also play a crucial role in the presentation and dissemination of various types of information in RAM. In recent years, substantial research has been devoted to automatic word extraction. However, the development of automatic models for extracting information from tables, which serve to convey structured and functional data, has not received comprehensive attention. Tables offer an efficient and effective means for readers to compare, interpret, and comprehend data, particularly when dealing with numerical values. Tabular data represent one of the most prevalent means of data presentation across a wide array of real-world applications, including recommender systems, online advertising, and portfolio optimisation. Notably, many machine learning competitions hosted on various platforms are predominantly focused on addressing challenges associated with tabular data. These competitions primarily revolve around the development of innovative solutions to problems related to tabular data, underscoring the paramount importance of tabular data in contemporary data science and machine learning endeavours. Nevertheless, the task of identifying tables within digital documents remains a challenge. To address this challenge, this research propose an automated information extraction model (ATIEM), specifically designed for tables. This model follows a self-supervised approach, focusing exclusively on tables, which sets it apart from previous approaches that have not adequately considered the distinctive features of tables, such as the relationships between table headers and corresponding values. The ATIEM model comprises two key components: a self-supervised transformer and a triple importer. Significantly, ATIEM achieving an F1 score of 90.4% on this specific subset, and automatic ontology establishment

approach can increase the number of triples and variety of relation types in the case, expanding them from 341 to 396 triples and the relation types from 39 to 40. By harnessing this enriched data, the model is efficiently trained to identify missing triples within the ontology, ultimately enhancing the comprehensiveness and semantic depth of the triples utilised in RAM.

List of figures

List of tables

Contents

Abbreviation

Abbreviation	Description
AEC	Architecture, Engineering And Construction
BIM	Building Information Modelling
CL2M	Closed-Loop Life Cycle System Management
DL	Description Language
EL	Expressing Language
GIS	Geographic Information System
GPS	Global Positioning System
IC-PRO-Onto	Infrastructure and Construction Process Ontology
IDEON	Intelligent Systems Technology Distributed Enterprise Ontology
IFC	Industry Foundation Classes
ISO	International Standard Organization
JADE	Java Agent Development Environment
LPGs	Labelled Property Graphs
NoSQL	Not Only SQL
OWL	The Web Ontology Language
PDA	Personal Digital Assistant
QL	Query Language
RDF	Resource Description Framework
RDFS	RDF Schema
RL	Reasoning Language
RTDSS	Roadside Tree Diagnosis Support System
SeRQL	Sesame RDF Query Language
SLR	Systematic Literature Review
SPARQL	SPARQL Protocol and RDF Query Language

SQL	Structured Query Language
SWRL	Semantic Web Rule Language
TDDS	Tunnel Defect Diagnosis System
VEACON	Vehicle Accident Ontology
W3C	World Wide Web Consortium
XML	Extensible Markup Language

1 Background and motivation

1.1 Introduction

This chapter provides an introduction for the background, research problems, aim, and objectives of this research; it also highlights the significance and contribution of this research.

1.2 Background

1.2.1 Importance of RAM

Roads are a critical asset for any community or society. They provide a means of transportation for people and goods, and are essential for economic activities and social interaction. Road networks can help improve access to essential services, such as education and healthcare, which can help reduce inequality and improve overall well-being of a society. Roads also have economic value, as they are a key factor in determining the feasibility and attractiveness of a location for businesses and residents. Well-maintained roads can increase property values and attract investments, while those in poor condition can have the opposite effect. In addition, road networks can help reduce the time and cost of transportation, which can have a positive impact on the overall efficiency of an economy. Thus, it is important for governments to have effective RAM systems (Halfawy et al., 2006).

Road asset management (RAM) is one of the largest infrastructure asset management sectors in the world. It is defined as a systematic process of maintaining, upgrading and operating roads (Reddy & Veeraragavan, 2011). For example, according to a report from the Department of Infrastructure and Transport of Australia, the value of road maintenance activities has reached $19.8 billion in 2022 (Kineber et al., 2022), which accounts for more than 5.2% of the total road network value in that country. RAM is conducted and standardised around the world, which, as a result, has promoted the process involved to a scientific management system (ISO 55000: 2014).

The key tasks for managing a road asset include: 1) developing a RAM plan. This plan should be based on a thorough assessment of the current conditions and needs of the road, as well as an analysis of future trends and challenges. 2) prioritising road maintenance and repair, which can help extend the lifespan of the road and minimise the need for more costly interventions. 3) monitoring and assessing the condition of the road. This can be conducted through visual inspections, or more detailed engineering assessments. 4) using data and technology to improve decision-making, and 5) engaging with stakeholders, such as local

communities, government agencies, and businesses. These tasks are vital for effective and efficient RAM.

Despite the importance of RAM, numerous practical challenges exist, such as massive information, dynamic data, isolated databases, and different stakeholders, which make it difficult to develop an effective and efficient RAM approach. For example, RAM involves a range of stakeholders, including government agencies, private companies, and the public. Engaging and coordinating with these stakeholders can be challenging. With massive amounts of data from different sources, objectively prioritising certain roads or projects over others can be difficult. Additionally, there is a growing need for quicker decision-making process in RAM, because the current decision-making process requires time-consuming data extraction from various sources. Moreover, uncertainties and non-uniform information in decision-making increase the cost and information loss in the process (Piyatrapoomi et al., 2004).

Textual information (e.g., standards, guidelines, and project documents), and road pavement data are stored in respective databases of the stakeholders, which increases the difficulty of sharing and using these data (Bennett et al., 2006). Current studies still focus on improving traditional management approaches. Given the fact that RAM contains complex and high frequency data sharing from various databases, applying a relationship-based integration system to manage various information, which can link data and provide quick and clear query to manage them is necessary (Halfawy, 2008).

The resolution of these problems requires a computer-based approach that is more cost-effective and efficient for the integration of databases (Liu & Chetal, 2005). Such an approach should be able to store information in a unified data format for easy reading and collection, and to edit the relationship of assets. In this way, road assets and their unique properties (such as location, pavement data, and other features) can be usefully linked to make informed decisions. In recent years, there are a few promising computer-based approaches including ontology, relational database, graph database, and automatic text extraction, which can be adopted; these are introduced in the following sections.

1.2.2 Ontology in RAM

As a data-driven and data-centred technology, ontology has significance in asset management because the properties and relationships of all the assets can be derived directly or by simulations (Innovation, 2009). Ontology, in computer science and information science, is defined as a description of the types and structures of objects, properties, processes, and relations (Smith, 2012). The concepts of Semantic Web and Linked Data, which have the same

core as ontology, were also considered to be ontology-based techniques in a few studies. Since Berners-Lee et al. (2001) introduced this concept into the computer science field, it has been rapidly applied in the architecture, engineering, and construction (AEC) field to facilitate engineering management (Ashraf et al., 2015). For example, Le and Jeong (2016) used ontology to improve the unification and interconnection of life-cycle data to support decision making in highway asset management. It can also be used to create a platform for stakeholders, who may have different backgrounds, to share their knowledge and understand asset management concepts more easily. For instance, Merdan et al. (2008) applied ontology in the transportation domain to share information among agents, and provide agreement and understanding on the commonly used concepts. Ontology can help managers in information integration steps with its digitised and linked data. For instance, an ontology-based life-cycle management approach is conducted to monitor heterogeneous real-time data in construction process (Corry et al., 2015). Zeb et al. (2015) also developed an ontology-supported asset information integrator system to assess capital assets.

Over the recent years, a few studies have also been conducted on the application of ontology in roads . Scientists found that ontology has significant value in RAM because the properties and relationships of all asset objects can be derived directly or through simulations (Reddy & Veeraragavan, 2011). As such, there is a need for more smooth information tracking and faster ontology reasoning in road asset field. For example, Grubic and Fan (2010) reviewed ontology-based supply chain management and categorised the studies into six ontology models, namely enterprise ontology, Toronto virtual enterprise (TOVE) ontologies, trans-European model, intelligent systems technology distributed enterprise ontology (IDEON), manufacturing system engineering ontology, and the model by Ye et al. (2008), which implemented the semantic integration of supply chain management.

However, a few research gaps remain. First, no systematic review in the field of RAM has been conducted. For example, Kiritsis (2013) reviewed the use of ontology in different engineering life cycle stages; however, the review did not identify road management aspects for which ontology can be usefully implemented. Grubic and Fan (2010) focused on ontologies in supply chain management; however, they only presented mature models and their application in this specific field. Secondly, there is a lack of standardised ontologies, which are especially designed for RAM. For instance, Bermejo et al. (2014) used ontology to manage traffic information and improve the road utilisation rate. Cordoba et al. (2017) applied an ontological expert system only to a single-lane road cross to improve the efficacy. Other existing studies have relatively narrow scope, such as risk management and highway assets

(Berdier, 2011; Corsar et al., 2015). Overall, a holistic view of the current development and implementation of the ontology technology has been lacking, and existing implementations cannot be directly extended and applied in all RAM processes.

1.2.3 Ontology and graph database

The mechanism of ontology involves transferring fragmented data from various formats into a uniform format that allows a computer to read and understand the information in a way similar to human logic (McGuinness & Van Harmelen, 2004b). Thus, special data models have been developed as a more natural way to represent data and information, such as ontologies (Saikaew et al., 2014).

There are two main types of databases used in ontology to store data. The first one is a relational database, which is adept at integrating structured data. However, entities and triples extracted from texts are unstructured knowledge. Although there is a word 'relational' within the name, relational databases do not appear to be suitable for storing interconnections among entities (Rahman et al., 2023). The word 'relational' refers to relating columns in a table, not relating knowledge in several different materials. Relationships among data exist to support information operations, which is totally different from relations among road asset data (where each data node is also called an entity). Hence, textual knowledge are usually stored as .txt or .csv files, which leads to difficulties in retrieval and analysis (Wang et al., 2020).

The other type is referred to as a graph database, which effectively handles knowledge that involves relationships. Typical graph databases include the resource description framework (RDF) and labelled property graph (LPG) databases (e.g., Neo4j). The first type is RDF-based data models. For example, RDF Triple Stores represents an information specification structure for the Worldwide Web Consortium (W3C). The data unit is stored in the form of *subject-predicate-object*, which is also known as *triples*, and resources can be linked by a set of triples (which form the graph). The second type is LPGs, which consist of a group of vertices and edges (Anikin et al., 2019). In LPGs, each vertex presents a unique instance, and each edge presents a unique relationship. A set of vertices and relationships form graphs, which provide a more intensive data structure (Angles & Gutierrez, 2018). Additionally, extra information can be attached to vertices and relationships as 'properties', which represent the main difference between the LPG and RDF methods (Angles et al., 2019). As an emerging technique, LPGs have not been extensively applied in ontology or engineering fields, although there is an increasing trend of considering this data model (Alocci et al., 2015; Das et al., 2014).

The suitability of data models needs to be investigated systematically by using carefully selected indicators. Based on previous studies, several evaluation benchmarks have been identified, which are data density, query efficiency, reasoning function and visualisation function (Alocci et al., 2015; Angles et al., 2019; Baken; Donkers et al., 2020). The differences between these two models have been compared in a few studies using these indicators. RDF graphs are one of the most classical and popular graph models used in ontology establishment; however, existing applications still have some challenges, such as large storage size and high device requirements (Das et al., 2014; De Abreu et al., 2013). On the other hand, LPGs attach the properties to vertices and edges, thereby improving the total structural efficiency (Das et al., 2014). This novel format has advantages, such as less storage space requirement and faster query paths (Vicknair et al., 2010). However, only a few studies have tried to gain an in-depth understanding of the advantages and disadvantages of the two models. For instance, Donkers et al. (2020) compared RDF graphs and LPGs in a smart home ontology, and concluded that LPGs have significantly better performance in full-text searching in their case. For dealing with linked data by graphs, RDF graphs perform better in creation and pattern matching, while LPGs have advantages in mining the depth and breadth paths of a large number of graphs (De Abreu et al., 2013). Although a comparison of data models was conducted from different perspectives, existing studies have mainly focused on the difference in storage features, and some have compared the query efficiency. Other indicators, such as reasoning and information visualisation, are also valuable and need to be analysed (Dudáš et al., 2018).

Manually scrutinizing and finalizing triples proves impractical, and the industry grapples with the absence of a computationally efficient approach for Knowledge Base Completion (KBC) and updates. Yet, creating automated methods poses challenges, demanding effective capturing of not just isolated information within nodes/edges but also discerning features, patterns of linkage, and paths across the entirety of nodes and edges in the datasets (Jiang et al., 2020). Using LPGs can help improve this automated process. For example, Zhu et al. (2022) adopted LPGs in building information modelling and geographic information system (BIM-GIS) information to improve automated information exchange and building condition monitoring. They found that the graph could significantly improve the data requiring multiple resource interactions and real-time updates.

Although some studies have highlighted the advantages of LPGs in automatic information processing, many problems still persist that need to be addressed (Bilal et al., 2016; Li et al., 2019). For example, a significant amount of tabular information exists that is stored in paper and electronic format (doc, pdf, etc.) for managing road assets. Research indicates that tables

contain 60% of the information, and that the values in tables have a disproportionately high priority (Jiang et al., 2020). Therefore, extracting information from tables is crucial. For instance, (Tensmeyer et al., 2019) developed a deep learning (DL)-based approach that takes contextual information into consideration to recognise biomedical entities in tables headers in randomised controlled trial articles, using a manually annotated corpus. However, the format and logic of the information in a table is different from that of a normal paragraph text. Related phrases are usually found before and/or after in a sentence with a symbol between them. Few studies have been carried out to optimise the automatic extraction of instances and relations for this feature of tables in RAM projects.

1.3 Problem statement

RAM is becoming increasingly complex and requires timely information to make decisions. Novel technologies, especially internet- and computer-based methods are needed in this field. Moreover, different stakeholders may have different understandings of an entity. Therefore, information exchange between different databases is difficult and inefficient. Some studies use ontology to solve these problems; however, they also have some limitations, including lack of specific ontology engineering approach for road assets, limited ontology techniques, and lack of an automatic mechanism for content generation. These limitations are elaborated as follows.

1.3.1 Lack of specific ontology engineering approach for road assets

Road assets consume a major portion of maintenance funding and resources. RAM is complicated owing to a number of factors, such as large volume of information, relations, multiple participants, and tight schedules. Implementing modern data management methods, such as ontology, can contribute to the success of RAM projects. However, studies on RAM concentrate on engineering techniques and approaches, while few efforts are made to improve the automation or computer-based management aspects. On the other hand, it is observed that although the general ontology development process is well defined, some specific features of RAM may require fine tuning before ontology can be used. For instance, a more static situation (e.g., in the design and planning stages) requires a standard and formal knowledge acquisition for ontology (Das et al., 2015). On the other hand, dynamic situations (e.g., operation and maintenance stages) require efficient data storage and high-speed data exchanging. However, existing studies have not identified the unique characteristics of these life-cycle stages,

resulting in a lack of uniform ontology engineering approaches to accommodate these challenges. Other engineering fields have already piloted some widely accepted models to improve the understanding and building of ontologies, such as TOVE and IDEON ontology for supply chain management (Grubic & Fan, 2010). The lack of best practices in RAM domain caused sporadic knowledge collection and weak ontology integration for linked data.

1.3.2 Difficulty in selecting suitable ontology techniques

The selection of suitable ontology techniques depends on the aim and scope of the implementation. For instance, ontology is a more efficient approach for searching the target information in a documentational dataset, such as finding a special requirement for traffic lights in RAM standards (Koukias et al., 2015a). Note that most of the studies in RAM used RDF, or even RDF serialisation (e.g., RDF/XML, Notation3, N-Triples, and Turtle) as the data models. However, current ontologies on RAM have not provided sufficient reasons for RDF being the most suitable approach for representation compared to other approaches (such as LPGs). When an ontology is required to be established, the available tools are also limited. More than 80% of ontologies under RAM were developed by Protégé (an RDF-editing software) (Koukias et al., 2015a).

Other ontological data models have recently been introduced in information management systems. Some of the latest studies in other fields have begun to use more efficient storage syntaxes, such as RDF* and LPGs (Gong et al., 2018). Gong et al. (2018) compared LPG- and RDF- triple models using an oilfield ontology, and observed that LPGs have advantages over RDF in query efficiency for large datasets. The friendly interface, low programming requirements, and open resources are the reasons for its popularity in this field (Gennari et al., 2003). However, while the homogenisation of ontology techniques may provide more opportunities for cooperation and comparison among ontologies, it also limits the opportunity of benefiting from using other innovative approaches (Das et al., 2014).

1.3.3 Lack of automatic mechanisms for special data types in RAM for ontology

The next challenge is automatically creating ontology elements and relationships based on existing data. Ontology techniques aids in the transfer of RAM data into machine-processable information; however, the initial transition from traditional datasets into ontology data formats still requires significant manual work; especially special and important data types, such as tabular data are required. An automatic mechanism to capture instances, properties, and relationships is required (Gould & Cheng, 2016). Some studies have been conducted to address

this specific problem. For example, Nyulas et al. (2007) created batch imprinting plug-ins for Protégé, which can automatically convert spreadsheet information into triples. However, such attempts are insufficient, because of the increasing mega data scale and structural complexity. Moreover, from the perspective of ontology creation, the rule-based automatic mechanism can achieve new data creation and mapping in the current ontology during the usage. In some relevant fields, such as tunnel and bridge maintenance, an automatic mechanism has been conducted for years. For instance, a semantic web-based tunnel defect diagnosis system was used to automatically set up the link within structural defects in underground transpiration tunnels (Hu et al., 2019). However, current new rules for automatic reasoning must be translated and manually input into the software.

1.3.4 Scope and aim/objective

To tackle the aforementioned challenges, this research endeavors to formulate an information extraction and integration approach. This approach is designed to construct comprehensive key concepts and relationships for RAM projects. It leverages the automatic extraction of entities and relations through innovative ontologies and machine learning models. The approach can improve the implementation of management effectiveness in RAM (especially in environmental impact assessment projects).

To realise the aim, the following objectives have been identified.

1) Objective 1 (O1): To obtain an in-depth understanding of ontology approaches and their implementations in RAM areas.

The number of studies of RAM and engineering ontology is large. Thus, a critical review has been conducted, following a popular review approach having eight steps. Web of Science was chosen as the main database, while other online databases, such as Google Scholar was chosen as the additional database. The articles were collected from the Web of Science database. In contrast, currently ontology in RAM has not been well-studied. First, articles were collected by searching the Web of Science database. Then, standards and case reports were collected from other online databases, such as the Construction Industry Institute (CII) and Mainroads.

2) Objective 2 (O2): To conduct a systematic comparison between ontology establishment techniques to identify the most proper one for RAM.

After a literature review in O1, two initial graph databases were identified: RDF and LPGs, which present the most classical and novel approaches, respectively. These two ontologies

were systematically examined through a detailed comparison on a few benchmarks identified from literature.

3) Objective 3 (O3): To develop ontological KBs to integrate the information in environmental impact assessment in RAM.

An ontology-based framework was to be created for RAM. The ontological KBs were developed based on standard guidelines and domain knowledge collected in O2. Establishing an ontology involved seven steps: 1) define the scope of ontology; 2) consider reusing ontologies; 3) acquire ontology knowledge; 4) define ontology structure; 5) define ontology establishing environment and data model; 6) establish ontology; and 7) validate and improve. The end product would be a machine-readable ontology for the RAM process, and some real-world scenarios were tested to demonstrate the development.

4) Objective 4 (O4): To realise automatic information extraction and ontology establishment for RAM from tabular data using a DL model.

No approaches currently exist for automatic data extraction and integration for tabular data in a graph-based ontology framework. To address this problem, this thesis developed a novel automatic method for tables to be sorted into groups and automatically put into ontology databases. In this methodology, a node represents a project entity surrounded by several interconnected nodes forming its neighborhood. All nodes and relations are encoded as numerical vectors, commonly referred to as embeddings. Textual data is translated into a computer-readable language through a vectored matrix. Subsequently, the decoder predicts missing triples from the table in the following steps: 1) it takes new embeddings as inputs, 2) for each node in the triples, it identifies nodes without existing relations, 3) it traverses these nodes and predefined relations, creating candidate triples, 4) it computes a validation score for each triple using a deep learning structure akin to the Robert model, and 5) establishes validated triples in the ontology.

1.3.5 Significance

Road assets are critical assets in the society (Morgan, 2012). In modern RAM, project planning and implementation require more and more detailed knowledge and automatic approaches (Fernandes, 2000). In traditional RAM, many activities, such as maintenance, relies on a lot of manual works to collect knowledge and data from paperwork, human experience to make informed decisions. This study addresses these problems by developing an ontological data management approach, where DL techniques are also applied to automatically extract entities and relationships. Accordingly, there are three main contributions.

First, this study is one of the first studies to propose an ontology to integrate asset information in the monitoring and assessment phases of road infrastructure. Previous information integration efforts in this aspect largely rely on paper materials, which were inefficient and labour-intensive. The lack of such integrated approaches can largely increase the cost of project and the conflicts between stakeholders, as in-time information cannot be used to effectively support management functions (e.g., information searching, reasoning, and decision-making). The proposed method allows project participants to retrieve information and their relations to perform essential management functions, which can facilitate RAM activities, i.e., the evaluation of project progress, asset statuses, and project participants' performance (Wang et al., 2020).

Second, this study addresses a critical issue in ontology implementation, which is related to the advantages and disadvantages of the most two commonly adopted ontology techniques, RDF and LPGs. From a theoretical perspective, RDF and LPGs both have graph-based information structure. RDF graphs are triples with standardised rules to present, while LPGs have the richness of detail given data (Alocci et al., 2015). Previous studies focus on data size, whilst other important factors such as querying efficiency, reasoning and data visualisation are rarely included.

Third, this research makes a practical contribution by providing an automatic information extraction and modelling tool for completing triples in ontology of RAM projects. Ontology is an effective tool to manage RAM projects, which usually involve complex data exchange and querying. However, the current RAM process requires manual information searching and transferring, which can delay the information flow, and hence is prone to errors. This research proposed an information extraction and integration approach to automatically extract target information from materials via DL techniques, and then automatically link them to an existing ontology Therefore, it is an early attempt in this field to improve the current RAMs. It can save time and cost for the project team (especially those lacking experience) to understand interconnections among project data elements and facilitate decision-making.

1.4 Thesis structure

This thesis has seven chapters which are summarised below and in Figure 1-3.

Chapter 1 describes the background, research problems, aim and objectives of this thesis, as well as the thesis structure.

Chapter 2 summarises the literature on road asset, information management in the AEC industry, and information extraction and integration approaches (i.e., ontology mechanisms and applications, and automatic information extraction models).

Chapter 3 introduces the research methodology. It outlines the research philosophy that underpins the research method. Then, the chapter introduces the method for comparison between ontology techniques, the method for establishing the ontology for EIA in RAM, and the method for developing the automatic information extraction model.

Chapter 4 performs a series of experiments to compare two ontology models based on five benchmark indicators, namely, data density, query efficiency, reasoning and data visualization.

Chapter 5 constructs an ontology for environmental impact assessment process in RAM to integrate extracted constraint entities and relations (i.e., entity-relation-entity triples). The development of the ontology is based on a widely adopted ontology guideline and comprehensive collection of domain knowledge in this field.

Chapter 6 Chapter 6 develops an automatic information extraction model and ontology establishment approach to identify tabular triples in ontology. The model is developed based on a Robustly Optimised BERT Pretraining Approach (RoBERTa) and consists of two parts: a pre-trained information extraction model and a triple-inputting module. Domain information is utilised to improve model performance. Experiment results are investigated to demonstrate the model's performance and the effect of utilising domain information.

Chapter 7 concludes important findings in the thesis, highlights contributions and implications, discusses limitations in this research, and suggests future studies.

2 Literature review

2.1 Introduction

This section aims to provide the latest advances in the development and implementation of ontologies in road asset management. It summarises the life-cycle stages of road asset management and reviews efforts of applying ontology for managing road assets (e.g., what asset types have been covered and what technique have been conducted). Based on the research aim and objectives stated in Chapter 1, this section explains the necessity and current development when implementing ontology. Detailed information for the review method is provided in Section 3.3.

2.2 Background

As road networks are expanding, efficient management of infrastructure and assets is becoming challenging for governments and industries. The design, construction, operation, and maintenance of these road networks require a large amount of data to be collected, stored, transferred, and analyzed (Najafi & Bhattachar, 2011). However, massive data and information transformation and exchange among isolated databases in project contractors, private agencies, and public organisations make information sharing rather difficult (Zhang et al., 2015). Traditional methods or systems of road asset management rely heavily on humans, and they also have other limitations such as being costly and highly uncertain (Möller & Beer, 2008). These problems require a more cost-effective, efficient, and computer-based approach for the integration of databases (Liu & Chetal, 2005).

Ontology, a term first appearing in the philosophy field, is defined as a description of the types and structures of objects, properties, processes, and relations in computer science and information science (Smith, 2012). The concepts of Semantic Web and Linked Data, which have the same core as ontology, were also considered to be ontology-based techniques in a few studies. Since Berners-Lee et al. (2001) introduced this concept into the computer science field, it has been rapidly applied in the architecture, engineering, and construction (AEC) field to facilitate engineering management (Ashraf et al., 2015). For example, Le and Jeong (2016) used ontology to improve the unification and interconnection of life-cycle data to support decision making in highway asset management. As a data-driven technology, ontology has significant value in road asset management because the properties and relationships within all asset objects can be derived directly or through simulations (Reddy & Veeraragavan, 2011).

Ontology can increase efficiency in data management. It can also be used to create a platform for stakeholders, who may have different backgrounds, to share their knowledge and understand asset management concepts more easily. For instance, Merdan et al. (2008) applied ontology in the transportation domain to share information among agents and provide agreement and understanding on the commonly used concepts.

Over recent years, isolated review studies have been conducted on the application of ontology in various road asset management subjects (Wu et al., 2021). For example, Pauwels et al. (2017) reviewed ontology technologies in the AEC industry, observing that ontologies can link domains and offer data interoperability and logical inference functions to the industry. Following this research, Yang et al. (2019) reviewed 116 papers and presented a comprehensive summary of the state-of-the-art ontology-based systems in engineering, and they proposed a roadmap to facilitate the application of ontology. As a part of road asset management, Grubic and Fan (2010) reviewed ontology-based supply chain management and categorised the studies into six ontology models, including enterprise ontology, Toronto Virtual Enterprise (TOVE) ontologies, Trans-European model, Intelligent Systems Technology Distributed Enterprise Ontology (IDEON), manufacturing system engineering ontology, and the model by Ye et al. (2008), which implemented the semantic integration of supply chain management. However, a few research gaps remain. No systematic review for the field of road asset management has been conducted. Over the past few years, the road asset management field has implemented ontologies because of their value in infrastructure management systems (Berdier, 2011; Corsar et al., 2015). However, existing reviews often focus on specific aspects of road asset management and have relatively narrow scopes. For example, Kiritsis (2013) reviewed how ontology aids in different engineering life cycle stages, but the review did not identify other road management aspects. Grubic and Fan (2010) focused on ontologies in supply chain management, but they only presented mature models and their application in certain fields. They lack a holistic view of the current development and implementation of the technology.

2.3 Road asset management and ontology

2.3.1 Quantitative analysis

This section presents a statistical analysis of the selected papers. Figure 2-1 shows the number of papers by publication type. Journal papers accounted for 79% (47 out of 69) of the selected publications, while proceedings from conferences contributed 21%. Note that almost

56% of the studies were published after 2014, which indicates an increasing interest of researchers in this topic.

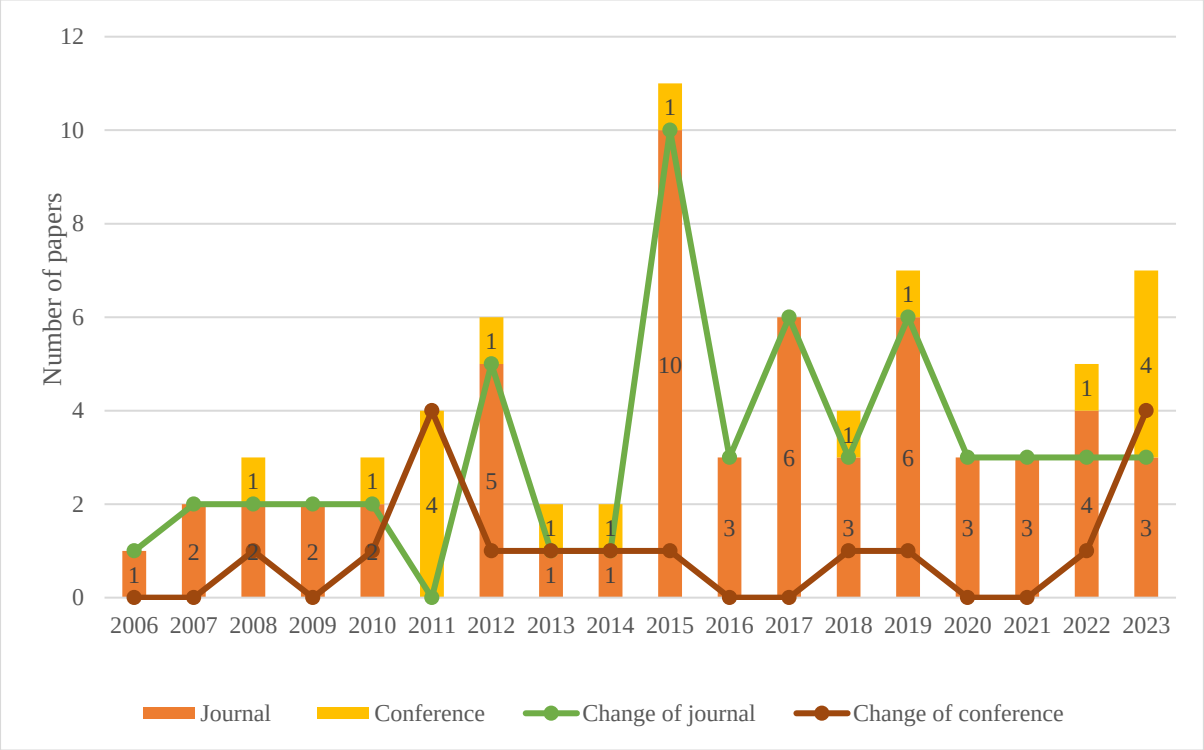


Figure 2-1 Trends of publications

Figure 2-2 shows the distribution of studies by country/region from 2006 to 2023. Forty-seven out of the 69 publications were from Europe and North America, demonstrating relatively higher research interests in this specific topic from these two regions.

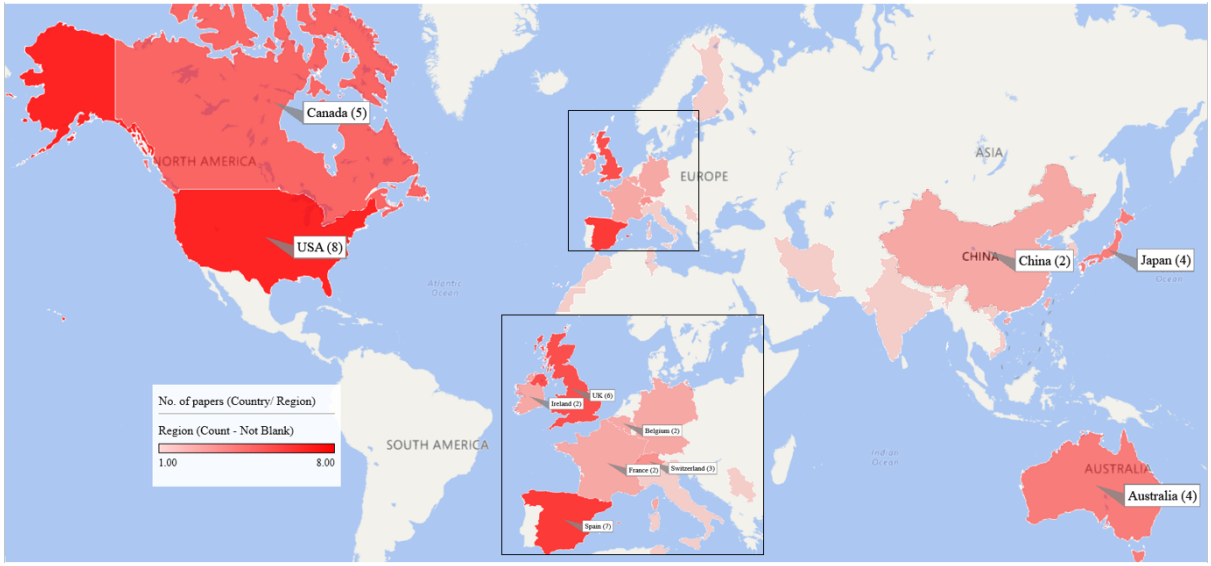


Figure 2-2 Distribution of publications by country/region.

2.3.2 Qualitative analysis

Based on International Organization for Standardization (ISO) 55000: Asset management – overview, principles, and terminology (ISO, 2014), relevant road asset management guidelines, including Guide to Asset Management (Austroads, 2016), and peer-reviewed studies on road asset management (Kiritsis, 2013; Yang et al., 2019), ontology implementation in road asset management can be described from two perspectives, i.e., asset type and life cycle stage.

2.3.2.1 Asset type

Based on the best practice and standards, assets in road asset management are primarily classified into five groups: traffic service assets, road assets, property assets, data assets, and other assets (ISO, 2014; Kiritsis, 2013). Table 1 lists previous studies on road asset management using ontology, categorized by asset type.

Table 1 Summary of asset types in ontology studies in road asset management.

Areas (Number of studies)	Reference
Traffic service assets (38)	Ceausu and Despres (2006); Hornsby and King (2008); Vallejo et al. (2009); Zhai et al. (2009); Houda et al. (2010); Wang and Wang (2011); Barrachina et al. (2012); Du et al. (2012); Gregor et al. (2012); Jelokhani-Niaraki et al. (2012); Malgundkar et al. (2012); Stocker et al. (2012); Bermejo et al. (2013); LécuéTallevi-Diotallevi et al. (2014); LécuéTucker et al. (2014); Corsar et al. (2015); Mohammad et al. (2015); Toulni et al. (2015); Watson et al. (2015); Zapater et al. (2015); Zhao et al. (2015); Gould and Cheng (2016); Fernandez et al. (2016); Consoli et al. (2017) ; Lee et al. (2017); Fernandez and Ito (2017); Czarnecki (2018); Nguyen and Nguyen (2019); Van de Vyvere et al. (2019); Wu et al. (2019); (Shaaban et al., 2020); (Shaaban et al., 2021) Spoladore et al. (2023) Isailović and Hajdin (2022)
Road assets (12)	Halfawy (2008); Hülsen et al. (2011); Yabuki et al. (2011); Kiritsis (2013); Zeb et al. (2015); Le and Jeong (2016); Cordoba et al. (2017); Zeb (2017); France-Mensah and O'Brien (2019); Lim et al. (2019); (Pokusaev et al., 2020); (Liu et al., 2021a)
Property assets (2)	Kaza and Hopkins (2007) Du et al. (2023)
Data assets (6)	Koukias et al. (2015b); Koukias and Kiritsis (2015); Niestroj et al. (2018); Ali et al. (2019); (Kupriyanovsky et al., 2020); (Dao et al., 2021) He et al. (2023)
Other assets (9)	Merdan et al. (2008); El-Gohary and El-Diraby (2010); Berdier (2011); Das et al. (2015); Zhang et al. (2015); Beetz and Borrmann (2018); Niestroj et al. (2019); Espinoza-Arias et al. (2022) Borgo et al. (2022)

- Traffic service assets. These are all assets relevant to traffic systems, such as signals, marking, lighting, and safety devices. Traffic service assets are the most important type of asset in which ontology is implemented.

The aim of road assets and infrastructure management is to provide services for road users (Zapater et al., 2015). As a result, more than 58% of the studies were related to traffic service

assets. On this specific subject, some studies focused on traffic management, including traffic condition information collection, analysis, and sharing. For example, ontology can be used to structure sensor-based information to predict traffic congestion, which can aid drivers in selecting better routes (LécuéTucker et al., 2014). In addition, the global positioning system (GPS) information of a road can also be digitised using ontology, which would provide more integrated data for road management authorities (LécuéTucker et al., 2014). Sensor ontologies can also aid the collection of digital information on weather, road works, events, moving objects, and accidents. Such collected data were processed using a computer and reused in a traffic system in Dublin city (Hornsby & King, 2008; LécuéTallevi-Diotallevi et al., 2014). Moreover, transportation and travel data can be analysed by ontologies for more efficient use of road assets (Corsar et al., 2015; Toulni et al., 2015; Zapater et al., 2015).

Most of the risks on roads are related to traffic; therefore, traffic management is also a priority (Zapater et al., 2015). Accidents on roads can cause significant problems, such as injury to road users, waste of time, increased cost, adverse effects on the environment, and damage to the economy (Barrachina et al., 2012; Thakkar & Lohiya, 2020). On this specific topic, Ceausu and Despres (2006) built an ontology for accidentology and terminology of on-road accidents. This was a preliminary study, but it confirmed the feasibility of using ontology in this scenario. Road event data (Jelokhani-Niaraki et al., 2012), road condition data from cameras (Mohammad et al., 2015), and traffic accident data (Wang & Wang, 2011) were then integrated into ontologies to understand their hierarchy, relations, and interconnections, which can also be reasoned, shared, and reused. The behaviour of road users (e.g., drivers) has a relatively high effect on risks; thus, improving users' behaviour was also investigated to minimise risks (Hülsen et al., 2011). To reduce the chance of accidents involving novice drivers, Nguyen and Nguyen (2019) applied a fuzzy ontology to collect road information to simulate traffic situations. By learning and understanding the emergency events on a road, novice drivers can gain extra experience before they actually begin to drive. Ontology can also increase information accuracy after an accident. For example, Watson et al. (2015) attempted to correct the under-reporting injuries caused by accidents using Linked Data, which provided implications for road safety research, policies, and funding.

Other fields in traffic service management have also used ontology techniques. For example, to achieve better traffic flow and decision making, an ontology was implemented to provide valuable and efficient information for traffic light systems (Van de Vyvere et al., 2019). Normal data fragments, real-time data, and long-term historical data can be used through traffic light ontologies to predict and minimise accidents on roads (Fernandez & Ito, 2017; Van de

Vyvere et al., 2019). Intelligent or automated transportation systems require a large amount of information and information processing, which ontologies can aid and provide support for informed decision making (Gregor et al., 2012; Zhao et al., 2015). Two studies used ontological approaches to plan and record data with vector features, such as determining the shortest path (Houda et al., 2010) and journey route planning (Lee et al., 2017).

- Road assets. Road assets are all the facilities and relevant information that belong to road systems, including earthworks, pavements, shoulders, and roadside areas.

The life-cycle management of road structures is important and requires a significant amount of information, and ontology can be applied to increase the efficiency of information storage and extraction (Ashraf et al., 2015). For example, Kiritsis (2013) presented a closed-loop life cycle system (CL2M) with ontology techniques for the management of engineering assets. Zeb et al. (2015) proposed a semantic web framework with a four-step method to share life-cycle information such as design knowledge and workflow. To facilitate the usage of this integrated approach, Du et al. (2023) propose a model of city infrastructure assets and their interdependencies, providing details on how asset properties and processes affect each other. This model is represented as ontologies in OWL 2 which can be read and interpreted by machines automatically.

Unlike the management of roads, highway management is a specific subject that has been investigated separately because of its high value and strategic significance in the social economy (Le & Jeong, 2016). As multiple agencies and stakeholders are involved in highway projects, the use of ontology can produce benefits by linking different stakeholders, improving the classification and interconnection of life-cycle data of highway assets, and supporting various decision-making procedures in highway asset management (France-Mensah & O'Brien, 2019; Le & Jeong, 2016).

A road crossing is another important road asset that includes multiple asset types, such as vehicles, users, signals, and assets. Studies in this domain typically focus on the decisions made by drivers when they are at a crossing (Cordoba et al., 2017; Hülsen et al., 2011). Ontology techniques have also been used for other road asset types, such as sewage systems (Halfawy, 2008) and roadside trees (Yabuki et al., 2011).

- Property, data, and other assets. This category refers to road management facilities, road information storage and management systems, and other general road systems and information that cannot be grouped into any of the above categories. For example, ontology has been used to provide a uniform understanding of guidelines and standards (e.g., ISO

19115-1 and ISO 55002) through documentational analysis (Niestroj et al., 2018). Some researchers have implemented ontology rules to present document content, which can be used in asset operation, maintenance, and configuration (Koukias & Kiritsis, 2015). Merdan et al. (2008) attempted to manage the relationships between stakeholders and projects by using a rule-based ontology framework for the cooperation of different tasks. Beetz and Borrmann (2018) focused on information integration using Linked Data during asset management.

2.3.2.2 Life-cycle stage

Scholars have widely acknowledged that there are four main management stages throughout the entire life cycle of road assets and products: planning, construction, operation, and maintenance (Austroads, 2016). We categorised studies based on the life-cycle stages that they focus on. The results are shown in Table 2.

Table 2 Summary of selected papers by life-cycle stages.

Life cycle stages (Number of studies)		Reference
Planning (4)		Kaza and Hopkins (2007); France-Mensah and O'Brien (2019); (Dao et al., 2021); (Borgo et al., 2022)
Construction (4)		Das et al. (2015); Zhang et al. (2015); (Kupriyanovsky et al., 2020);
Operation & Maintenance (47)	General (27)	Hornsby and King (2008); Zhai et al. (2009); Houda et al. (2010); Yabuki et al. (2011); Stocker et al. (2012); Jelokhani-Niaraki et al. (2012); Gregor et al. (2012); Malgundkar et al. (2012); Du et al. (2012); LécuéTallevi-Diotallevi et al. (2014); Corsar et al. (2015); Koukias et al. (2015b); Toulni et al. (2015); Zapater et al. (2015); Zeb et al. (2015); Fernandez et al. (2016); Fernandez and Ito (2017); Czarnecki (2018); Van de Vyvere et al. (2019); Merdan et al. (2008); Berdier (2011); Consoli et al. (2017); (Liu et al., 2021b);(Kupriyanovsky et al., 2020); (Liu et al., 2021a); (Espinoza-Arias et al., 2022); He et al. (2023)
	Risk management (19)	Ceausu and Despres (2006); Vallejo et al. (2009); Hülsen et al. (2011); Wang and Wang (2011); Barrachina et al. (2012); Bermejo et al. (2013); LécuéTucker et al. (2014); Mohammad et al. (2015); Watson et al. (2015); Zhao et al. (2015); Gould and Cheng (2016); Cordoba et al. (2017); Niestroj et al. (2018); Wu et al. (2019); Niestroj et al. (2019); Nguyen and Nguyen (2019); (Shaaban et al., 2020);(Shaaban et al., 2021) ;Poveda-Villalón et al. (2022)
Entire life cycle (10)		Halfawy (2008); El-Gohary and El-Diraby (2010); Kiritsis (2013); Le and Jeong (2016); Zeb (2017); Beetz and Borrmann (2018); Ali et al. (2019); (Pokusaev et al., 2020); Isailović and Hajdin (2022) ;(Du et al., 2023)

- Planning and construction. These two life-cycle stages have attracted limited research interest. The design of work flow was the main purpose for the studies in these two stages. Ontology played a role to ensure that knowledge is standardised. For example, Zhang et

al. (2015) develope a construction safety knowledge ontology for the workers for fast-training purpose.

- Operation & Maintenance. This stage refers to the operation and maintenance of road-related assets. More than 70% of the papers are related to operations, probably because that this stage requires a significant amount of information for effective decision making (Bennett et al., 2007). In these studies, ontologies are proven to be effective in supporting fast information and data exchanging. Some notable examples include the management of traffic and asset condition information (Czarnecki, 2018; Houda et al., 2010; Malgundkar et al., 2012; Toulni et al., 2015), road equipment management (Gregor et al., 2012), and road structure management (Yabuki et al., 2011). Only a few of studies were related to road maintenance activities. For instance, Berdier (2011) developed an urban ontology for maintaining road systems and aiding organisations to manage engineering activities when lacking coordination tools.

Risk management, as a key topic in operation and maintenance, is listed separately in Table 3 because of the large number of publications on this subject. Researchers have used ontologies to achieve efficient data exchange and build synonymity for accidents and road events to reduce risks (Vallejo et al., 2009; Wang & Wang, 2011; Wu et al., 2019). In this domain, processing real-time data from sites to the management system was a key issue, which can be supported by formalized ontological information.

- Entire life cycle. Some studies used ontologies for efficient data exchange in software data integration (Beetz & Borrmann, 2018; Halfawy, 2008), knowledge sharing among stakeholders (El-Gohary & El-Diraby, 2010), documentation sharing (Ali et al., 2019), and highway management (France-Mensah & O'Brien, 2019) in the entire life-cycle of roads. Ontologies have also been used to improve collaboration in supply chain management during the construction of road projects to reduce costs and avoid risks (Das et al., 2015). (Poveda-Villalón et al., 2022) the Linked Open Terms (LOT) methodology, a comprehensive and streamlined approach tailored for ontology construction. This methodology draws inspiration from existing methodologies while centering its orientation towards semantic web advancements and technologies.

2.4 Ontology techniques used in engineering field

This section aims to analyse ontology techniques from the perspective of ontology engineering, which is related to the ontology knowledge formalization and presentation process

(Scheuermann & Leukel, 2014). The analysis framework of this paper follows the work of Yang et al. (2019) and Ashraf et al. (2015), while some adjustments (e.g., specific ontology modelling approach and access situation) have been made. The analysis focuses on the principles, methods, and tools for initiating, developing, and maintaining ontologies (Scheuermann & Leukel, 2014). The main components in ontology engineering are ontology modelling approach (knowledge development and formalization), ontology tool, data representation, serialization and querying (ontology implementation and presentation) and accessibility (Yang et al., 2019). Ontology modelling approach represents what type of ontologies are used and what domains they target at. After the modelling approach, what tools or platforms will be used to edit ontology from the software engineering perspective needs to be addressed. These two steps aim to formalize ontology from documents to knowledge. Data representation, serialization and querying languages refer to professional techniques used to store, form and use ontology from the computer science field and their implementation in road asset management will be analysed. Figure 2-3 presents the relationships of these steps in ontology engineering. The results of the overall analysis are shown in Figure 2-3.

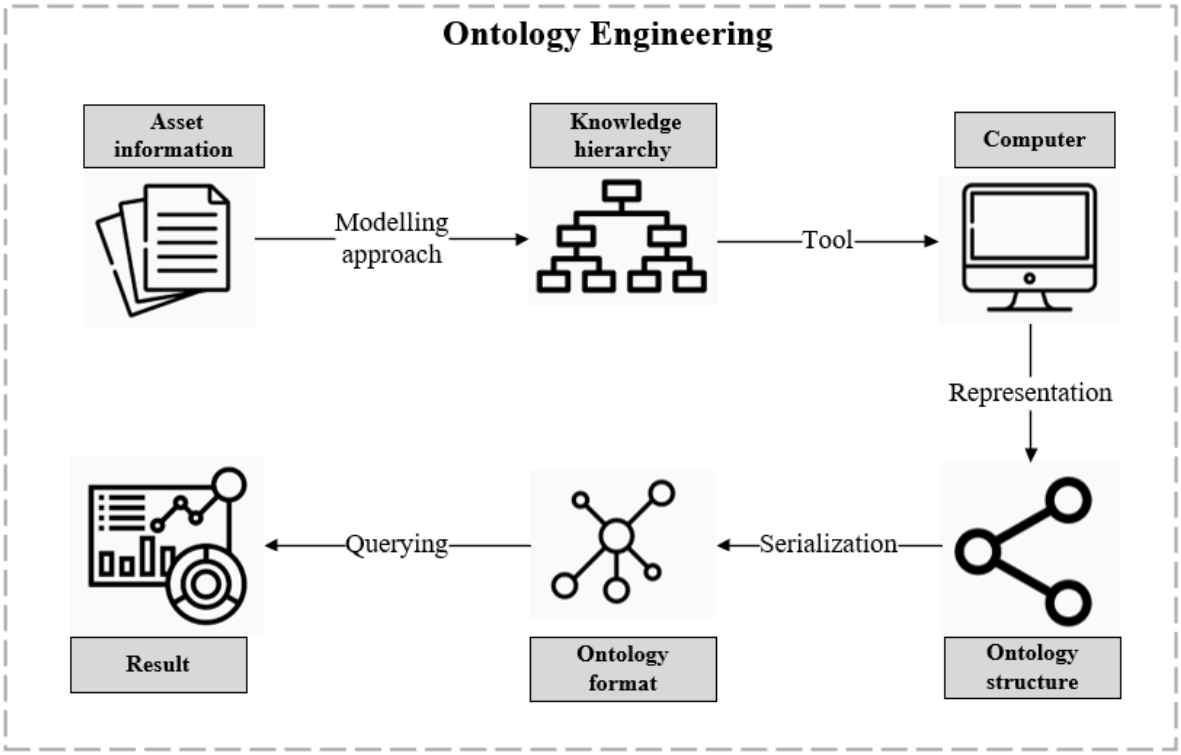


Figure 2-3 The processes in ontology engineering of road asset management.

Table 3 Summary of the selected papers from the ontology engineering perspective.

Author	Ontology engineering					Open access
	Modelling approach	Tool	Data Representation	Data Serialization	Data Querying	
Ceausu and Despres (2006)	ACCident TO Scenarios (ACCTOS)	-	OWL[1]	-	-	-
Kaza and Hopkins (2007)	Information System of Plans (ISoP)	-	-	-	-	Y
Merdan et al. (2008)	-	JADE[2], Protégé[3]	OWL	-	-	-
Hornsby and King (2008)	Suggested Upper Merged Ontology (SUMO)	-	Relational database	-	SQL[4]	-
Zhai et al. (2009)	-	-	RDF[5]	-	SeRQL[6]	-
Vallejo et al. (2009)	-	Protégé	OWL	-	OWL-to-PROLOG[7]	-
Houda et al. (2010)	-	Protégé	RDF	-	SWRL[8]	Y
El-Gohary and El-Diraby (2010)	Infrastructure and Construction PROcess Ontology (IC-PRO-Onto)	-	OWL	N-triples	-	Y
Svetel and Pejanović (2010)	-	-	RDF	XML[9]	-	-
Berdier (2011)	-	-	RDF	XML	-	-
Yabuki et al. (2011)	Roadside Tree Diagnosis Support System (RTDSS)	Hozo[10]	MySQL[11]	-	-	Y
Wang and Wang (2011)	Ontology-based traffic accident risk-mapping (ONTO_TARM)	-	-	-	-	-
Jelokhani-Niaraki et al. (2012)	-	-	OWL	-	-	-
Barrachina et al. (2012)	VEhicular ACcident ONtology (VEACON)	-	OWL	-	-	-
Gregor et al. (2012)	-	-	RDF	N-triples	-	Y
Du et al. (2012)	-	-	OWL	-	-	-
Kiritsis (2013)	Linked Design Ontology (LDO)	Protégé	OWL	JSON-LD[12]	-	Y
LécuéTallevi-Diotallevi et al. (2014)	Semantic Traffic Analytics and Reasoning for CITY (STAR-CITY)	-	OWL (OWL2EL[13])	-	-	Y
LécuéTucker et al. (2014)	-	-	OWL	-	-	Y
Das et al. (2015)	-	Protégé	Cassandra[14]	XML	-	-
Zeb et al. (2015)	Ontology-supported asset information integrator system (AIIS)	-	OWL	XML	-	-
Zhao et al. (2015)	-	-	-	-	SWRL	-
Zhang et al. (2015)	Ontology-based job hazard analysis (JHA)	Protégé	-	-	SWRL	-
Mohammad et al. (2015)	-	-	-	-	SWRL	-
Corsar et al. (2015)	-	Linked Open Data[15]	RDF	N-triples	-	Y
Zapater et al. (2015)	Road traffic information web service (WSs)	-	OWL	-	-	-
Toulni et al. (2015)	Vehicular Ad-hoc NETwork (VANET)	-	OWL	-	-	-
Koukias and Kiritsis (2015)	Technical Documentation Ontology (TDO)	-	-	-	-	-

Fernandez et al. (2016)	-	Protégé	OWL	-	-	-
Gould and Cheng (2016)	-	Protégé	OWL	-	-	Y
Le and Jeong (2016)	-	-		XML	-	Y
Fernandez and Ito (2017)	The Semantic Sensor Network	Protégé	OWL	Turtle	-	-
Consoli et al. (2017)	-	-	RDFS[16]	N-triples	-	Y
Cordoba et al. (2017)	SesToCross	-	-	-	-	-
Lee et al. (2017)	University activity ontology (UAO)	-	-	-	-	Y
Zeb (2017)	Eco asset ontology (EA_Onto)	Protégé	OWL	-	-	-
Niestroj et al. (2018)	-	-	OWL	-	-	-
Beetz and Borrmann (2018)	-	-	OWL	-	SPARQL[17]	Y
Wu et al. (2019)	Topological semantic trajectory (TOST)	-	MySQL	-	-	-
Niestroj et al. (2019)	The OpenStreetMap (OSMAP) ontology	Protégé	OWL	-	-	Y
Van de Vyvere et al. (2019)	-	-	RDFS	-	-	-
Ali et al. (2019)	Ontology and latent Dirichlet allocation (OLDA)	Protégé	OWL	-	-	Y
France-Mensah and O'Brien (2019)	Integrated highway planning ontology (IHP-ONTO)	Protégé	OWL	-	SWRL	Y
Nguyen and Nguyen (2019)	-	Protégé	OWL	-	-	-
(Poveda-Villalón et al., 2022)	the Linked Open Terms		OWL			
(Borgo et al., 2022)			OWL			
(Du et al., 2023)			OWL 2			

Note:
1. OWL = The Web Ontology Language
2. JADE = Java Agent Development Environment
3. https://protege.stanford.edu/
4. SQL = Structured Query Language
5. RDF = Resource Description Framework
6. SeRQL = Sesame RDF Query Language
7. http://www.jiprolog.com/
8. SWRL = Semantic Web Rule Language
9. XML = Extensible Markup Language
10. http://www.hozo.jp/
11. https://www.mysql.com/
12. JavaScript Object Notation for Linked Data
13. https://www.w3.org/TR/owl2-profiles/
14. https://cassandra.apache.org/
15. https://www.w3.org/egov/wiki/Linked_Open_Data
16. RDFS = RDF schema
17. SPARQL = SPARQL Protocol and RDF Query Language

2.4.1 Ontology modelling approach

In this review, a total of 23 ontology modelling approaches were identified. 15 out of the 23 modelling approaches follow Ontology Development 101, a widely accepted ontology guide, and have three steps, which are specification, acquisition and formalization (Fernandez et al., 2016). The first step, specification, determines the ontology scope, which can often be reflected by the name of the constructed ontologies, e.g., Ontology-based traffic accident risk-mapping (Wang and Wang (2011)) was for traffic accident risk-mapping and VEhicular ACcident Ontology (Barrachina et al. (2012) was for vehicular accident. In the next step of acquisition, knowledge resources are collected to build concept and relationships. In the context of road asset management, most of the resources are collected from guidance, standards, literatures, and project documents. For example, El-Gohary and El-Diraby (2010) referenced major enterprise projects (e.g., Industry Foundation Classes) and specific literature about construction management to establish an ontology model (IC-PRO-Onto) for road construction. The final step formalization defines taxonomy and lexical term definition to form a final ontology hierarchy. From knowledge to ontology instance and class, basic algorithm (e.g., description logics (DL)) and artificial intelligence techniques are widely applied in knowledge representation paradigm, and concept can be named and linked by ordered rules.

Although the modelling approaches often involve the aforementioned three steps, a few differences were noticed. Road asset management field has more social properties, thus, ontologies in this field consider more human factors, and more informal ontologies were formalized than other engineering domains (Kiritsis, 2013). In the acquisition step, some studies considered more human participation in knowledge collection. Merdan et al. (2008) proposed a multi-agent and knowledge-intensive framework based on the multi-agent system and the material-handling ontology for road agents, which highlighted the valuable opinions from agents. Other works also used focus group to collect first-hand experience to replenish the latest information in ontology framework (Yabuki et al., 2011). As for the formalization step, most studies developed ontology hierarchy directly from knowledge pool and used DL to form a formal ontology, but there were some special cases that consider semi-formal or

informal ontologies. These two types of ontologies contain less explicit information, but they can map various potential links between instances and classes by logic programming (Stephan et al., 2007). For instance, some of them are conducted to present more integrated hierarchies (e.g., Kiritsis (2013); Koukias and Kiritsis (2015)), and others emphasized detailed relationships between instances (e.g., Merdan et al. (2008)). Semi-formal and informal ontologies reduce strict definitions for class and relationship, and provide flexibility to road asset management decision making process with similar accuracy of knowledge extraction (Stephan et al., 2007).

The establishment of specific ontology models has some advantages. First, they provide common setting up steps of ontology from original information to knowledge-meta process, which can be reused by similar research in the future (Yang et al., 2019). Second, they provide a clear and intuitive description of the key elements within them (Yang et al., 2019). Another minor advantage is the unique naming of ontology models that can provide convenience for people to search, find, refer, and use them (Baldwin, 1990).

2.4.1.1 Tool

After ontologies has been formed, they require a development environment to implement, and many tools, either for research or business, have been developed. The selected papers were remarkably consistent that most of the studies used Protégé, which is a tool developed by researchers from Stanford University. It can be run on a variety of platforms, manage many standard data formats such as RDF and Turtle, and support extensions (Noy et al., 2003).

In the early application stage, Houda et al. (2010) used Protégé as a validation tool in their research to check if a new ontology improved the information management process of travel planning. After years of development, the latest version of Protégé has embedded many useful functions such as information querying, reasoning, and visualisation. Being applied in practices and research, Protégé has demonstrated its advantages including ease for the beginner, open for the secondary development, and vast popularity among researchers for ontology establishment in road assets and other AEC projects (Das et al., 2015). According to an online survey, Protégé is the most frequently used tool (Asim et al., 2018a).

Despite it is easy and interesting to use, some researchers argued that the functions of it is limited (Khondoker & Mueller, 2010). For applications in industry or

government, the functionalities of tools such as live streaming data, may require additional expansion. Thus, Yabuki et al. (2011) developed the platform HOZO to edit an ontology for roadside trees. In addition, since Protégé is based on OWL, it would encounter some problems when using external modules that were developed based on the original RDF, which will be discussed further in Section 3.3.3 (Noy et al., 2003). However, the general recommendation is that these tools must be used carefully, and users must fully understand the purpose of the target ontology.

Other tools are also used to build ontologies in road asset management areas. For instance, Merdan et al. (2008) proposed a multi-agent and knowledge-intensive ontology through Java Agent Development Environment (JADE), which is a well development platform. Outside the road asset domain, there are many tools available for developing ontologies. For instance, SWOOP is a light-weight ontology editor used in the area of biology and bio-tech, which is based on Web and easy to use for beginners. NeOn Toolkit is another tool which has an extensive set of plug-ins to support engineering ontology, especially heavy-weight projects (e.g., multi-modular ontologies and ontology integration in building projects). Possible reasons for not using these tools in road asset management includes: 1) Protégé is a mature platform; and 2) the tendency to follow existing practices.

2.4.1.2 Data representation

Data representation refers to how formalized knowledge from ontology engineering stage can be stored into computer readable information. It contains both data structure and database types used when implementing ontology (Berners-Lee et al., 2001). The resource description framework (RDF) store and Web ontology language (OWL) were found as the most widely used storage model and representation languages, while some other techniques were developed to support them.

- RDF core

The RDF was developed as a standard data model for data exchange and storage on the web (Decker et al., 2000). With the feature of being a stable data format and facilitating data integration, it was selected as the core of the ontology and semantic web (Decker et al., 2000). By presenting instances or objects as nodes that are identified by a unique resource identifier (URI) and linked by edges (relationships), such a data format makes information reusable by both humans and computer applications (Horrocks et al., 2003). Or in other word, a ‘subject-predict-object’ relation can be

defined by RDF, and this is the first step to formalize engineering information to ontology.

In other words, RDF is the basis of many developed ontologies in road asset management. Because of its long development history, many studies may have common processes and similar steps, which is convenient for researchers and engineers to share and use their ontologies (Decker et al., 2000). However, the extension of functions is limited, and users require more complex abilities to satisfy the requirements (Horrocks et al., 2003). Thus, RDF-based techniques can be used in most of conditions and road assets, and they are a good starter for any ontology study.

- RDFS and OWL

The RDF Schema (RDFS) and OWL were designed to enrich the default classes and relationships in RDF. Two of the selected studies specifically highlighted RDFS as their data representation language. RDFS was subsequently created as an evolution of the traditional RDF. It consists of various classes, comments, and elements. For instance, RDFS can develop extra subclasses for existing RDF class, which cannot be defined by default RDF-based language. The first study that used RDFS is the work conducted by Consoli et al. (2017), who provided a road maintenance RDFS with more available vocabularies. However, studies using RDF or RDFS frequently focused only on the basic framework establishment for a new domain because of its powerful class definition function. While other functions such as various relationship between class and subclass (rather than the simple definition as ‘is subclass of ’), or automatically information mapping by logics, were not considered (Haase et al., 2004).

Over twenty-two of the selected studies used OWL-based ontology in their research. OWL was developed by the World Wide Web Consortium (W3C) Web Ontology Working Group and published as a standard and recommended ontology language in 2004 (McGuinness & Van Harmelen, 2004a). It expanded the functions of RDFS to provide more embedded elements, such as complex class expressions for ontology (W3C, 2020). In the field of road asset management, some studies attempted to use OWL. For instance, Kiritsis (2013) created a closed-loop life-cycle management platform for road assets. By using OWL, it provided a wider understanding of ontology in this domain and the ability to apply ontology techniques in a complex environment. Moreover, it extended the resections function of RDFS, which became the rules for defining particular relationships. Another study was conducted by Jelokhani-Niaraki et

al. (2012), who observed that the OWL classes in spatio-temporal ontology can be reasoned, shared, and reused by the rules.

The new version of OWL, OWL2 has a series of evolution such as OWL2 Expressing Language (EL), OWL2 Query Language (QL), and OWL2 Reasoning Language (RL), for different contexts (Neumann & Weikum, 2010). Compared with OWL, the OWL2 series can be considered as a whole, reasoning algorithms for the OWL profiles, and they exhibit higher performance and are easier to implement in road asset management. For example, LécuéTallevi-Diotallevi et al. (2014) selected OWL2 EL to improve city road management ontology in the data transformation process, which achieved easy updating and flexible composition of stream operations.

In other words, OWL semantics provide more possibilities than RDF for ontology. It provides a more mature and professional vocabulary for ontology and extends functions such as reasoning for the road asset management process. OWL-based techniques allow ontology to have extra development potential and more uniform data than before. However, with the development of OWL, its compatibility with the original RDF is increasingly limited. As a result, because some of the important plug-in modules (e.g., online module) from computer science are based on RDF, OWL ontologies cannot benefit from these extra and useful functions (Motik & Horrocks, 2006).

Other storage and representation format was rarely used in road asset management domain. For instance, MySQL is an open-source relational database management system that structures data by using information in tables. Cassandra is a NoSQL database, with an aim to provide relation (e.g., graph database) other than the tabular relations used in MySQL. NoSQL databases can handle large volume of data, process high-speed querying and is friendly to plug-ins. With ontologies being increasingly established in road asset management domain, integrating ontologies in NoSQL databases is also possible(Saikaew et al., 2014).

2.4.1.3 Data Serialization

After data representation, instances, relationships and classes need to be serialized into different syntaxes for general use. Extensible markup language (XML) syntax for RDF, usually referred to as RDF/XML, is the most classic and easy-to-use format. For instance, an ontology (VEACON) created XML-based messages to provide flexible and expressive relationships between instances (Barrachina et al., 2012). However, RDF

also appeared in the coding and query process of the study, which proved that RDF and XML have a symbiotic relationship (Barrachina et al., 2012).

Additionally, other syntaxes have also been developed for RDF, such as N-Triples, JSON-LD and Turtle (Decker et al., 2000). N-triples have a simple line structure which consists of a subject, predicate and object separated by a space. Four of selected studies used this syntax, which is easy to parse and can assist compression. JSON-LD is an attempt to store new ontology using an existing format JSON. As for Turtle, it is more readable to human users, and it also has the ability to provide data stream to the management system (Beetz & Borrmann, 2018). Only two studies mentioned that they chose JSON-LD and Turtle to provide more professional RDF data in their road management process. Different syntaxes can provide more features to ontology such as easy to read by human and higher dynamic performance (Horrocks et al., 2003).

2.4.1.4 Data Querying

Data querying refers to searching required information in ontology by certain languages. In this work, extra reasoning languages implemented to improve current data interpretation are also discussed in this section.

SPARQL Protocol and RDF Query Language (SPARQL) is a query language designed and trimmed for all RDF formats. It enables schema-instance inconsistencies to be queried through the formulation of corresponding codes (Beetz & Borrmann, 2018). If an ontology is implemented using relational databases, the structured query language (SQL) languages are required for its data query, manipulation, and control. Only one study in the literature used SQL queries to derive the relationships between road objects (Hornsby & King, 2008). The limited query function narrowed the SQL application in road asset management (Hoang & Tjoa, 2006).

Some of the other languages listed in Table 4, e.g., SeRQL (Sesame RDF Query Language), are variants of SQL, while the other, e.g., OWL-to-ProLog (Programming in Logic), are specific functional languages to convert data formats. Similar to OWL, for which its variants, i.e., OWL2, OWL2 RL, and OWL2 EL, have their own characteristics and suit different use contexts, these extension languages have their own application contexts. For example, OWL2EL provides a more efficient classes definition (LécuéTallevi-Diotallevi et al., 2014). In summary, the less used languages may fit specific knowledge domains or engineering scenarios better, but also have

higher application requirements. Moreover, subsequent research in the same field may have to revert to other popular techniques such as OWL (Motik & Horrocks, 2006).

A special language is Semantic Web Rule Language (SWRL), which is a combination of OWL Description Language and the rule markup language. The extension of rules for OWL enables ontology to understand road information without extra input, which saves space and time to achieve a more efficient ontology (Zhao et al., 2015).

2.4.1.5 Accessibility

Only eighteen ontologies have been shared online for public access, which can benefit road engineers and researchers in understanding and reusing these models (Beetz & Borrmann, 2018). Some of the datasets are shared on GitHub (an online forum for sharing projects). The ontologies that used the Linked Data techniques also have their own online databases, i.e., the Linked Open Data (a cloud website). It provides a place to update and upgrade the ontologies as well as a cloud that uses Linked Data to link the nodes of different datasets (Bizer et al., 2011). Thus, information from different domains can be automatically read by computers (Parundekar et al., 2010). However, researchers from the road ontology field have seemingly not fully considered accessibility.

2.4.2 Ontology in road asset management

- Ontology modelling in various life-cycle stages

The overall process for developing and implementing ontologies in various life-cycle stages of roads follows the widely accepted ontology modelling guide, e.g., Ontology Development 101, with a few adjustments.

Planning. Ontology engineering in the planning phase focuses on hierarchy and relationship design because of the importance of information structure, such as identifying detailed decision making logic during the planning phrase (Kaza & Hopkins, 2007). Because of this specific feature, there was a heavy focus on ontology acquisition and formalization in modelling. In these two steps, detailed and compete knowledge for preparation is collected and implemented in ontology for usage. Researchers also chose language and tool that can highlight the relationships between instances, such as OWL in data representation and querying (France-Mensah & O'Brien, 2019).

Construction. Numerical and physical properties of materials and structures in the construction stage poses new requirements for the ontology modelling. Thus, ontology structure were designed classical and simple to directly reveal the construction process, and the properties were linked with instance properly for easy reading and querying by human (Zhang et al., 2015). In addition, ontology modelling in this stage also starts to consider engineers' experience. However, the acquisition step is still often implemented within the concepts and procedures from industry standards. To present as much property information as possible, the implementation always needs tools that have complex property coding and storing abilities. A notable case was the work completed by Das et al. (2015), which chose Protégé to finish a construction supply chain management ontology.

Operation & Maintenance. Since ontology has advantages in efficient information exchange and processing, most of the attempts were focused on the fields that have high data demand and liquidity, such as traffic information management or other activities in this stage (Bennett et al., 2007; Mohammad et al., 2015; Wang & Wang, 2011). Comparing with construction stages, the influence from human experience in this stage is significant (Delir Haghighi et al., 2013). In addition, more complex hierarchies were developed from ontology formalization since more supporting knowledge is required. The importance of manual works in road asset management caused the selection of semi-formal and informal ontologies in formalization step, since they required more flexibility on information management. As for ontology representation stage, because of the requirement for high-speed data exchanging process, some innovative techniques (e.g., NoSQL databases) which has outstanding performance on local device or online started to be selected. In this stage, querying and reasoning functions were also highlighted by works. Although ontology can do part of reasoning work, current computer logic may not as good as managers, and the cost would be higher as well (Delir Haghighi et al., 2013).

Road assets and their whole lifecycle management can benefit from ontology implementation by three aspects: 1) it can form abundant and critical knowledge pool for road asset management, which provides both standard and up-to-date information in a fast-changing environment; 2) it can help interpret human experience and further integrate human experience with existing knowledge, which provides solid background for informed decision making; and 3) it can improve data exchange efficiency and help

achieve real-time information reporting and responding for reduced time, cost and potential risks.

- Other ontology engineering techniques in road asset management

The implementation of the ontologies developed in various road life-cycle stages needs supporting techniques of data representation, serialization and querying to digitally represent ontology. The most commonly used data structure at this stage remains to be RDF. It achieves basic ontology functions such as searching for different types of relations in the traffic information area. Ontology needs describing languages to present full information. RDFS and OWL can enrich RDF by complex classes and direct description, which significantly improves the ability to store and present road asset information (Jelokhani-Niaraki et al., 2012; Kiritsis, 2013). The number of studies using other supporting languages, such as OWL2 EL and OWL2 DL, is limited. Specific reasons are that MySQL performs inefficiently when large volumes of data with complex data structures are involved, but it is good at structuring information and exaction efficiency, which is suitable for simple but repeating ontology building such as traffic lights system (Schram & Anderson, 2012). In contrast, databases of NoSQL can provide alternative opportunities to overcome obstacles related to scalability and flexibility, and it has attracted interest from researchers (Saikaew et al., 2014). It is suitable for areas such as traffic flow and road condition monitoring which requires large volume and streaming data.

In data serialization, most studies used the basic RDF/XML as the syntax. Other syntaxes have not been used often because the completeness of ontology framework is usually first priority, and this can be fully presented by XML. Additional advantages such as easy to read, are not as important as completeness (Asim et al., 2018a). The ultimate aim of ontology is to search relevant information for informed decision making in road asset management. Thus, querying and reasoning functions are developed to automatically extract the required information. SWRL is the mostly used querying and reasoning language with an advantage of expressing potential relationship and property. In emerging areas in road asset management filed, such as hazard analysis and smart city management, many latest techniques (e.g., SWRL and OWL2) have been implemented to improve the ability for data querying and reasoning to improve the analysis ability of ontology (LécuéTallevi-Diotallevi et al., 2014; Zhang et al., 2015).

2.5 Automatic information extraction for ontology

2.5.1 Information Extraction approaches with ontology

Information extraction (IE) is a branch of natural language processing that focuses on extracting relevant information from text sources. Information extraction (IE) is a process that involves automatically retrieving structured data or knowledge from unstructured or semi-structured textual sources. The goal of information extraction is to identify specific pieces of information, such as entities (e.g., names of people, organizations, locations), relationships between entities, and events, within a body of text. This process is often used to transform large volumes of text data into a more structured format, which can be easily processed, analyzed, and integrated into databases or other systems.

Ontology-based information extraction (OBIE) has emerged as a subfield of IE, which utilizes domain-specific ontologies to facilitate the extraction of semantic information in a particular domain or application (Wimalasuriya and Dou, 2010; Labský et al., 2008; Anantharangachar et al., 2013). OBIE has been applied in various fields, including law (Wyner and Peters, 2011), clinical reporting (Biagioli et al., 2005), and mechanical engineering (Li and Ramani, 2007).

In the construction industry, researchers have explored the extraction of information from unstructured engineering documents in a few areas. Makki et al. (2009) employed natural language processing (NLP) techniques to extract risk-cause relations for ontology updating and compliance checking, which improved the complexity of domain ontology. Zhang and El-Gohary (2013) utilized semantic modeling and NLP techniques for automated analysis and processing of regulation documents, such as code analysis. Liu and El-Gohary (2017) developed an ontology and semi-supervised conditional random fields (CRF)-based technique to extract information related to bridge conditions and maintenance activity. Tixier et al. (2016) introduced an NLP-based system capable of accurately extracting precursor and outcome information from unstructured injury reports.

2.5.1.1 ***Rule-based***

Rule-based approach involves the construction of logic rules to infer missing triples in a knowledge base based on existing ones. These rules, often in the form of

Horn logic, capture relationships between entities and can be used for reasoning. Rule-based methods require less training data compared to statistical methods and provide additional insights into the reasoning behind the identified relations (ZhongHe et al., 2020). Logic rules provide a compact and intuitive representation of knowledge facts and have been extensively used in early knowledge base studies, including expert systems. However, handcrafted rules can be subjective and prone to errors. As a result, some researchers have turned to Markov logic networks, which transform rules into graphs and apply Markov models to handle uncertainty during reasoning.

Another approach to rule-based IE involves the use of reinforcement learning or sequential models like LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit) to automatically mine rules from information within the knowledge base (Fu et al., 2016). This automated rule mining eliminates the need for manual rule development. However, when it comes to predicting missing triples, rule-based methods may face limitations, as many triples cannot be discovered solely through rules (Murphy, 2012).

It is important to note that rule-based KBC approaches have their strengths and limitations. While rules can provide valuable insights and inference capabilities, they may not capture the full complexity of the underlying data and may struggle with predicting missing triples.

2.5.1.2 ***ML-based***

Traditional machine learning (ML) models are primarily designed for structured data, such as images represented as 2D or 3D tensors. Convolutional Neural Networks (CNNs) are commonly used in image processing tasks, where filters can operate on each pixel node and scan information from its neighborhood with a fixed size and order. However, knowledge bases (KBs) contain nodes with varying neighborhood sizes and no inherent order, making it challenging to directly apply conventional CNN filters to unstructured KB data.

Graph-based models, particularly Graph Neural Networks (GNNs), offer a distinct advantage in handling KB data and extracting features. GNNs enable CNN operations to be performed on the graph structure of the KB, facilitating the extraction of meaningful features (Zhang et al., 2018b).

In the literature, there are two main types of GNNs: spectral models and spatial models. The key difference lies in how they process graphs prior to feature extraction using relevant tools, such as CNNs. Spectral models first generate the graph Laplacian

matrix, followed by eigen-decomposition of the graph based on the matrix. This projection of the graph into the Fourier domain allows for CNN operations (Kazemi et al., 2020). Spectral models have been commonly used in early studies; however, they require additional computation power for eigen-decomposition and are dependent on the specific graph's Laplacian matrix. This restricts the applicability of a model trained on one graph to another graph with different structures.

2.5.2 Information extraction models

2.5.2.1 Classical Models

Supervised and semi-supervised learning on tabular datasets often avoid using neural models due to their black-box nature and high computational requirements. Instead, when linear relationships are expected, various modeling approaches are commonly employed. In more complex scenarios, non-parametric tree-based models are preferred (Liu et al., 2019). Tools like XGBoost (Zhang & El-Gohary, 2013) and LightGBM (Kremer et al., 2020) are frequently used due to their advantages, including interpretability, the ability to handle different types of features (including null values), and good performance in both high and low data scenarios.

2.5.2.2 Deep learning models

Deep learning has also made its way into the tabular domain, although classical methods are still widely used. One example is TabNet, which utilises neural networks to emulate decision trees by emphasizing a small number of features at each layer. The attention layers in TabNet employ a sparse layer instead of the regular dot-product self-attention seen in transformer-based models, allowing only specific features to pass through.

Another approach, VIME (Clark et al., 2020), uses MLPs (multi-layer perceptrons) for pre-training based on denoising. TABERT (Raja et al., 2020), inspired by the BERT language transformer model, is trained on semi-structured test data to perform language-specific tasks. While there are several other studies that leverage tabular data, their problem settings are beyond the scope of this discussion (Tensmeyer et al., 2019).

Deep learning approaches have garnered considerable attention across diverse research and engineering fields. These methodologies offer distinct advantages over conventional ontology engineering tools. For instance, LogMap (Espinoza-Arias et al.,

2022), a well-established ontology alignment system, primarily relies on lexical similarity and lacks the capability to capture textual contexts effectively. In contrast, BERT (He et al., 2023), a system based on language models (LM), utilizes the attention mechanism inherent in the transformer architecture to generate contextual text embeddings . Consequently, it exhibits greater resilience to linguistic variations such as synonyms and polysemies. Another notable example pertains to ontology completion. Traditional systems, which rely on formal logics and/or heuristic rules, are proficient at inferring entailed knowledge, exemplified by tools like HermiT for ontology reasoning .

In terms of transformer models for general tabular data, TabTransformer (Fu et al., 2016) employs a transformer encoder to learn contextual embeddings exclusively on categorical features. Continuous features are concatenated with the embedded features and passed through an MLP. However, one limitation of this model is that continuous data does not go through the self-attention block, resulting in the loss of information about correlations between categorical and continuous features. In our proposed model, we address this issue by projecting both continuous and categorical features into a higher-dimensional embedding space and passing them through the transformer blocks. Additionally, we introduce a new type of attention that explicitly allows data points to attend to each other, leading to improved representations.

2.5.2.3 Self-supervised model

Self-supervised learning, which involves training models on unlabeled data using a pretext task, followed by fine-tuning on labeled data, has proven effective in improving model performance in language and computer vision tasks. Similar techniques have also been applied to tabular data. Several tasks commonly used for self-supervision on tabular data include masking, denoising, and replaced token detection.

Masking, or Masked Language Modeling (MLM), involves masking individual features in the data, and the model's objective is to predict or impute their values. Denoising introduces various types of noise into the data, and the model aims to recover the original values. Replaced token detection (RTD) inserts random values into a feature vector, and the model's task is to detect the locations of these replacements. These techniques have been used in previous studies on self-supervision with tabular data (Liu et al., 2019).

There are also lightweight methodologies, such as UPON Lite (De Nicola and Missikoff, 2016), which intent to place end users at the center of the process. UPON Lite (De Nicola and Missikoff, 2016) starts from the premise that ontologies can be developed by domain experts with a minimal intervention of ontology engineers. However, so far, no one has managed to prove this fact or, at least, under which conditions this premise is true.

2.6 Major gaps found from review

Based on the analysis and findings from previous sections, the major gaps in the implementation of ontology in road asset management include the lack of ontology automatic mechanism, limited options of ontology techniques, lack of online sharing of ontologies for easy access and discussion, lack of a link between ontology and other engineering techniques to obtain necessary cooperation, and limited consideration of user convenience. In addition, recommendations for further research on ontologies in certain domains are also presented. A detailed analysis for limitations and future direction is provided in the following sections.

2.6.1 Lack of specific ontology engineering approach for road asset

Based on the review from Section 3.3.1, it is found that although the general ontology development process is defined by widely accepted document and other well-known publications, some specific features of road asset management may require special attention. For instance, a more static situation (e.g., in the design and planning stage) requires a standard and formal knowledge acquisition for ontology(Das et al., 2015). On the other hand, dynamic situations (e.g., operations and maintenance stage) require efficient data storage and high-speed data exchanging. However, existing studies have not identified the unique characteristics of these life-cycle stages and formed typical ontology engineering approaches to accommodate these challenges. The lack of best practice in this domain caused sporadic knowledge collection and weak ontology integration for linked data. Other engineering fields have already piloted some wide-accepted models to improve the understanding and building of ontologies, such as TOVE and IDEON ontology for supply chain management (Grubic & Fan, 2010).

2.6.2 Lack of an automatic mechanism

Based on the review from Sections 3.3.1, 3.3.2 and 3.3.3, ontology techniques aids in the transfer of road asset management data into machine-processable information. However, the initial transition from traditional datasets into ontology data formats still requires much manual work. An automatic mechanism to capture instances, properties, and relationships is required (Gould & Cheng, 2016). Some of the research groups are trying to address this specific problem. For example, Nyulas et al. (2007) created batch imprinting plug-ins for Protégé, which can automatically convert spreadsheet information into triples. However, such attempts are insufficient because of the increasing mega data scale and structural complexity. Meanwhile, from the perspective of ontology creation, the rule-based automatic mechanism can achieve new data creation and mapping in the current ontology during the use process. In some relevant fields, such as tunnel and bridge maintenance, an automatic mechanism has been conducted for years. For instance, a semantic web-based tunnel defect diagnosis system (TDDS) was used to automatically set up the link within structural defects in underground transpiration tunnels (Hu et al., 2019). However, current new rules for automatic reasoning must be translated and manually input into the software.

In future research, an automatic rule-creation method is recommended to further reduce manual work (Hu et al., 2019). Future research can elicit and formalise both explicit and implicit rules on integrated instances and relationships via a specific rule language. The first research to use SWRL in this field was conducted by Houda et al. (2010), who used rules to automatically provide a proper travelling plan. In 2015, Zeb et al. (2015) and Zhang et al. (2015) extended the automatic creation and reasoning ontology to asset integration and analysis of site working hazards, respectively. In the next stage, machine learning techniques can be included in the rule-creation system to facilitate the semantic annotation process and reduce human intervention. Currently, relevant applications can be observed in auto-creating rules and guidelines on computers for road assets (Ali et al., 2019).

2.6.3 Difficulty in choosing suitable ontology techniques

The selection of suitable ontology techniques depends on the aim and scope of the implementation. For instance, ontology is a more efficient approach for searching the

target information in a documentational dataset, such as finding a special requirement for traffic lights in road asset management standards (Koukias et al., 2015a). However, current ontologies on road asset management have not provided sufficient reasons for RDF or OWL being the most suitable representation approach instead of other approaches (such as OWL2). Note that all of the selected studies within the review used RDF, even RDF serialisation syntax stores (e.g., RDF/XML, Notation3, N-Triples, and Turtle) as the data models. As an important conclusion from Table 4, when researchers attempt to establish an ontology, the option of tools appears to be singular. More than 80% of ontologies under road asset management (which mentioned the tool used in their research) selected Protégé.

Other ontological data models have been introduced in information management systems. Some of the latest studies in other fields have begun to use more efficient and performable storage syntaxes such as RDF* and labelled property graphs (LPGs) (Gong et al., 2018). These novel formats are graph-based models, which have advantages such as using less storage space and having faster query paths (Vicknair et al., 2010). Gong et al. (2018) compared LPGs and RDF triples models using an oilfield ontology and observed that LPGs have advantages over RDF in query efficiency for large datasets. The friendly interface, low programming requirements, and open resources are the reasons it is popular in this field (Gennari et al., 2003). However, while the homogenisation of ontology techniques may provide more opportunities for cooperation and comparison between ontologies, it also limits the opportunity of benefiting from the innovation with other approaches (Das et al., 2014).

Future studies are encouraged to focus on the latest techniques, or their latest version, based on their advantages (such as professional vocabulary and better reasoning function) in relevant fields. For example, the OWL2 language can formalise ontologies and automatically correct logic errors in the ontology mining process (De Abreu et al., 2013). Other mentioned storage approaches (e.g., MySQL and databases of NoSQL) can also be adopted in the road asset management field, depending on the specific requirements of projects (such as roadside tree management) Yabuki et al. (2011).

Another finding is that a few studies did not apply existing ontology modelling approach and created their own ontology development methods, such as IC-PRO-Onto (El-Gohary & El-Diraby, 2010). Beginning from scratch might cost researchers more

effort, but such a strategy is still recommended for future research because it can expand the current research body and provide a more detailed roadmap for subsequent research in the relevant road management or asset management domains (Grubic & Fan, 2010). However, the models should be reasonable, using best practices to avoid the risk of mistakes. The above finding is drawn based on the review from Sections 3.3.3, 3.3.4, 3.3.5 and 3.3.7.

2.6.4 Lack of sharing ontologies

Ontologies of different domains can be linked by advanced techniques (e.g., Linked Data) to form a large ontology cloud even if they have been built in their specific domains (Bizer, 2009). If a research group transferred open access information (such as traffic flow, asset management guidance, and standards) into an ontology, an option to share the ontology online for public read, reuse, and develop it is available (Carbon et al., 2009). However, as mentioned in Section 3.3.4, the majority of the selected papers have not shared their database online. A consensus in the computer science field is that researchers should provide open access to their outcomes to collect feedback and update the versions (Carbon et al., 2009). Although ontology is also a computer-based technique, not all researchers have made their ontologies publicly available. By interlinking the nodes in different datasets, even in different formats, the range of ontological information can be expanded and developed in a more friendly manner for all stakeholders and parties in a large road project (Parundekar et al., 2010). Moreover, Beetz and Borrmann (2018) made analysing the different road models from various projects feasible by linking them in an integrated ontology. Studies that have not conducted the Linked Data technique to interlink the databases can also upload the ontologies online for other purposes such as permanent storage, maintenance, and communication with users (Zaveri et al., 2013).

According to these findings and gaps, the authors suggest that a final ontology study should be published online, which can aid researchers to gain a better understanding. This step also provides a platform for the developer to upgrade and fix bugs if there are any. For instance, LécuéTucker et al. (2014) first established a traffic congestion prediction model and then opened it to the public in 2014, sooner after another study that updated and implemented the model in an actual city (LécuéTallevi-

Diotallevi et al., 2014). Moreover, the Linked Data also requires the ontology dataset to be published online to benefit the future development of relevant techniques. However, researchers may have other concerns, for example, over the secrecy of the research; thus, researchers do not need to make ontologies publicly available. Such finding is drawn based on the review from Section 3.3.6.

2.6.5 Lack of coordination with other techniques

As a novel concept, the implementation of ontology in road asset management is still relatively independent and lacks coordination with other new road asset management techniques. Although many knowledge domains and ontology tools appeared in this review, a limitation was also identified in that current ontologies lack cooperation with other latest and computer-based techniques. For instance, with the development of Industry 4.0 and Intelligent Cities, data flow from the bottom (e.g., construction sites) to top (e.g., departments of government) is required (Lom et al., 2016). Ontology, as a novel machine-based information management process, should have borne advantages in coordinating with other computer-based techniques to improve efficiency (Zhang & Yin, 2008). Surprisingly, this is not evident, and other techniques are improving in this field. In the road building and maintenance sector, building information modelling (BIM) and the industry foundation classes (IFC) data model are applied to a uniform data format and a digital information sharing platform (Angjeliu et al., 2020). Many papers on integrating BIM and classical the geographic information system (GIS) to achieve better functions such as locating the structure elements have been published (Karimi & Iordanova, 2021; Zhu et al., 2019). Moreover, with a similar development aim and history, BIM, GIS, and ontology could be coordinated by using some plug-ins (Chi et al., 2015; Niknam & Karshenas, 2017). However, attempts have rarely been made to coordinate these approaches with ontology. For instance, the ontology built based on a BIM model does not consider the construction site layout because of the incompatibility between two techniques (ontology and BIM) (Niknam & Karshenas, 2017). Thus, updating BIM and IFC information frequently to reflect the current condition and schedule in ontology is not currently possible, which would improve the accuracy of planning time (Zhang et al., 2015). Future studies could provide more opportunities for the cooperation between

ontology and AEC relevant tools, which also improves the acceptability of ontology in these industries. The above finding is drawn based on the review from Sections 3.3.5, 3.3.6 and 3.3.7.

2.6.6 Not considering human users

Although ontology is based on computers and the Internet, its final aim is to provide services to human users. Several studies have mentioned interactions with human users. For example, an ontology built for single-lane road crossing considered experience from experts and then optimized the option of drivers (Cordoba et al., 2017). However, few of them consider human users as an important and separate consideration when establishing ontologies. Similar to other concepts in computer science, there is a problem of how to effectively make the techniques practical in a friendly manner to human users (Darejeh & Singh, 2013). To achieve this, the knowledge pool on ontology must be developed from a human logic perspective. Currently. Most existing ontologies were extracted from project documents directly and missed out on the investigation involving humans (Pauwels et al., 2017). This method may cause the logic of human beings to lack in the ontology and leave the problems to the future ontology users (Darejeh & Singh, 2013). To solve this problem, an expert system can be used to collect the instances and relationships by providing the knowledge input (Cordoba et al., 2017). The event data from end users may also be considered to be regularly updated to the ontology as an adjustment.

Another reason for this gap is that some of the studies used existing software (e.g., Protégé) that have available user interfaces, while some of the studies were based on original programming software. The outlook of the ontology is also important for users from industry to accept this novel approach (Yang et al., 2019). Only a few of the studies discussed these performance scenarios, such as the visualisation function of ontology. Improving these aspects should be considered in future research.

2.7 Chapter summary

As a novel and efficient method of knowledge management, ontology provides a machine-processable technique to establish structured knowledge/information for effective management. The advantages, disadvantages, and future directions of

ontology in road asset management, which relies heavily on acquiring and using data, are attracting much research attention over the past few years. This paper aims to provide a thorough and systematic review of ontology, including its development and implementation, in road asset management. It is observed that: 1) most ontologies in road asset management target at traffic service and road assets; 2) most ontologies are not designed to support the monitoring and operation stage; and 3) RDF-based language and OWL semantics are the two most popular ontology technique. From the review, it is found that the current development and implementation of ontology in road asset management also have a few limitations, including the lack of a specific ontology engineering approach, the absence of an automatic mechanism to capture instances, properties, and relationships, limited ontology techniques and automatic information extraction approaches in this field, and the absence of sharing and linking ontologies of different domains.

3 Research method

3.1 Introduction

In this chapter, the adopted research methodology will be introduced, and consists of four sections that correspond to the four objectives, summarised in Section 1.3. The research philosophy will be introduced in Section 3.1, followed by a demonstration of the overall research design and mapping between research methods and objectives in Section 3.2. The research methods for realising Objectives 1–4 will be introduced in Sections 3.3–3.6, respectively. Finally, the chapter will be summarised in Section 3.7.

3.2 Research philosophy

In the realm of research, paradigms serve as the roots and stances that underlie the basic beliefs that guide how a researcher understands or conducts his/her work (Killam, 2013) (Guba & Lincoln, 1994) provided a comprehensive definition of a paradigm, describing it as "a basic belief system based on ontological, epistemological, and methodological assumptions". These elements are intrinsically linked and interdependent under a paradigm, and as such, different types of research are guided by different paradigms (Hood & Wilson, 2001). Thus, knowing the underlying philosophical beliefs is essential for research.

In the context of research, ontology refers to the theory of being or in existence, specifically dealing with the nature of reality (Aliyu et al., 2015). The Merriam-Webster dictionary (2020) provides a more comprehensive definition of ontology, describing it as a branch of metaphysics that focuses on the nature of being and the relationships that exist among them. An individual researcher's ontology reflects their personal beliefs about what is real or true. In general, there are two distinct and contrasting types of ontology: realism and relativism. Realists maintain that there is only one objective reality that can be discovered and measured through objective means, while relativists contend that reality cannot be found, but rather is constructed through people's experiences, resulting in the existence of multiple equally valid realities (Killam, 2013).

Epistemology, or the theory of knowledge, is concerned with the relationship between researchers and the objects of their study (Aliyu et al., 2015). Within this framework, two contrasting epistemological positions can be identified: objectivism

and subjectivism. The choice of epistemological belief is often informed by the researcher's underlying ontology (Killam, 2013). For example, a realist who subscribes to an ontology of objective reality would argue that "truth" about the world exists independently and can only be discovered through objective measures. On the other hand, a relativist who subscribes to an ontology of multiple subjective realities would argue that "truth" can vary depending on an individual's experiences and context. In this view, constructing meaning or reality through social interaction is more important than searching for an objective "truth." The choice of epistemological belief has significant implications for how research is conducted, as it shapes the researcher's perspective on the role of subjectivity in the acquisition of knowledge.

Methodology is a systematic approach to acquiring knowledge about the world (Killam, 2013) and is heavily influenced by a researcher's ontology and epistemology. Researchers who hold an objective view of reality often prefer quantitative methodologies, which involve using measurable methods, such as experiments and questionnaires to collect and analyse data (Aliyu et al., 2015). On the other hand, researchers who hold a subjective view of reality tend to prefer qualitative methodologies, such as in-depth interviews, to understand people's experiences. However, both quantitative and qualitative methodologies have their respective strengths and weaknesses. Therefore, researchers, such as Steckler et al. (1992) and Kelle and Erzberger (2004) recommend integrating these two methodologies to leverage the benefits of both.

Axiology, which is mainly concerned with ethical issues, is an important aspect of research methodology (Hood & Wilson, 2001). It is the theory that deals with the nature, types, and standards of value judgments, especially in morality, according to the Merriam-Webster dictionary (2020). Positivism is the most traditional paradigm, and as the thinking of the scientists evolves, alternative paradigms emerge (Koschmann, 1996).

This research aims to improve information integration and extraction, and add automated knowledge mapping; it verifies the proposed approaches in RAM projects. The automatic knowledge extraction and mapping model is an all-quantitative model. Additionally, the research proposes the hypothesis that using ontology can improve the performance of RAM, which should be tested in experiments. Thus, the research is closer to deductive and quantitative research, and is based on objectivism epistemology

and realism ontology. Moreover, some subjective opinions from domain experts were also utilised to evaluate the ontology. Thus, it can be argued that this research is a mixed study and belongs to the post-positivism paradigm.

3.3 Overview of the proposed method

Figure 3-1 illustrates an overview of the adopted research methods. A critical review method was adopted for the literature review for ontology and the relevant knowledge in engineering and RAM field (Objective 1), as presented in Chapter 2. After a comprehensive review, the second section pertains to the progress of a comparison between two most-widely used data models: RDF and LPGs, and aims to find the most suitable approach to integrate information in an ontology. The creation of an ontology is focused on in Objective 3, where ontology takes in text sentences as inputs and generates entities and relations, also known as triples, as outputs. The last section refers to the automatic information extraction approach developed in Objective 4, where the inputs are tablular information in RAM materials, and the outputs are new triples in ontology by natural language processing (NLP)

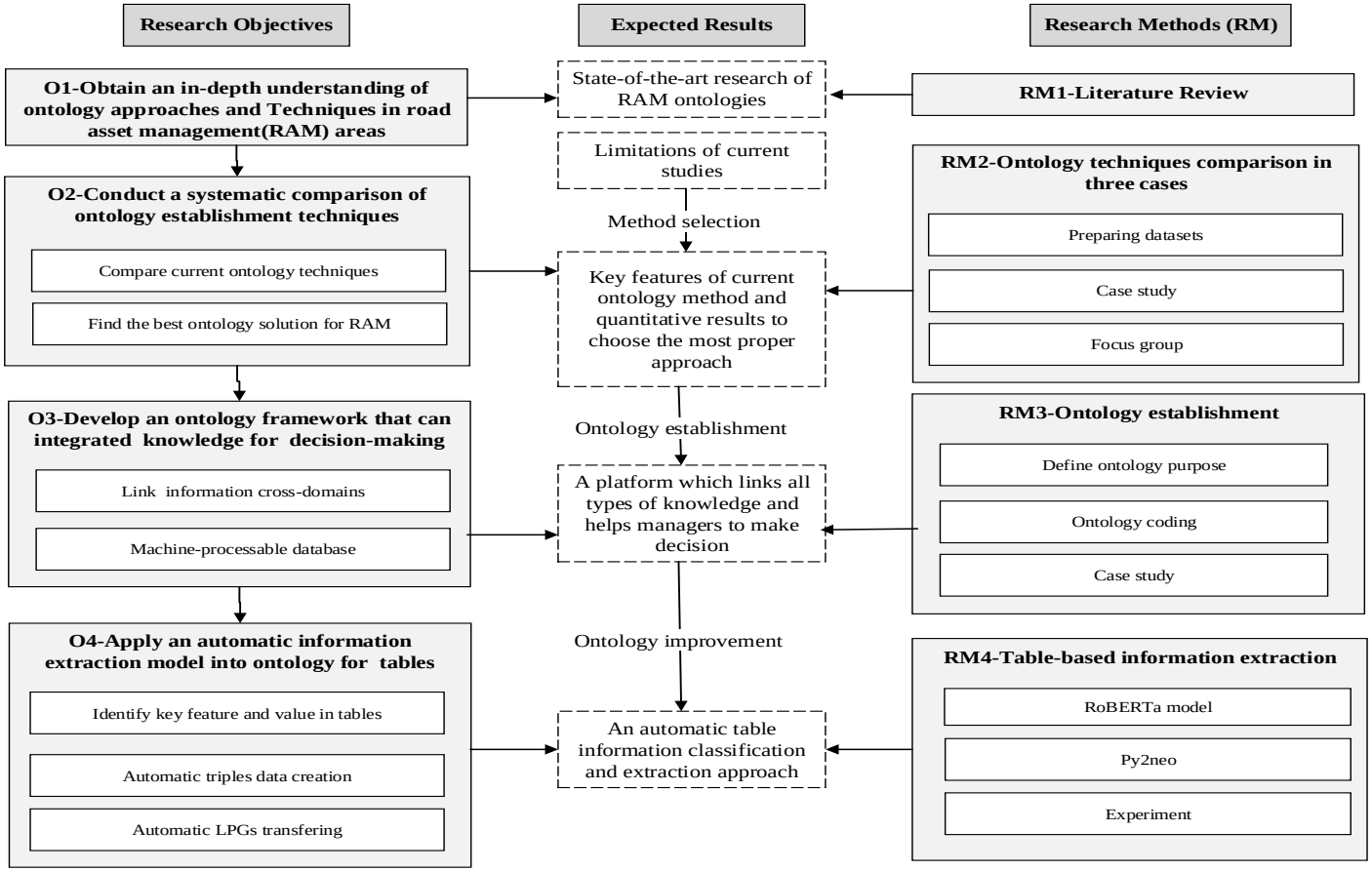


Figure 3-1 Overview of adopted research methods

3.4 Systematic literature review (SLR) methodology (Objective 1)

A systematic literature approach proposed by Yang et al. (2019) was used in this study. The review process involved paper selection (filtering), quantitative analysis, qualitative analysis, and result discussion. Such a method has also been adopted by other similar review studies (Kiritsis, 2013; Pauwels et al., 2017; Yang et al., 2019).

The scope of the review was confined to the development and implementation of ontology in road asset management. In total, eight steps were adopted during the review process, and a detailed explanation of the review process is shown in Figure 3-2. Priority was given to the Web of Science database owing to its wide coverage and high quality, while Scopus, IEEE Xplore, and Google Scholar were also considered (Jiang & Wu, 2019). The searching strings were defined based on previous studies in the same research field, e.g., Yang et al. (2019), Kiritsis (2013); Le and Jeong (2016). Based on these studies, 'semantic' or 'semantic web' and 'Linked data' are the most relevant keywords for ontology, while 'traffic asset' is a typical substitute word for 'road asset'. Thus, the final search strings were set as ('ontology' OR 'semantic' OR 'Linked Data') AND ('road' OR 'road asset' OR 'traffic asset'). Note that conference papers from the computer science field were also considered in this study because conferences are also an important means of communicating quality research on ontology in the computer science field (Freyne et al., 2010).

After collecting more than 500 papers in Step 1, a manual process was adopted to filter papers by examining their titles, keywords, and abstracts. Only peer-reviewed journal papers, conference papers from leading conferences, and other papers that use ontology in road asset management were retained. After filtering, 97 publications were identified.

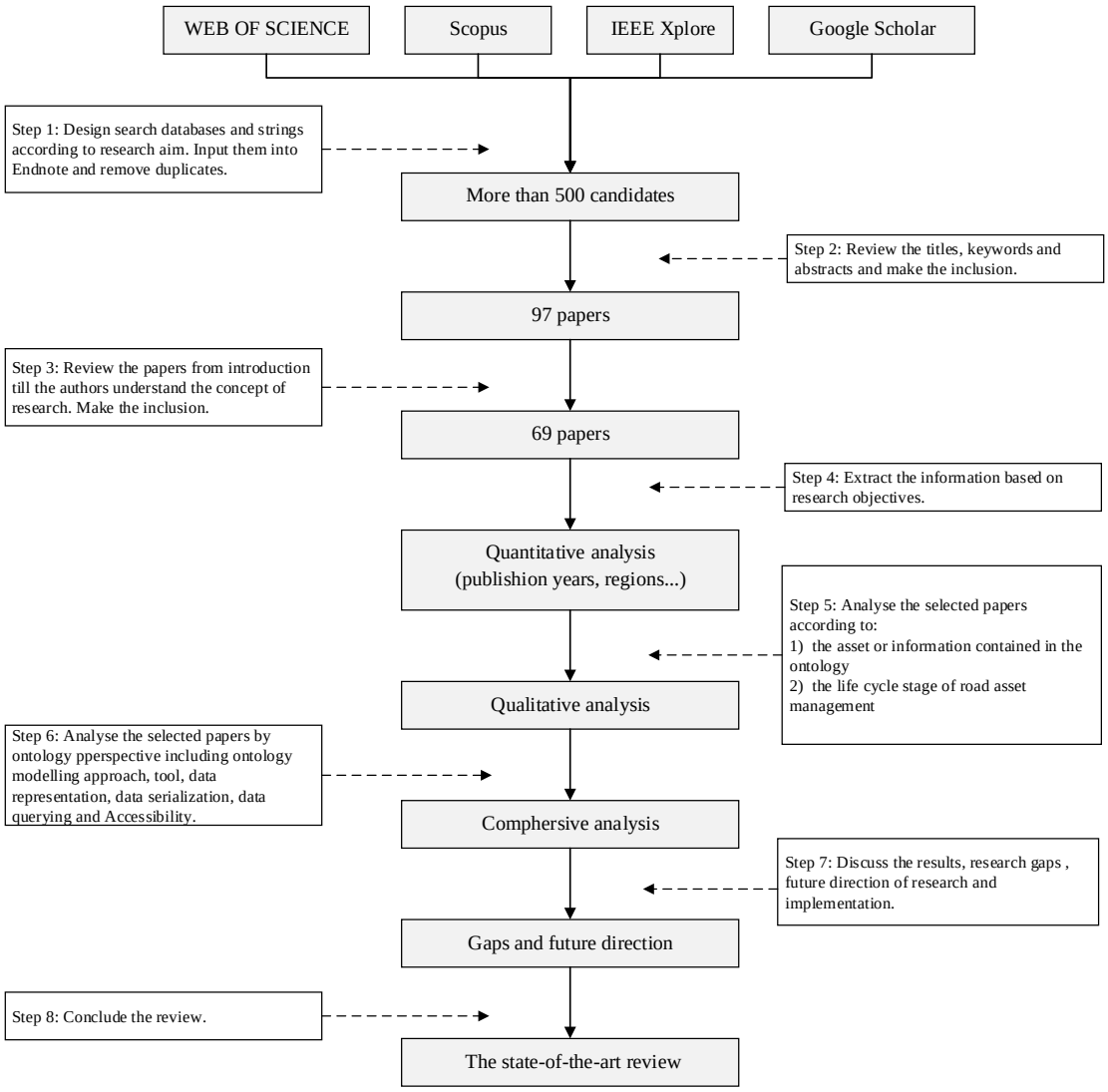


Figure 3-2 The process of the systematic literature review.

Note that the term ‘road’ in this review only refers to surface pavements and objects that move on them, such as vehicles (Park et al., 2016).

In Step 3, a further filtering process was conducted manually. Only papers closely related to the development and implementation of ontology in road asset management were included. After filtering, 69 papers were identified and included in the analysis.

3.4.1 Analysis codes

The 69 selected papers were coded and analysed through codes in Table 4, which were developed from Li et al. (2019) and Yang et al. (2019). These codes can be categorised into three groups. The first group is related to the publications, including

year, author, journal/conference title, and country/region. The second group is related to the implementation domains, focusing on the asset type and life-cycle stage where the ontology is implemented. Asset type represents what asset types have been targeted at by using ontology techniques, and life-cycle stage represents the life cycle stages of road assets where the ontology techniques are applied. The third group of codes focuses on the ontology techniques, which include the ontology modelling approach, tool, data representation, serialization, querying and accessibility. These codes consist of all necessary processes from knowledge formalization to ontology presentation. In addition, the gaps identified during the review process are analysed, and future directions are discussed.

Table 4 Codes for the review

Area	Code	Description
Publication-related information	Year	Year of publication
	Author	Authors
	Publication venue	Journal/conference at which the paper was published
	Location	Country/region where the study originated
Implementation domains	Asset type	Asset type of ontology implementation
	Life-cycle stage	Life-cycle stage of the ontology implementation
Ontology techniques	Modelling approach	Ontology knowledge collection and formalization
	Tool	Tool and platform used to create ontologies
	Data representation	Data model and description language
	Data serialization	Serializing data into machine interpretable syntax
	Data querying	Information searching and reasoning
	Accessibility	Whether or not the development has open access to readers
	Limitation	Limitation of current ontology implementation

3.5 Conduct a systematic comparison of ontology establishment techniques (Objective 2)

The aim of this research is to perform a critical comparison between two graph technologies for ontology: RDF and LPG. Experiments were chosen as the main research method, and they have been widely used in similar comparison works, which have shown the ability of this method to give clear results (Schmidt et al., 2008). Those

experiments will be conducted in an ontology that representing a real-world bridge maintenance project. To achieve a more comprehensive comparison, other supporting methods, such as a literature review and focus group evaluation, are also chosen as additional qualitative comparison approaches. In this research, comparisons that apply the RDF data model are referred to as RDF-based approaches, while those that apply LPGs are referred to as LPG-based approaches.

3.5.1 Comparison procedure

A general overview of the comparison procedure is presented in Figure 3-3. The comparison benchmarks are selected from the literature, which has introduced existing research gaps and research focus areas. Then, these variables were tested in a series of experiments.

Firstly, a proper ontology was selected as the main experiment context. Three different sizes of ontology datasets were obtained from the ontology to test the performance of RDF and LPG. The five benchmarks identified from the comprehensive literature review were used as the evaluation criteria. Based on the features of these benchmarks, both quantitative and qualitive analyses were performed since some of the benchmarks (e.g., visualisation) are hard to measure using certain parameters (De Abreu et al., 2013). The first three benchmarks, i.e., data density, query efficiency and reasoning function, were compared quantitatively. In contrast, visual behaviour is a relatively subjective ability that needs a subjective evaluation method to achieve the aim of comparison. Thus, a review and a small focus group that provided qualitive analyses were implemented to assist the experiment. The detailed comparison approaches are explained later.

In addition, to achieve the above steps, various tools and plug-ins that provide different functions must be used. Protégé 5.5.0 (Noy et al., 2003), Apache Jena 3.16 (Schmidt et al., 2008) and Neo4j 4.3.2 (Baton & Van Bruggen, 2017) were chosen as the main tools, and other useful tools were introduced when used.

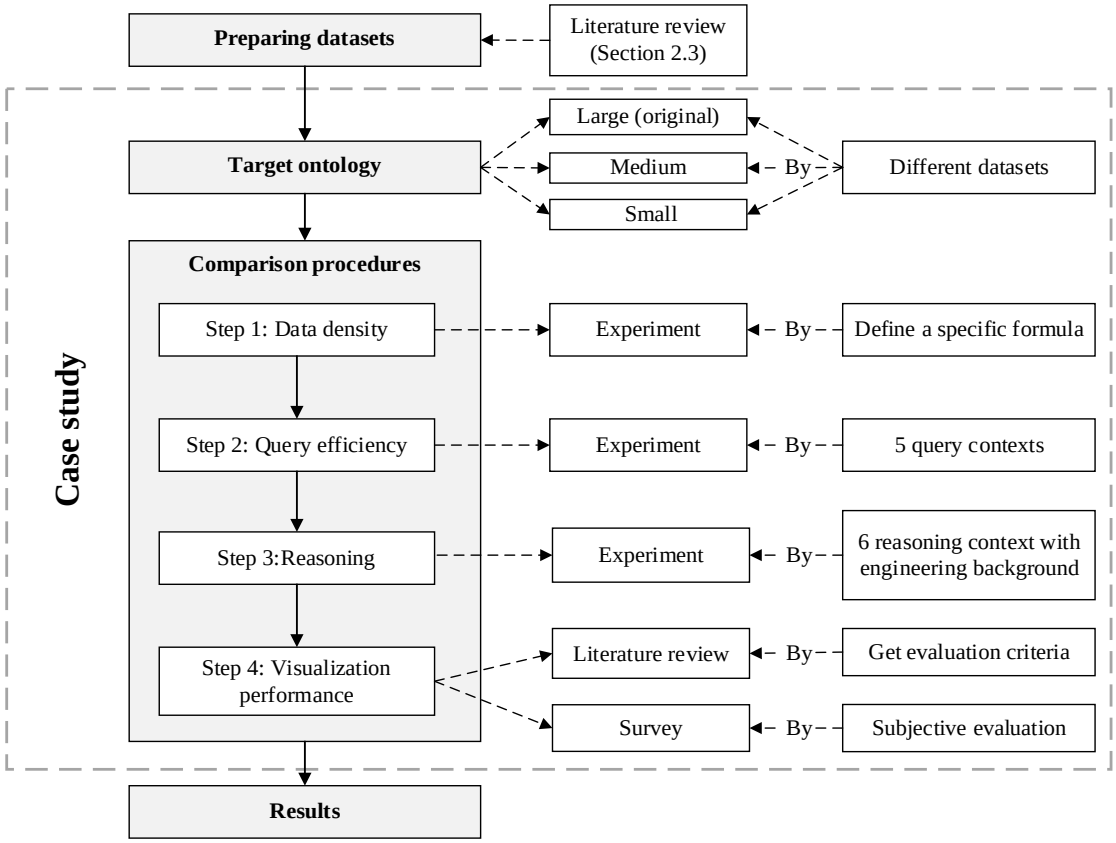


Figure 3-3 Overview of the research steps.

3.5.2 Data collection

Comparative experiments within the realm of academic research invariably necessitate the utilization of one or more ontologies as a fundamental component of the experimental framework. These datasets are not only expected to encompass a representation of logical information culled from real-world domains, including domains such as traffic systems or biological entities, but they must also exhibit a range of different scales. Selecting an ontology of a reasonable size is of paramount importance, as it plays a pivotal role in ensuring that the experiments remain both manageable and efficient. The processing of a substantial number of triples, a task which demands formidable computational resources, engenders not only considerable computational costs but also extended time requirements.

However, ascertaining the optimal number of triples or vertices that an ontology should comprise to facilitate a meaningful and insightful comparison is a formidable challenge. To strike an appropriate balance that ensures feasibility and enables the

elucidation of key distinctions, it is essential to turn to pertinent comparative studies for guidance. These comparative works offer invaluable reference points, enabling the assessment of the size and characteristics of ontological datasets. It is worth noting that different research endeavors may employ distinct metrics for quantifying the size of the data, encompassing measures such as the count of triples in Resource Description Framework (RDF), the tally of vertices in Labeled Property Graphs (LPGs), or the assessment of data volume in bytes. These diverse indicators serve as useful benchmarks, underpinning the establishment of a rational foundation for configuring the size and scope of ontological datasets to be deployed in the context of comparative investigations.

Accordingly, a set of criteria has been established for the selection of ontologies. A suitable ontology must satisfy the following prerequisites:

1. It must have been actively maintained in the most recent six-month period.
2. It should have been developed using readily available ontology construction tools.
3. It needs to be applicable within the Architecture, Engineering, and Construction (AEC) industry context.

To maintain uniformity and consistency throughout the experiments, a well-established and mature conversion tool, Neosemantics (N10s), will be employed for the conversion of datasets. N10s has been in continuous development and practical use for many years, serving as a trusted and widely employed tool in numerous research endeavors and practical applications.

3.5.3 Case ontology description

To construct this ontology, an array of information resources was tapped, encompassing standards, manuals, and project documents. These project documents included case reports, work plans, and quality evaluation reports. After amassing an ample amount of knowledge and data, the ontology was structured, beginning with the delineation of classes and subclasses (or hierarchical levels). In this hierarchical structure, the ontology is divided into three primary classes, namely bridge components, project participants, and rehabilitation tasks. Subsequently, detailed information and values were populated within each level of the hierarchy, while additional edges were

introduced to convey the relationships between the various conceptual components. The culmination of this process resulted in the comprehensive storage and presentation of all pertinent information in a computer-readable dataset.

Following the selection of the ontologies earmarked for testing, the subsequent phase entails importing the data into both RDF and LPG (Labeled Property Graph) tools to facilitate their utilization. Given that the ontology was initially conceived and edited using OWL (Web Ontology Language), it can be seamlessly integrated into RDF-based tools. However, it's important to note that RDF information cannot be directly represented in LPG-based tools and necessitates conversion into LPG format.

For this data transformation, the Neosemantics (N10s) plug-in is deployed. N10s excels in its capacity to efficiently import RDF data into LPGs and export it back into RDF format. This tool has undergone rigorous testing across numerous projects and has consistently demonstrated its competence in preserving data integrity. Consequently, the converted datasets are meticulously compared with the originals. This comparison scrutinizes each concept, relationship, and property information to ensure the avoidance of any inadvertent information loss or omissions.

3.5.4 Indicators for comparison

1) Data density

The first benchmark required all three datasets to be tested. The method of storing the maximum amount of data in a limited space is one of the most important factors for measuring a data model or format (Anderson et al., 2009). In this study, this ability was tested in a set of experiments, and assessing the density of different data structures can use the equation developed by De Abreu et al. (2013). However, it did not include the property data embedded in the LPGs, which is also a special structural feature. Thus, an update of the formula by these authors was defined as follows:

$$\rho = \frac{r + p}{s \cdot n(n-1)}$$

where

n represents the number of vertexes,

p represents the number of properties,

s represents the storage size read from disk,

r represents the number of relationships.

2) Query efficiency

Query efficiency represents the basic information processing capability of models and represents the main improvement from manual work to computer-based methods (Constantinov et al., 2015). By using SPARQL (SPARQL Protocol and RDF Query Language) in Jena, the time needed to query certain information can be determined. In Neo4j, the declarative graph query language Cypher, which is used to design LPGs, was used to measure the time needed for the same queries.

3) Reasoning

Many ontology reasoning tasks are finished by performing inferences, which allows for the creation of new knowledge from existing information without manual work. By inferencing/reasoning, the users can understand the process of obtaining the information from a database that is not explicitly stored. The majority of RDF reasoning methods are rule-based reasoning, which sets logic rules based on the real world and then obtains the results. OWL description logics (DL) use an object-oriented modelling paradigm to describe information and provide an automatic deduction process (Wang et al., 2004). For example, 'A-B' and 'B-C' present two linear relationships of the same type, and DL could then automatically create the relationship 'A-C' as the reasoning result. In addition, the Rule Markup Language (ML) and the Semantic Web Rule Languages (SWRL) are designed for various and complex requirements, such as those in the engineering, business and biology fields (Arndt et al., 2017). Using the improved languages, the RDF-based approach can support a more complex reasoning process, such as semantic reasoning. In this case, the ontology can reason out the delayed works based on SWRL (Arndt et al., 2017).

On the other hand, Neo4j can load and write reasoning results in the RDF (Baton & Van Bruggen, 2017). However, LPG development has been limited in terms of fully fledged stores or dedicated reasoning engines (Baton & Van Bruggen, 2017). A similar situation is observed in which LPGs present limited development in this field as dynamic updating functions. For instance, the 'A-B-C' case can also be reasoned in Neo4j by Cypher, but it needs extra plug-in (e.g., N10s) to start reasoning. Plug-ins provide simple reasoning functions, such as identifying one vertex that belongs to one class and has default properties (rule-based). Another available plug-in is called

GraphScale, which empowers Neo4j with scalable OWL reasoning (Arndt et al., 2017). The approach is based on an abstraction refinement technique that builds a compact representation of the graph that is suitable for in-memory reasoning.

This function was tested in six simulated scenarios developed from the selected ontology, and the scenarios also contained enough engineering background to support a project. How well the RDF and LPG approaches performed in reasoning extra information as required was assessed. Scenarios, codes, tools, and plug-ins used by the models are listed in Table 6. It should be noted that LPG, which is a relatively new graph technology that was not primarily developed for reasoning, may have intrinsic limitations on this benchmark (Gong et al., 2018). However, a comparison is still necessary to obtain experimental results.

4)Visualization performance

Ontology data models not only produce data in machine-readable formats but can also provide visualized information for human users. Both RDF and LPG methods require plug-ins for visualizing information. Data visualization has become a hot topic in information presentation, and studies have focused on the visualization performance of ontologies. For instance, Dudáš et al. (2018) reviewed available ontology visualization methods and introduced certain supporting functions. These works provided a subjective method for performing comparisons.

However, different RDF or LPG tools have particular advantages based on their theoretical core of data structure. Thus, Protégé software for building and maintaining ontologies using RDF was chosen for these benchmarks. It is also one of the most widely accepted and used tools for ontologies (Asim et al., 2018b). By using OWL and OntoGraf plug-ins, ontologies can be presented from text to flexible graphs. However, the visualization function of Neo4j is one of the outstanding features (Donkers et al., 2020). The original data were automatically stored in graphs, and detailed properties can be read from the inferences.

The default plug-ins OntoGraf in Protégé and Neo4j Browser were used for the comparison, and there are two reasons for this decision. Firstly, although there are many visualization plug-ins for both Protégé and Neo4j (e.g., OntoViz, TGVizTab, and Bloom), most of the ontologies in previous and current studies still use default tools to visualize data (Akrivi et al., 2006). Secondly, the default tools have the same core, which both implement a 2-dimensional vertex-link visualization method that visualizes

ontologies as a vertex network (Dudáš et al., 2018). This will minimise the risk that the performance may be affected by tools with different cores. Moreover, they can display vertexes under certain classes, edges between vertexes and other visualized information, such as properties and classes. However, LPGs were designed to be visual in nature using graphs to display information, and this method can easily model the visualized information.

5) Review

As a relatively subjective function, a review work needs to collect information on the evaluation. The scope of the review was confined to the development and implementation of ontology visualization methods for the RDF and LPGs. Combined with other well-accepted data visualization evaluation methods, a basic functional and availability comparison has been performed to present the features of different data models.

6) Focus group

Another additional method was the use of a focus group evaluation for visualization comparison. A focus group will be used to verify the characteristics and comparison of the two methods. A total of 14 scoring items will be used in the interview, which are summarized from other comparative experiments in the literature.

In the subsequent phase, the determination of the number of participants and their selection criteria is imperative. It is recommended that a focus group comprising 5-12 participants strikes a balance between the depth and breadth of data collection (El-Sabek & McCabe, 2018). Accordingly, this study invited ten domain experts, chosen based on the following criteria: 1) possessing extensive work experience (i.e., over 8 years) in bridge maintenance, 2) active involvement in at least one major bridge rehabilitation project within the past five years, and 3) representing diverse backgrounds to encompass various project stages and perspectives. The profiles of the ten participants are outlined in Table 3-2. The cohort comprises project-level stakeholders (e.g., the owner, contractor, designer, maintenance team, and supply company), as well as external entities primarily comprising relevant authorities such as Department of Transportation (DoTs) and municipal bureaus. Consequently, it can be contended that these experts are well-positioned to offer comprehensive and invaluable insights on the pertinent topics of inquiry.

Eight participants were invited from an expert pool who had years of road asset management /construction experience, and academics and industry members were invited. Although the number of participants was small, the evaluation process was still creditable (Toner, 2009).

Table 5 Profiles of the focus group participants

Expert No.	Years of experience	Background
1	7	Industry
2	7	Industry
3	7	Industry
4	10	Industry
5	10	Industry
6	10	Academia
7	11	Academia
8	11	Academia

3.6 Ontology development (Objective 3)

The development of the EIAO framework followed the most widely accepted practices outlined by Ontology Development 101 and the procedure suggested by Noy and McGuinness (2001). The development procedure was adjusted based on the features of EIA in contrast to the other ontology frameworks. As shown in Figure 3-4, it takes seven main steps to establish an ontology, which are 1) define the scope of ontology; 2) consider reusing existing ontologies; 3) acquire knowledge of ontology; 4) define ontology structure; 5) define ontology establishing an environment and data model; 6) establish ontology, and 7) validate and improve. The final outcome would be a machine-readable ontology for the EIA process, and some real-world scenarios will be used to demonstrate the feasibility of the development.

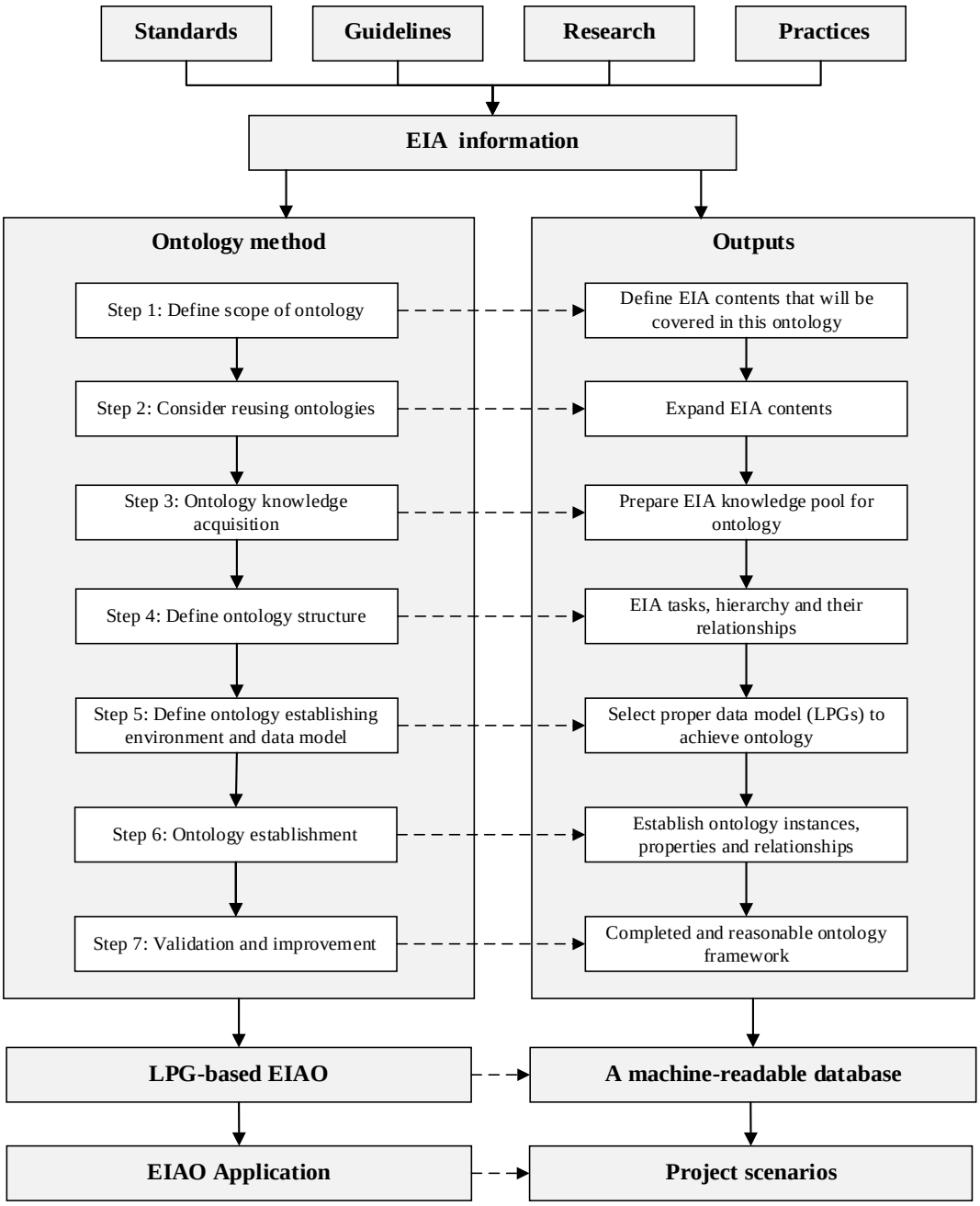


Figure 3-4 EIAO method and outputs

3.6.1 Define the scope of ontology

The scope of EIAO covers EIA planning, operation and maintenance activities in RAM. The main components include identifying environmental hazards, environmental impact, and EIA monitoring actions (audit). In these components, hazards and impact information can be treated as reasons to conduct an EIA audit, and audit actions are the most important and expensive steps in the whole process (Morgan, 2012). Thus, this EIAO will first focus on integrating auditing activities, and also combine the necessary

hazard and impact knowledge in the knowledge structure as supporting information. It intends to transfer the current EIA knowledge and activities into a computer-processable database, thereby providing digital information searching functions.

3.6.2 Consider reusing existing ontologies

When establishing a new ontology, researchers would consider reusing existing ontologies to avoid repetitive work and unintended errors. The only relevant study found from the literature in the current context is a work conducted by Garrido and Requena (2011), which is a knowledge mobilisation ontology for EIA. However, it only covers activities for identifying environmental impact hazards. Some ontologies, including common taxonomies of road management and construction projects (e.g., Das et al. (2015); Zhang et al. (2015)), also fit into the EIA development scope and hence have been adopted as supporting materials for the following steps.

3.6.3 Acquire knowledge of ontology

Once the scope is clearly defined, the subsequent pivotal step involves establishing a comprehensive knowledge repository for an EIA in the context of road infrastructure (EIAO for roads). The primary sources of knowledge pertinent to EIAO for roads encompass a wide array of references, as follows:

Standards: These include internationally recognised standards, such as “ISO 55001 for Asset Management” and “ISO 12006 for Building Construction”.

Environmental authorities: Reputable organisations and authorities that specialise in environmental matters, such as the Organisation for Economic Co-operation and Development.

RAM and research institutions: Institutions, such as Austroads and the Environmental Protection Authority, are instrumental in generating knowledge and conducting research in the field.

Books: Authoritative texts and books serve as valuable sources of knowledge. For instance, "Methods of Environmental Impact Assessment" by Morris and Therivel (2001), and "Environmental Impact Assessment: Theory and Practice" by, Wathern (2013) are pivotal in providing insights.

Research papers: Pertinent research papers, studies, and academic works also contribute to the knowledge pool.

This broad array of knowledge sources forms the foundational pillars of the knowledge pool for EIAO, thus ensuring a well-rounded and comprehensive resource for road environmental impact assessments.

3.6.4 Define ontology structure

3.6.4.1 Define EIA ontology structure

From a technical perspective, EIA is fundamentally an analytical process aimed at identifying cause-and-effect relationships. Its purpose extends to the quantification, evaluation, and mitigation of the environmental consequences resulting from a given project, as articulated by Gómez-Pérez (2001). This foundational definition serves as the basis for the extraction and justification of key concepts within the ontology, which include:

Environmental impacts: These encompass tangible and intangible effects that a project may exert on the environment, both positive and negative.

Environmental elements prone to impacts: These refer to various components of the environment that are susceptible to being influenced by the project activities.

Industrial activities: This category pertains to specific actions and processes conducted within an industrial context, often contributing to environmental impacts.

Substances or contaminant elements: These are materials, compounds, or agents having the potential to contaminate or otherwise affect the environment.

Human actions with impact potential: These encompass a range of human behaviours and actions that possess the capacity to generate environmental impacts.

Environmental indicators or impact measurement units: This category involves tools and metrics used to quantify, measure, and assess the environmental impacts under consideration.

Impact assessment: This encapsulates the systematic evaluation and appraisal of the environmental consequences engendered by a project, and is a pivotal component of the EIA process.

The core concepts and relationships are visually represented in Figure 3-5. These elements have been directly derived from the earlier EIA definition. The figure exclusively showcases these fundamental concepts and relationships, as they constitute the foundational framework. It is worth noting that further additional concepts exist in the sub-levels of the ontology hierarchy.

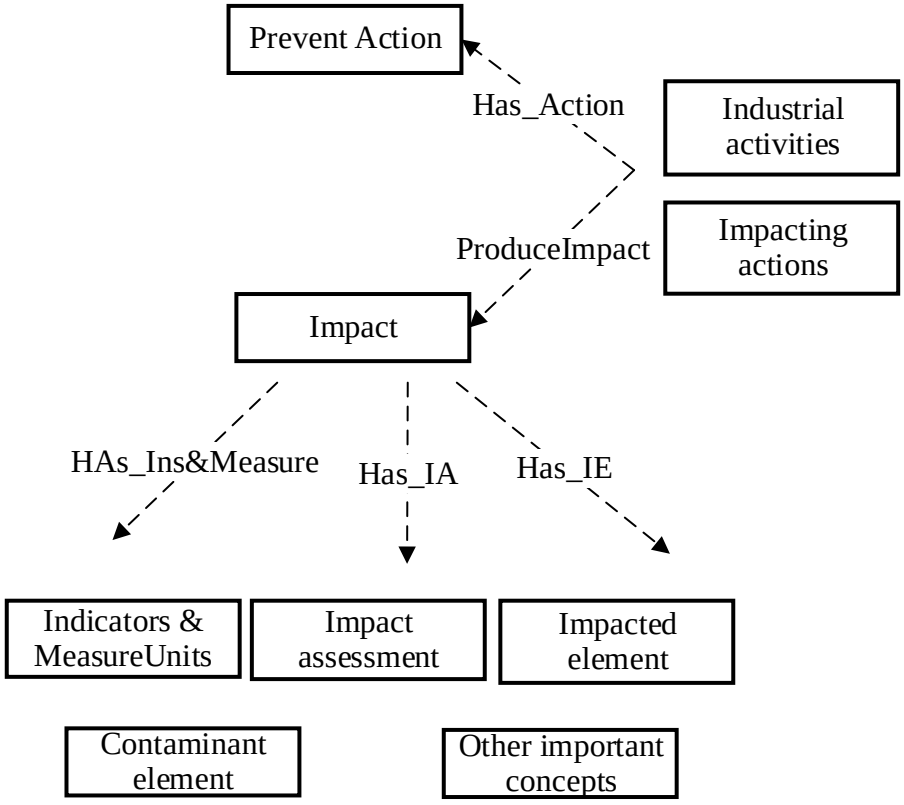


Figure 3-5 Main concepts and relationships of EIA

These relationships play a significant role in enhancing the KB and can be effectively employed to execute queries within the ontology, particularly in the context of reasoning tasks. For instance, they facilitate inquiries regarding the environmental impacts stemming from a specific industrial activity or the environmental indicators utilised for the evaluation of EIA activity.

The subsequent subsections provide comprehensive explanations of the concepts presented in Figure 3-6. It is essential to note that the concepts of "impact," "impacting

action," and "impacted element" receive more detailed elucidation, as they are deemed the most critical concepts within the ontology.

3.6.4.2 Define specific EIA decision-making by ontology

Once a comprehensive structured EIA knowledge is acquired, the next step involves utilising the ontology framework to assist decision makers in making informed decisions and judgments. This methodical EIAO provides a structured and comprehensive approach for addressing and mitigating environmental impacts and potential hazards associated with road infrastructure projects. The framework is unwavering in its commitment to ensuring strict adherence to environmental regulations and standards. The visual representation of Figure 3-8 effectively encapsulates this intricate EIA process.

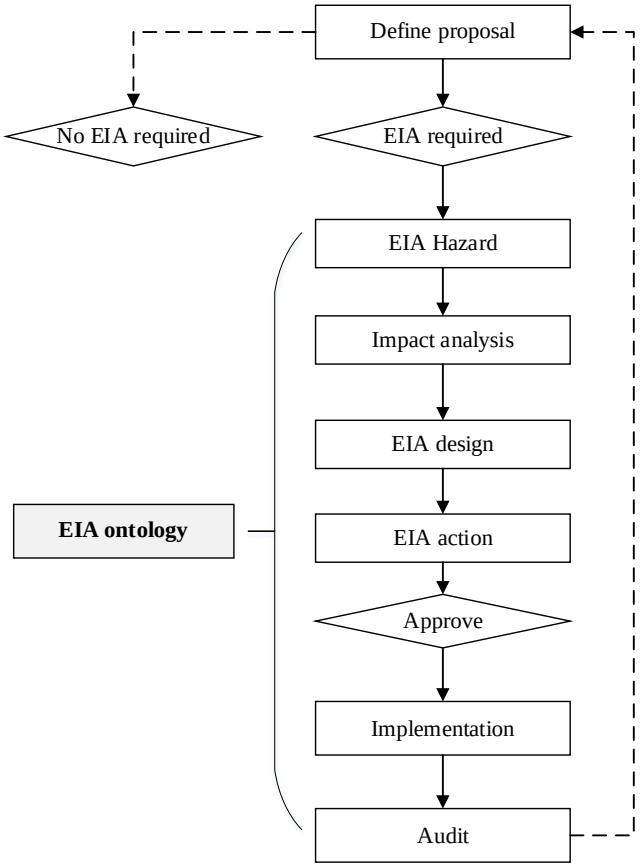


Figure 3-6 EIA system flowchart

3.6.5 Define ontology establishing environment and data model

Many languages and data models use different structures, such as RDF, Web Ontology Language (OWL) and LPGs. The RDF format is one of the most popular

standards for establishing an ontology, and is based on the expression *subject–predicate–object* (also known as a *triple*) that represents the relationships between instances. However, it also has limitations, such as using more storage space, limited original operational logic and lack of compatibility with other programs (Gong et al., 2018). In this research, novel LPGs has been used as an ontology establishment model, and the popular tool, Neo4j has been adopted as the implementation environment.

LPGs are a multiple labelled graph model, which a group of Swedish computer engineers developed after RDFs were developed (Anikin et al., 2019). LPGs present the information by nodes, link those nodes by edges, and enrich them via embedded properties. Using graph-based structures, those objects and relation types in an RDF can be added to various properties more powerfully (De Abreu et al., 2013).

In LPGs, all features can be presented in one instance (node), and the link between two nodes can be named based on the relationship (edge). As an LPG can use fewer links to represent the same amount of information, it can significantly reduce the querying time and efficiently deal with complex relationships (Gong et al., 2018).

3.6.6 Establish ontology

This phase is dedicated to the transformation of knowledge into the LPG format. Within the context of the EIA, data can be categorised into four distinct types: 1) drawing data —this category encompasses graphical data, such as inspection figures; 2) tabular data — it comprises structured data presented in tabular formats, often found in spreadsheets; 3) raw digital documents — this type includes unprocessed digital documents, such as those in Adobe PDF and Microsoft Word formats; 4) other paper-based materials — this category extends to various physical records and materials, including maps and paper-based documents.

To facilitate the conversion of these data types into the LPG format, a four-dimensional data model has been employed. These four dimensions correspond to the definition of ontology instances (In), classes (Cl), properties (Pr), and relationships (R).

3.6.7 Validation and improvement

The EIAO has been implemented in a standard Neo4j environment, using Cypher as a query language (Zhang et al., 2015). A case study was conducted to test the

development, using two main functionalities: searching and reasoning information, the most accepted factors when assessing an ontology (Scholer et al., 2002). Table 3 lists some of the query and reasoning functions that EIAO can provide.

Ontology validation involves assessing both semantic and syntactic correctness. Semantic validation typically includes several methods, such as posing competency questions, consulting domain experts, and comparing the new ontology with the existing ones. In the case of the current EIAO, as there are no similar ontologies, only the first two validation methods were applied.

Competency questions serve as a straightforward approach for self-checking the semantics of ontologies. These questions should align with the inquiries outlined in Step 1 of ontology development and cover ontological classes, instances, and relationships. Examples of such questions include 1) How many sub-classes belong to a specific constraint class? 2) What are the relationships among certain entities? 3) What nodes are associated with a particular task or procedure? 4) Which tasks or procedures have experienced an "out of control" state, and to what extent? 5) Who is the top-performing participant in terms of constraint removal? To ensure the ontology contains sufficient information to answer these questions, artificial instances may be generated for testing. A total of five scenarios were identified throughout the ontology development process, and periodic self-checks were conducted to enhance the semantic validity of EIAO.

Additionally, the improvement in query time by EIAO in Neo4j was measured and compared with manual checking and RDF-based ontology. For each query, the time of finding the specific information in printed documents manually was recorded. As for the RDF-based ontology, the Neosemantics (N10s) plug-in enabled the LPGs to be transferred into the RDF-based data, and SPARQL could achieve most of the queries for the RDF. The entire conversion process was executed within Apache Jena runtime environment, which is a robust platform designed for the implementation of RDFs and SPARQL; it aligned seamlessly with the specific needs and prerequisites of the experiment. Throughout this process, a stringent control mechanism was applied to the information imported into both the RDF and LPGs to mitigate and prevent any potential errors or inconsistencies. This meticulous approach ensured that the data were accurately and reliably transferred to both the models. As a result, the distinctions and

variations between manual data processing, RDF, and LPGs became readily apparent and could be discerned with precision and clarity.

Detailed information on the case demonstration process, data preparation, and evaluation method will be presented in Chapter 5. Important criteria, such as clarity, correctness, and complexity were measured in the tasks separately (Gómez-Pérez, 2001). For instance, clarity could be defined by feedback from application and interviews (i.e., score or word comments from managers and customers on improving the behaviour and user-friendly level). The results were passed back to the ontology development process to optimise it. Correctness and complexity could be collected from comparing the ontology information with the original documents to infer if ontology was missing out some data after the transfer. The final EIAO could automatically identify a related resource for a certain activity, suggest a fast-responding path, and visualise all the relevant information.

3.7 Design of automatic information extraction model for special data in RAM (Objective 4)

Documents serve to present and share different types of information. Automatic word extraction has been examined heavily in the past few years. However, automatic models for extracting information from tables which provide a way to display structural and functional information, have not been fully investigated. Tables are a useful and efficient way for readers to compare, interpret, and understand data, particularly numeric values. Tabular data is the most prevalent data type in various real-world applications, including recommender systems (Cheng & Ugrinovskii, 2016), online advertising (Song et al., 2019), and portfolio optimisation (Ban et al., 2018). Many machine learning competitions, for example, those hosted on platforms, such as Kaggle and KDD Cup, are predominantly focused on addressing challenges within the tabular data domain. These competitions often revolve around developing innovative solutions to tabular data-related problems, reflecting the significance of tabular data in contemporary data science and machine learning endeavours. However, identifying tables within digital documents can be challenging.

Thus, we proposed an ATEIM for table. This model is based on a self-supervised approach that focuses solely on tables; this sets it apart from previous approaches which

do not consider the specific features of tables (such as correspondence between header and value). The model includes two key parts, namely a self-supervised transformer and a triple importer.

3.7.1 Automatic table information extraction and ontology improvement

Figure 3-7 illustrates the overall architecture of the model. In the context of RAM, the establishment of a domain using ontology necessitates a specific approach for processing tabular data. Firstly, it is essential to employ a transformer to convert matrix-format information into individual information nodes. Subsequently, a pre-trained model is utilised to map the main characteristics of the information to correspond with the previously established KB, using the logic and format of triples. This process enables the creation of data that can seamlessly integrate with the existing ontology, thereby enriching the existing KB. After automatically identifying nodes that match the desired features, this method automatically supplements new information at locations, where it matches with the contextual nodes or relationships. As a result, a new ontology is generated, and the previously lost table information is transformed into structured knowledge, thereby enhancing the overall completeness and comprehensiveness of the RAM knowledge pool.

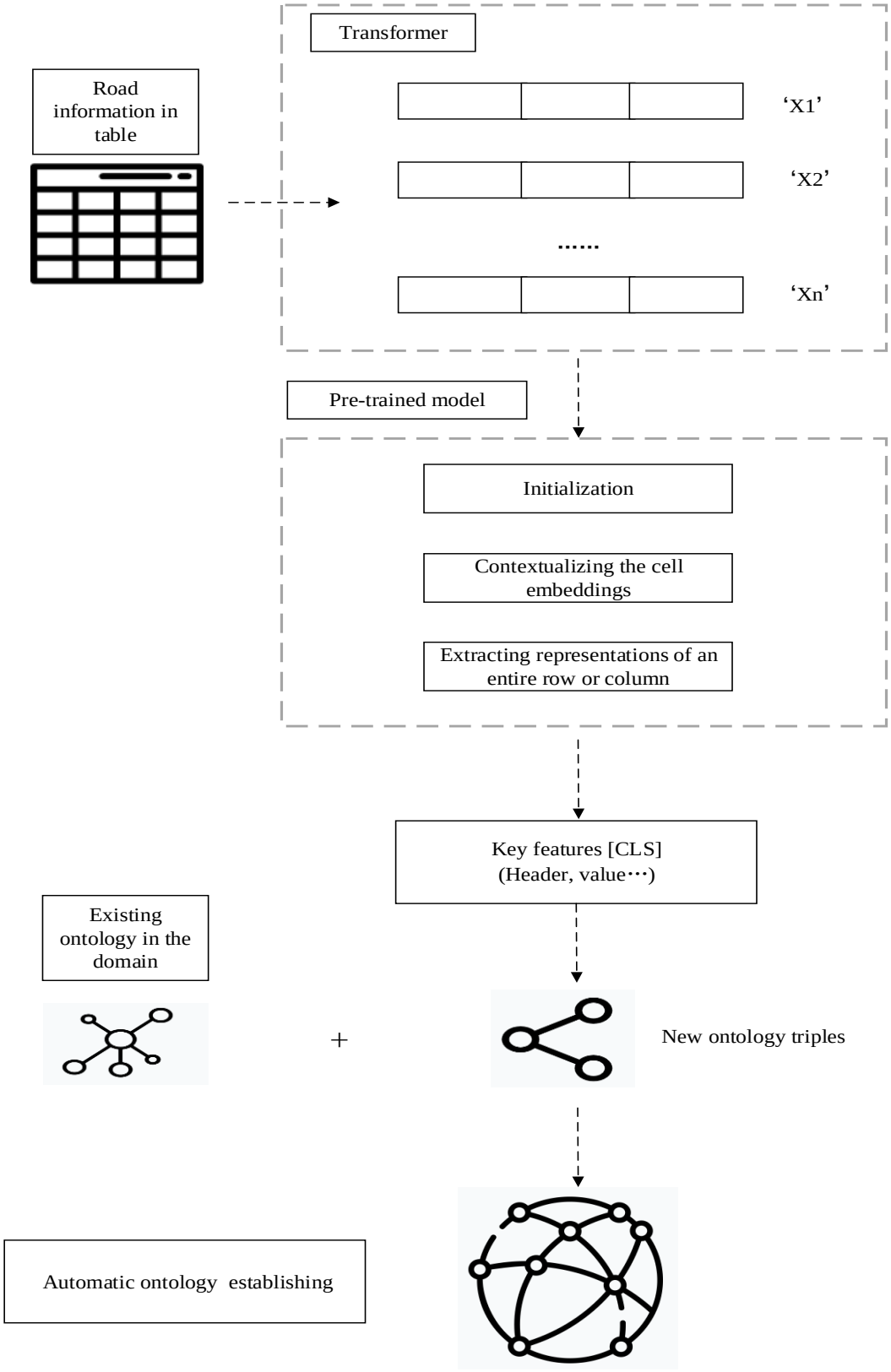


Figure 3-7 Automatic information extraction and ontology establishment method

3.7.2 Data inputs and outputs

The model discussed in the section utilises the same material, which was generated in EIAO Objective 3, with tables as the input and extracts table information in the form of triples in the EIAO, thus resulting in a more comprehensive ontology. An example of the model application is shown in Figure 3-5, where the model firstly identifies positional information (i.e., [CLS] token, which is a special token type in BERT model) from both rows and columns. Secondly, based on a sufficient amount of training, the model determines whether each position corresponds to a semantic word or non-semantic information (in this example, simplified as table headers or numerical values). The model then proceeds to infer relationships between positions based on their respective locations. In this particular example, the model can discern a directional relationship 'has_value_of' between the position ''Xi,j'' below the 'header'.

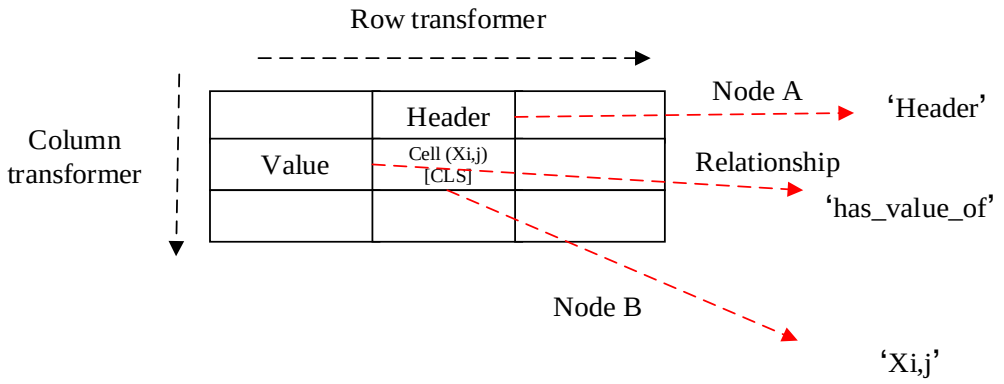


Figure 3-8 Transformation model

3.7.3 Overall design of the ATIEM model

We commence by initializing the cell embeddings xij using a pre-trained BERT model (Devlin et al., 2018). Specifically, for each cell $(i,j)(i,j)$, we input its contents into RoBERT and extract the 768-dimensional [CLS] token representation. This process enables us to leverage the robust semantic text encoder of RoBERT to compute representations of cells out-of-context, a crucial aspect considering that many tables contain cells with lengthy text (e.g., Notes columns). Furthermore, RoBERT has demonstrated the ability to encode a certain degree of numeracy (Wallace et al., 2019), facilitating the representation of cells containing numerical content. We maintain this fixed BERT encoder throughout training to mitigate computational costs. Lastly, we

incorporate learned positional embeddings into each of the [CLS] vectors to constitute the initialization of xij. Specifically, we employ two sets of positional embeddings, More specifically, we have two sets of positional embeddings, p (r) i ∈ R H and p (c) j ∈ R H, which respectively model the positions of rows and columns. These embeddings are randomly initialized and fine-tuned via TABBIE's self-supervised objective.

The ATIEM model was designed to process an input table with M rows and N columns. It generates embeddings for each cell within the table, and also produces embeddings for each column () and each row () present in the table. The design of process has three main steps : 1) Initialisation**:** The cell initialisation employs a pretrained robustly optimised bidirectional encoder representations from transformers (BERT) approach (RoBERTa) model, where the cell content is processed by RoBERTa, and the [CLS] token's dimensional representation is extracted. This process is valuable for handling cells with long-form text found in many tables. RoBERTa's ability to encode numeracy is particularly useful for representing cells with numerical content. 2) Contextualising the cell embeddings: Uncontextualised RoBERTa cell embeddings are typically computed independently for each cell in the table. It introduces a row transformer to encode the cells within each row and a column transformer for the columns. This method facilitates contextualisation of the embeddings, while avoiding the computational complexity of linearisation. 3) Extracting representations of an entire row or column: To capture the complete contents of entire rows or columns in the ATIEM, adaptations are made to the row and column transformers by incorporating special tokens. By integrating these special tokens and extracting the final-layer cell representations, the ATIEM facilitates the use of comprehensive embeddings that capture the contents of entire rows or columns in various table-related tasks.

3.7.4 Pretraining

With regard to the ATIEM's training objective, we have adopted the self-supervised ELECTRA objective proposed by Clark et al. (2020) for text representation learning. This objective involves applying a binary classifier to each word in a text and determining whether the word is a part of the original text or has been corrupted. While the ELECTRA objective was initially developed for more efficient training compared

to RoBERTa's masked language modelling objective, it is particularly well-suited for tabular data.

RoBERTa is a language model introduced by Liu et al. (2019), and was based on the BERT architecture, which is a popular model for NLP tasks. RoBERTa builds upon BERT by applying various modifications and training techniques to improve its performance. The key differences between RoBERTa and BERT lie in the training methods. RoBERTa is trained on a larger corpus of unlabelled text data, using dynamic masking patterns and longer sequences, resulting in a more comprehensive language representation. It also benefits from advanced pre-training techniques, such as using larger batch sizes, training for more iterations, and removing the next sentence prediction objective. RoBERTa has achieved state-of-the-art performance on various natural language understanding benchmarks and tasks, demonstrating its effectiveness in tasks, such as text classification, named entity recognition, sentiment analysis, and question answering. It has been widely adopted in both academia and industry for a wide range of NLP applications.

Based on these features and requirements, RoBERTa was selected as the model framework in this research. It involves randomly masking a word in a table and then using the remaining known words to predict the masked word. The transformer model parameters are updated through backpropagation and gradient descent based on the difference between the predicted and actual words.

In the context of tabular data, detecting corrupted cells is a fundamental task in table structure decomposition pipelines (Raja et al., 2020; Tensmeyer et al., 2019). Incorrectly predicted row or column separators, as well as cell boundaries, can lead to corrupted cell text. We adapt the self-supervised ELECTRA objective proposed by Clark et al. (2020) for text representation learning, which places a binary classifier over each word in a piece of text and asks if the word either is part of the original text or has been corrupted. While this objective was originally motivated as enabling more efficient training compared to BERT’s masked language modeling objective, it is especially suited for tabular data, as corrupt cell detection is actually a fundamental task in table structure decomposition pipelines such as (Nishida et al., 2017; Tensmeyer et al., 2019; Raja et al., 2020), in which incorrectly predicted row/column separators or cell boundaries can lead to corrupted cell text. In our extension of ELECTRA to tables, a binary classifier takes a final-layer cell embedding as input to decide whether it has

been corrupted. More concretely, for cell (i, j), we compute the corruption probability as

$$Pcorrupt(i,j) = \delta(w|X, Li, j)$$

where L indexes Robert's final layer, σ is the sigmoid function, and w is a weight vector of the same dimensionality as the cell embedding. Our final loss function is the binary cross entropy loss of this classifier averaged across all cells in the table. The ATIEM extends the ELECTRA objective to tables by employing a binary classifier that takes a final-layer cell embedding as an input to determine whether the cell has been corrupted. Having described TABBIE's model architecture, we turn now to its training objective.

3.7.5 Automatic ontology establishing

After obtaining the triples in the table, we need to import them into the original EIAO. Here, we use the py2neo API, which allows the users to use Python as a programming language in the neo4j database, and automatically write the table data to Neo4j.

3.7.6 Model experiments

(1) Experimental data collection and pre-processing

Triples stored in the EIAO model were gathered for training and testing the ATIEM model. However, here, the triples needed to be extracted from tables in road asset materials. To ensure effective training, tables with and without lines drawn were both simplified into Microsoft Excel format. We aim for as controlled of a comparison with other methods (Yin et al., 2020) as possible, as its performance on table QA tasks indicate the strength of its table encoder. TaBERT's pretraining data was not publicly released at the time of our work, but their dataset consists of 26.6M tables from Wikipedia and the Common Crawl. We thus form a pretraining dataset of equivalent size by combining 1.8M Wikipedia tables with 24.8M pre-processed Common Crawl tables from Viznet (Hu et al., 2019).

(2) Training and validation

Before the experiments, the ATIEM model was also trained using the training and validation datasets, and then was evaluated in the testing dataset. Numerous mature pre-trained transformer models from an open-source repository exist, which have been trained on a billion-scale corpus (i.e., BERT and RoBERTa). This type of pre-training is typically performed using self-masking for unsupervised learning.

For binary classification, the model predicts whether the masked word corresponds to a header or not. For multi-class classification, the model predicts the specific type of header or table value that the masked word belongs to. This scenario represents a typical 'named entity recognition' prompt problem, where the goal is to identify and classify specific entities within the table based on the predicted values for the masked words.

A simple validation was adopted to check the accuracy for identifying the header and corresponding values after applying RoBERTa and the pre-trained data.

(3) Automatic triples inputting

After obtaining the triples from the tables, the last step involves inputting them into the EIAO and linking them with the previous triples if they are describing the same term. In this experiment, py2neo was used as a bridge between the text type and LPGs type. By running a specific code in Neo4j, the entities would be automatically created and linked by relations. In this case, a simplified relationship definition was adopted, and the experiment only considered one type of relationship between nodes: The 'header' 'has_a_value_of''value', where 'header' and 'value' were automatically extracted from last step.

(4) Performance metrics

To assess the quality of ATIEM's table representations, we evaluated its performance on three downstream table-centric benchmarks, namely column population, row population, and column type prediction. These benchmarks were designed to measure the model's semantic understanding of tables. In the majority of configurations for these tasks, the ATIEM achieved superior performance compared to another model, transformer-based augmented BERT (TABERT) (Yin et al., 2020) and other baseline models, thereby establishing new state-of-the-art results. It is important to note that we did not specifically investigate the ATIEM's performance on table-and-text tasks, such as WikiTable-Questions. Our focus was not on integrating the ATIEM into complex task-specific pipelines, as outlined in previous works. However, exploring this direction in future research would be of great interest.

Additionally, we compared the pretrained models trained with different cell corruption strategies for downstream tasks. The first strategy, FREQ, used a frequency-based cell sampling approach exclusively. The second strategy, MIX, consisted of a 50/50 mixture of frequency-based sampling and intra-table cell swapping. In the MIX strategy, half of the intra-table swaps were required to come from the same row or column, adding an additional level of challenge to the objective. The purpose of this comparison was to evaluate the impact of these different strategies on the performance of the pretrained models in downstream tasks. The quantitative indicators included:

- MAP (mean average precision). MAP measures the average precision of a ranked list of documents retrieved for a given query. It calculates the precision at each relevant document rank and takes the average of these precision values across all queries.
- MRR (mean reciprocal rank). MRR, on the other hand, measures the effectiveness of a retrieval system by considering the rank of the first relevant document in the ranked list. It calculates the reciprocal rank for each query, which is the inverse of the rank of the first relevant document. The MRR then takes the average of these reciprocal rank values across all the queries.
- F1 score. It is a commonly used evaluation metric in binary classification tasks. It combines precision and recall to provide a single measure of a model's performance. The formula for calculating the same is listed below:

$$F1\ Score = \frac{2 \times precision \times recall}{precision + recall}$$

(5) Benchmark models

Next, we have selected popular and emerging models in recent years for experimental comparison. TAbert can be used for comparison in row and column experiments because it can achieve separated calculation of rows and columns. The other models can only achieve overall calculation; therefore, comparisons were made in the overall score calculation.

- TABERT is a language model introduced by Yin et al. (2020). It is an extension of the BERT model that incorporates table information for enhanced understanding and processing of tabular data. Unlike the traditional BERT, which primarily focuses on sequential text, the TABERT incorporates the structural information present in tables. It leverages both the content and

context of the table cells to improve the representation learning. TABERT uses a novel table encoding technique that converts tables into text representations, which are then combined with a textual input to provide a comprehensive representation of the data.

- BERT-TextRank: This model, proposed by Shi et al. (2022), utilised the BERT pretrained model to extract semantic information and employed the TextRank technique to enhance the entity description. The TextRank model uses three keywords.
- RoBERTa-BiLSTM: Inspired by the existing entity linking methods, we constructed the RoBERTa-BiLSTM model. It leverages bidirectional long short-Term memory (BiLSTM) to capture contextual information by concatenating the forward and backward hidden-layer vectors, thereby enabling a comprehensive extraction of the textual information.
- RoBERTa-Attention: Inspired by the work of Chen et al. (2023), this model incorporates an attention mechanism to enhance the representation of the textual information. The attention module focuses on important information and reduces the interference of irrelevant information, thereby improving the model's performance.
- RoBERTa-TextRank: This model applies the TextRank technique to RoBERTa, enhancing text topics and facilitating the extraction of semantic information. The TextRank model uses three keywords.

3.8 Summary

This chapter has provided an overview of the research methodology employed in this thesis. It began by introducing the research philosophy, which aligned with the post-positivism paradigm. The methodology is primarily deductive and quantitative, based on objectivism epistemology and realism ontology. However, qualitative methods, including focus groups, were also used to obtain relevant domain knowledge. Sections 3.5, 3.6, and 3.7 presented specific research methods. In Section 3.4, a comparison between the RDF and LPGs data models was presented to identify the better environment for ontology in RAM. Section 3.5 focussed on constructing EIAO to integrate these triples and support information computation, reasoning, and updates.

Section 3.6 introduced the development of ATIEM model to address automatic table information extraction and inputting those into ontological knowledge database, which could continuously enrich the established ontology. By employing these components, the RAM system could be automated, thereby facilitating project the management through a timely integration of valuable information.

4 A systematic comparison of ontology techniques

4.1 Introduction

This chapter compared two popular graph data models for ontologies within an AEC project background, identifying the superior model in different application contexts. The research utilized a machine with the following configuration: Intel Core i7-6700HQ @ 2.60 GHz, 16 GB DDR4 RAM @ 2133 MHz, and a 960 GB SSD. Additionally, specific processes required an internet connection.

4.1.1 Data collection

Comparisons always require one or more ontologies to implement the experiment. The datasets not only require a description of logical information (e.g., traffic information or biological information) from the real world but also need to have different scales. A reasonable size can keep the experiments controllable and efficient since processing a large number of triples requires a powerful device and long-term costs. However, determining the number of triples or vertexes that the ontology should have to perform a significant comparison is a challenge. Therefore, to obtain a reasonable ontology size that can ascertain the main differences, relevant comparison works are listed in Table 6. It should be mentioned that different research may use different indicators to measure the data size, such as triples (RDF), the number of vertexes (LPGs) or bytes.

Table 6 Database size in relevant work

References	Research method	Comparison target	Model
Haase et al. (2004)	Review	5 RDF-based languages	RDF
De Abreu et al. (2013)	Experiment	1000-100 k (thousand) vertexes 100 k-1 m (million) edges	RDF and LPGs
Holzschuher and Peinl (2013)	Experiment	83,500 vertexes	LPGs

Gong et al. (2018)	Experiment	50 k-2.5 m storage	RDF
Drakopoulos et al. (2017)	Experiment	31 vertexes 499 edges	LPGs
Vicknair et al. (2010)	Experiment	1 k-100 k triples	RDF and LPGs
Alocci et al. (2015)	Experiment	230 k vertexes	RDF and LPGs
Das et al. (2014)	Experiment	10 k-25 m triples	RDF
Angles (2012)	Review	8 graph data models	RDF and LPGs
Abdelaziz et al. (2017)	Experiment	100 m triples	RDF
Schmidt et al. (2008)	Experiment	10 k-25 m triples	RDF
Donkers et al. (2020)	Experiment	1,000 vertexes	RDF and LPGs
Jouili and Vansteenberghe (2013)	Experiment	500 k triples	RDF
Gorawski and Grochla (2020)	Experiment	1,000-3,000 vertexes	LPGs
Guia et al. (2017)	Experiment	1.2 GB (Gigabyte)	LPGs
Neumann and Weikum (2010)	Experiment	1 m triples	RDF
Anikin et al. (2019)	Experiment	10 m triples	RDF
Lampoltshammer and Wiegand (2015)	Experiment	100-10 k triples	RDF and LPGs
Sharma et al. (2018)	Experiment	520 k vertexes	LPGs
Thakkar et al. (2018)	Experiment	3 k-90 k vertexes	RDF and LPGs
Constantinov et al. (2015)	Experiment	100 k vertexes	LPGs

Based on the ontology datasets collected in the research above, the proper and reasonable size for cross comparisons between RDF and LPGs can be relatively small, and 1000-10,000 triples or vertexes are sufficient.

Thus, a set of criteria has been determined for selecting ontologies. A qualified ontology must be 1) maintained in the last half year, 2) built by using available ontology tools, and 3) applicable within the AEC industry. The final datasets were chosen from a study by Wu et al. (2021), and they contained a mature ontology that included approximately ten thousand triples (RDF), and then LPGs were obtained by converting RDF triples. A mature converting tool, the Neosemantics (N10s), will be used convert the datasets to keep the experiments under the same environment. It has been developed and operated many years and been applied in a lot of research and cases. For instance, Sfoungari (2021), Urbieta et al. (2021) and Berges et al. (2021) all used this tool to transfer their ontologies (COVID-19 knowledge ontology, automotive global ontology, and Industry 4.0 Big Data ontology) between RDF and LPGs, which proofed that it is a lossless manner. Additionally, to reduce other random errors that may be caused by using different ontologies, datasets of different scales in the same ontology were obtained to perform the comparison (Holzschuher & Peinl, 2013).

4.1.2 Case ontology description

The selected ontology is a concrete bridge rehabilitation process, which includes common rehabilitation constraints, procedures, tasks, and participants. In this ontology, a task often contains several procedures. The target bridge is a suspension bridge which is 400-m long and 42-m wide and built in China. The total project took approximately six months to complete in 2018. The main information resources contain standards, manuals, and project documents (e.g., case reports, work plans and quality evaluation reports). After generating enough knowledge and data, the classes, and subclasses (or hierarchy) of ontology were firstly defined (e.g., the whole ontology has three main classes, namely bridge components, project participant and rehabilitee task). And then, the detailed information and value were filled into each level, and extra edges were also added to express the relationships between the concepts. Finally, all relevant information were stored and presented in a computer-readable dataset.

4.1.3 Three datasets

Three different sizes of ontology datasets were extracted. The overview of their information hierarchies is shown in Figures 3, 4 and 5 respectively.

- Large ontology dataset (Figure 4-1): The largest dataset is the bridge maintenance ontology itself. It consists of bridge component, project participant, rehabilitation task and constraint, and other subclasses, such as deck system, hazard treating and replacement task.
- Medium ontology dataset (Figure 4-2): The medium-size dataset, which is rehabilitation task, is extracted from one subclass of the large dataset. It includes detailed information and work procedures related to rehabilitation, such as replacement of structure components, activities related to reinforcement, hazard analysis and environmental impact assessment task.
- Small ontology dataset (Figure 4-2): The small dataset, which is environmental impact assessment is a subclass of the medium dataset. It includes detailed environmental impact assessment activities during the project rehabilitation, such as air and noise monitoring, waste management, visual impact management and license management.

Thus, three datasets with sizes from small to large were set up and prepared for quantitative comparison. These datasets can also represent small, medium, and large bridge maintenance projects.

After the selection of the ontologies for testing, the next step is to import the data into both RDF and LPG tools for use. Since the ontology was originally created and edited by OWL, it can be directly used by RDF-based tools. However, RDF information cannot be presented directly in LPG-based tools and needs to be transferred to LPG format. The Neosemantics (N10s) plug-in is used to import and export RDF data into LPGs, and it has been tested in many projects and proved its ability to maintain data integrity. The transformed datasets will be compared with original one and check each concept, relationship, and property information to avoid information missing. Finally, the three datasets used for the experiments are listed in Table 7.

Table 7 Selected datasets

Size	RDF (triples)	LPGs (vertexes+edges)	Engineering background
Small (S)	777	276+347	Simple daily environmental assessment procedure for bridge rehabilitation management.

Medium (M)	2800	923+1866	Complete workflow for rehabilitation task when a risk was found.
Large (L)	15444	4314+7650	Whole life cycle of a bridge rehabilitation management project, which is from assessment discussions, design discussions, construction and maintenance.

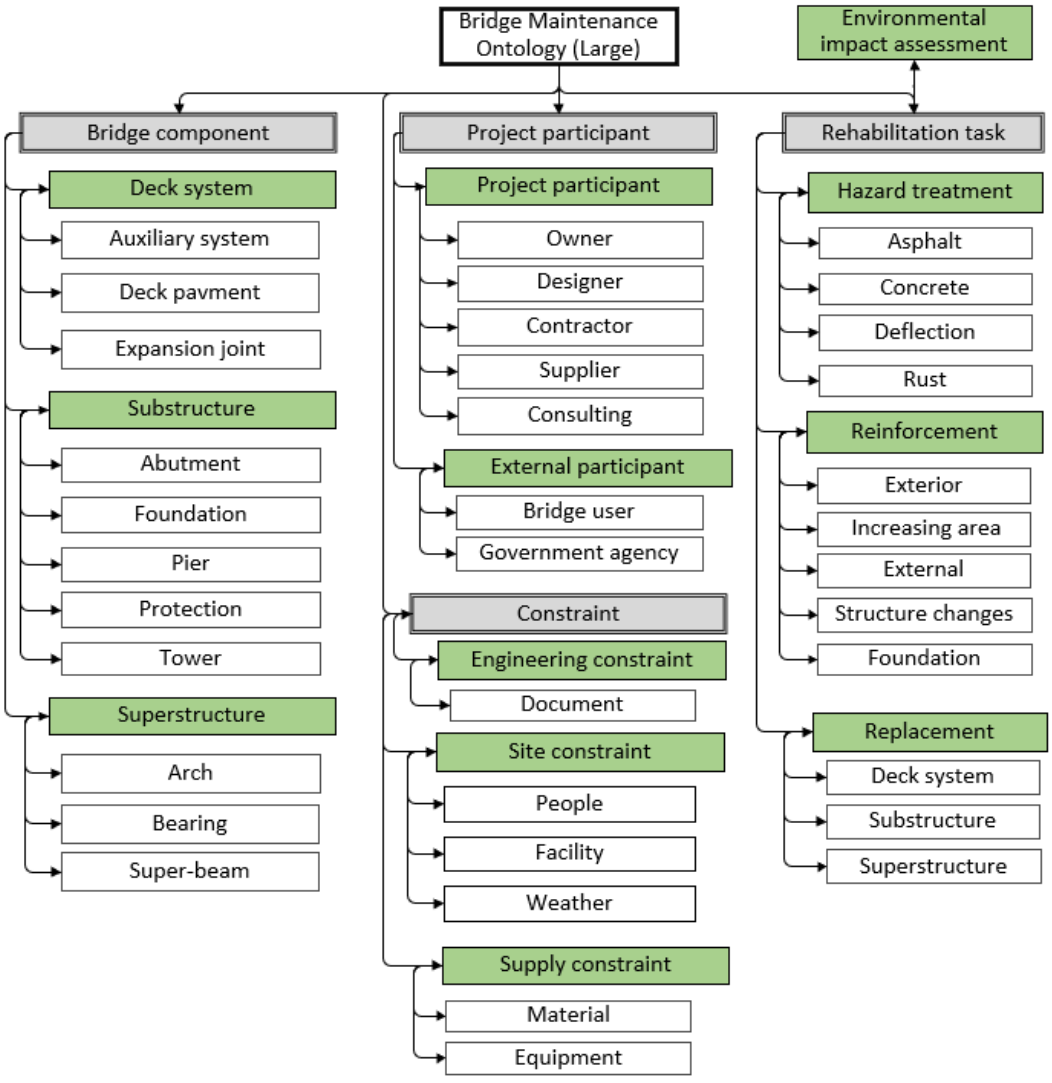


Figure 4-1 Overview of the large dataset.

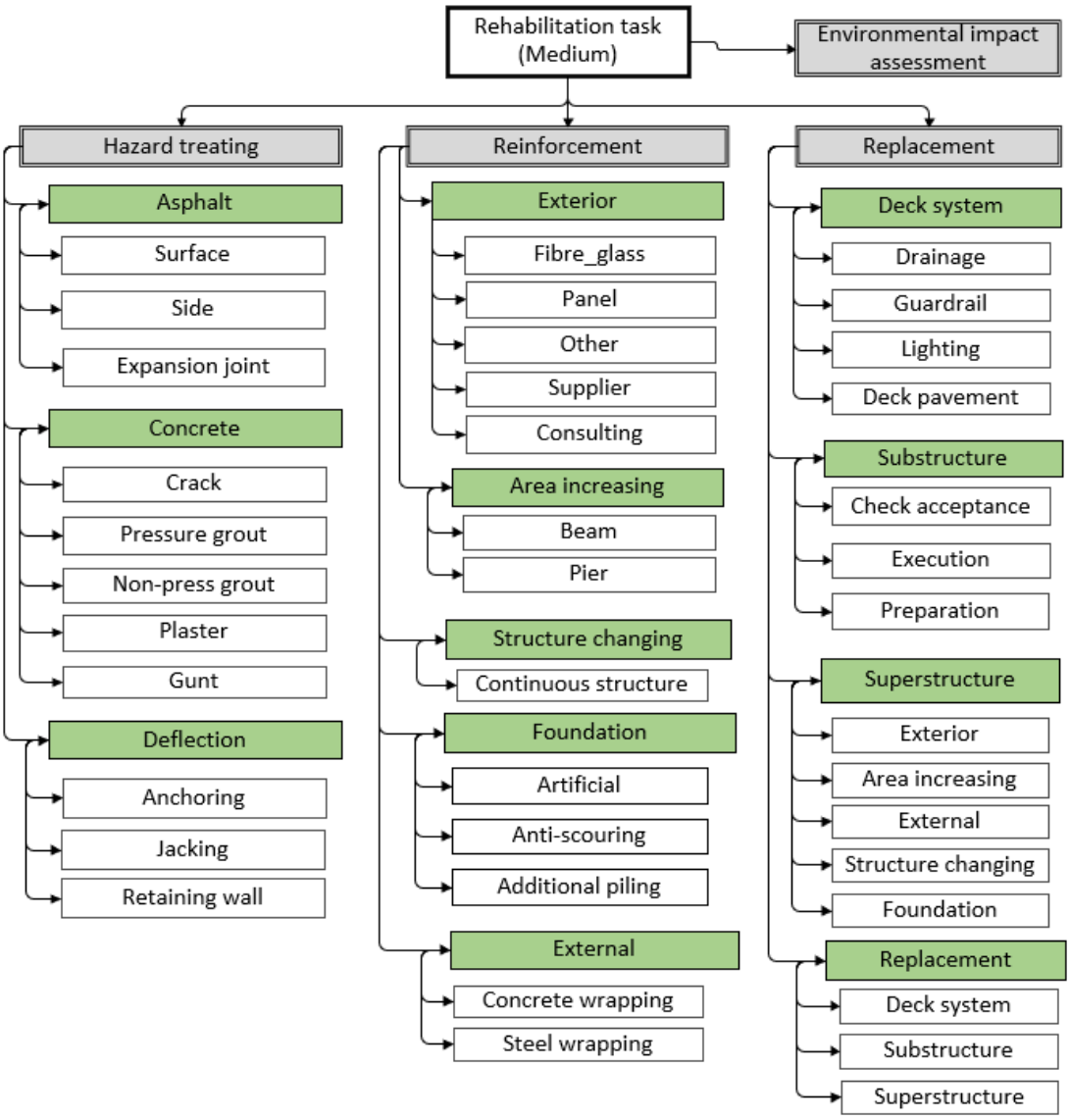


Figure 4-2 Overview of the medium dataset.

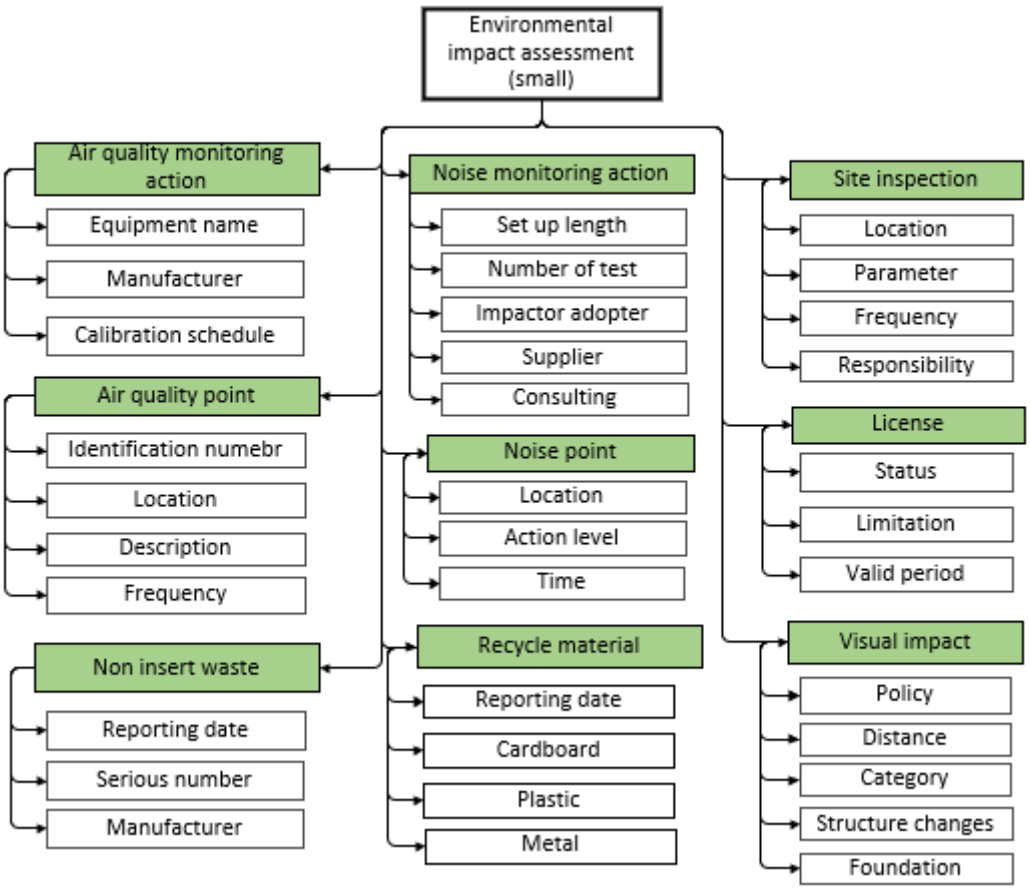


Figure 4-3 An overview of small dataset.

4.1.4 Indicators for comparison

1) Data density

$$\rho = \frac{r + p}{s \cdot n(n-1)}$$

where

n represents the number of vertexes,

p represents the number of properties,

s represents the storage size read from disk,

r represents the number of relationships.

The index indicates the information density, with a larger value indicates a higher data density. In this case, all the variables can be extracted from ontology files based on codes or file properties. RDF data were stored as local files, and their size could be

directly measured. The ontology was first loaded into Apache Jena, which provided a stable environment for the RDF graph and embedded some basic functions. By using a specific language via a native API or web endpoint, the information of triples (including the number of vertexes and edges) could be executed directly. On the other hand, LPGs were stored in the default cloud of the Neo4j server. Thus, the required information (e.g., the number of vertexes and edges) was exported by the code '*:info*' and shown in the interface.

2) Query efficiency

All three datasets (large, medium and small) were tested in this section; therefore, the influence of the size of the dataset can also be analysed. Both RDP and LPGs have the ability to search information from simple to complex ontologies. As ontologies have engineering backgrounds, queries were also defined with engineering meanings. to perform a comprehensive comparison, each query was executed three times against each dataset. The designed queries are listed in Table 8.

Table 8 Query codes list

Description	Application	RDF codes	LPGs codes
Find all vertexes	Find all information points of the project.	*SELECT?vertex WHERE {?vertex rdf:type onto_exp:Thing. }*	*MATCH (n) RETURN n*
Find all relationships	Find all relationships between the project elements.	*SELECT?relation WHERE { {?relation rdfs:subPropertyOf onto_exp:top-property} UNION {?relation rdfs:subPropertyOf onto_exp:top-data-property}*	*MATCH p=()-->() RETURN p*
Find all classes (labels)	Find information categories/resource of projects.	*SELECT?class WHERE {?class_hidden rdfs:subClassOf?class. }*	*CALL db.labels ()*
Find vertexes with certain feature	For example, find the constraints	*SELECT?vertex*	*MATCH (n:{is-timely removed:'false'}) RETURN n*

	which has not been removed timely	*WHERE {?vertex cbrpmo:is-timely removed-false. }*	
Find the relationships with certain features	For example, find the sub-property of constraints)	*SELECT?relation WHERE {?relation rdfs:subPropertyOf cbrpmo:constrains. }*	*MATCH p=()-[r: subPropertyOf]->(n:con straints) RETURN p*

3) Reasoning

This function was tested in six simulated scenarios developed from the selected ontology, and the scenarios also contained enough engineering background to support a project. How well the RDF and LPG approaches performed in reasoning extra information as required was assessed. Scenarios, codes, tools, and plug-ins used by the models are listed in Table 11. It should be noted that LPG, which is a relatively new graph technology that was not primarily developed for reasoning, may have intrinsic limitations on this benchmark (Gong et al., 2018). However, a comparison is still necessary to obtain experimental results.

Table 9 Reasoning codes list

Reasoning tasks	Scenario description (Scenario 1-6)	RDF tool & plug-in	Reasoning rules	LPG tool & plug-in	LPG rules
Reasoning vertexes/relationships	Project manager wants to know all procedures that belong to a certain task.	Protégé HermiT	*Procedure(?p) ∧ Task(?t) ∧ part-of(?p,?t) → sqwrl:select(?p)*	Neo4j N10s	*n10s.inference.nodesInCategory(n:task) return n*
	When a procedure has been finished, its constraints should be automatically removed to release other constraints. The manager then finds a problem in which certain constraints have not been rapidly removed and he/she wants to determine the reason.	Protégé HermiT	*Constraint(?c) ∧ Procedure(?p) ∧ constrains(?c,?p) ∧ is-timely removed(?c, false) ∧ has-reason(?c,?r)→ sqwrl:select(?c,?r)*	Neo4j N10s	*n10s.inference.getRels(n:constraints)->(p: is-timely removed {has-reason} XOR (p.age < 30)* *Where p XOR (p: is-timely removed 'false')* *return n,p*
	Find and order critical constraints in the constraint network based on the out-degree of constraints (i.e., the number of edges that link	Protégé HermiT	*Constraint(?c) ∧ has-out-degree(?c,?l) → sqwrl:select(?c,?r) ∧ sqwrl:orderBy(?r)*	Neo4j N10s	*Match p = allshortestPath (n->(p: degree)XOR(p:degree < 0))*

	a constraint with other constraints or tasks/procedures).				
	Project manager wants to identify whether one task is delayed. Then, he/she also wants to know how the delay will affect the future work.	Protégé HermiT	*Task(?t) ^ (has-total-schedule-delay some xsd:integer[> 0])(?t) ^ is-succeeded-by(?t,?ts) →is-delayed(?t, true) ^ sqwrl:select(?ts)*	Neo4j N10s	*n10s.inference.hasLabel(n: has-total-schedule-delay>0)* *match (n)->(p)* *return n,p*
Reasoning specific properties	The project manager wants to find delayed constraints for a certain procedure based on removal delay.	Protégé HermiT	*Constraint(?c) ^ is-constrained-by(?p,?c) ^ (has-removal-delay some xsd:integer[>0])(?c) → is-timely removed(?c, false)*	Neo4j N10s	*n10s.inference.getRels(n:constraints)->(p: has-removal-delay)* *return n,p*
	The total duration of the procedure from commencement of a task is calculated based on sequential work dependency.	Protégé HermiT	*Procedure(?p1) ^ Procedure(?p2) ^ is-succeeded-by(?p1,?p2) ^ has-actual-duration-from-start(?p1,?adfs) ^ has-actual-duration(?p2,?ad) ^ swrlb:add(?y,?adfs,?ad) → has-actual-duration-from-start(?p2,?y)*	-	-

Note: '-' means that the function is currently unavailable/undeveloped.

4) Visualization performance

Ontology data models not only produce data in machine-readable formats but can also provide visualized information for human users. Both RDF and LPG methods need plug-ins to visualize information. Data visualization has become a hot topic in information presentation, and studies have focused on the visualization performance of ontologies. For instance, Dudáš et al. (2018) reviewed available ontology visualization methods and introduced certain supporting functions. These works provided a subjective method for performing comparisons.

Nonetheless, different RDF or LPG tools have particular advantages based on their theoretical core of data structure. Thus, Protégé software for building and maintaining ontologies using RDF was chosen for these benchmarks, and it is also one of the most widely accepted and used tools for ontologies (Asim et al., 2018b). By using OWL and OntoGraf plug-ins, ontologies can be presented from text to flexible graphs. However, the visualization function of Neo4j is one of the outstanding features (Donkers et al., 2020). The original data were automatically stored in graphs, and detailed properties can be read from the inferences.

The default plug-ins OntoGraf in Protégé and Neo4j Browser were used for the comparison, and there are two reasons for this decision. Firstly, although there are many visualization plug-ins for both Protégé and Neo4j (e.g., OntoViz, TGVizTab and Bloom), most of the ontologies in previous and current studies were still using default tools to visualize data (Akrivi et al., 2006). Secondly, the default tools have the same core, which both implement a 2-dimensional vertex-link visualization method that visualizes ontologies as a vertex network (Dudáš et al., 2018). This will minimise the risk that the performance may be affected by tools with different cores. Moreover, they can display vertexes under certain classes, edges between vertexes and other visualized information, such as properties and classes. However, LPGs were designed to be visual in nature using graphs to display information, and this method can easily model the visualized information.

5) Review

As a relatively subjective function, a review work needs to collect information on the evaluation. The scope of the review was confined to the development and implementation of ontology visualization methods for the RDF and LPGs. The databases were selected as research websites (e.g., the Web of Science, Scopus, IEEE Xplore and Google Scholar) and relevant forums (e.g., GitHub and Stack Overflow). The search strings (‘ontology’ OR ‘RDF’ OR ‘LPGs’) AND (‘visual’ OR ‘visualization’) were used to determine the differences.

Studies have been conducted to provide an overview of visualized ontologies. For instance, Akrivi et al. (2006) evaluated four visualization techniques for RDF in Protégé by a group of users, and they voted the 'class browser' plug-in as the most effective method. The previously mentioned case by Dudáš et al. (2018) conducted a more comprehensive review on ontology visualization tools, which analysed 37 ontology visualization tools in detail and marked their functions. Combined with other well-accepted data visualization evaluation methods, a basic functional and availability comparison has been performed to present the features of different data models. A fourteen-criteria evaluation system was implemented, and the criteria are listed below:

- Visualization method: the method (structure) of the visualization function,
- Large ontology capacity: present large datasets (e.g., over 10 k triples/vertexes),
- List review: review the information as lists,
- Table review: review the information as tables,
- Zooming: zoom in or out of the figure,
- History tracking: show the history of the action,
- Query: query the information directly in the visualized information,
- Filter: choose the presenting information by filter,
- Click selecting: the information can be clicked,
- Drag and drop: the information can be moved smoothly,
- Textual editing: edit textual information directly,
- Visual editing: change the visual style,
- Class checking: review the class information directly,
- Annotation: the annotation function, and
- Property characteristics: review the embedded properties on the inference.

Hence, the defined functions were then tested in both graphs using the small dataset as an example. The first task was finding whether both graphs supported the function and marked the function. After this task, an overview of the visualization function can be made. However, some of the functions, such as zooming and annotation, were supported in both models, which required an additional evaluation to determine the best performance.

6) Focus group

A systematic survey was conducted that asked participants to make judgements about '*Which model has a better visualization performance in this certain criterion?*' by letting them mark each visualization function from 1 (worst) to 5 (best). In addition, some subjective

questions were raised to collect valuable comments on graph visualization (e.g., '*Are ontology and information visualization useful for project management or other processes?'*). The detailed survey questions are included in Appendix A.

After introducing the basic survey information, the small dataset was visualized in two graphs. To better understand the survey, a comparison example of 'zooming' is presented in Figure 4-4 and 4-5. After being shown the zooming process, participants were asked '*Which one do your think perform better on zooming function*?' By taking the average marks, a subjective evaluation was performed.

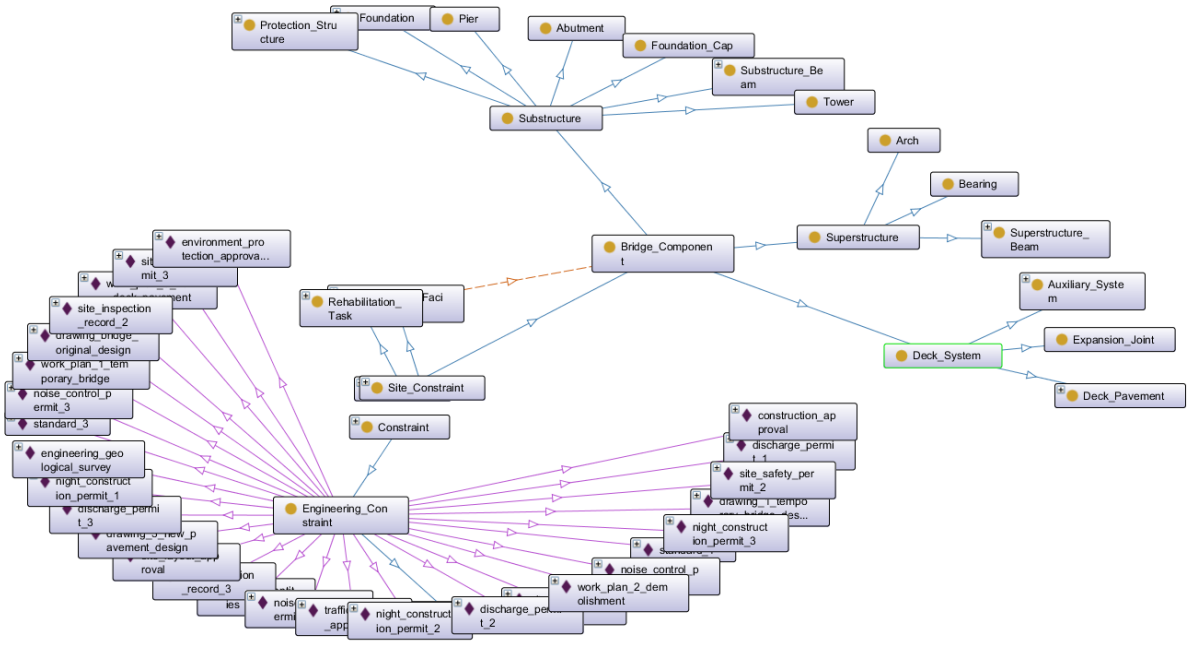


Figure 4-4 Zooming in function of Protégé (RDF)

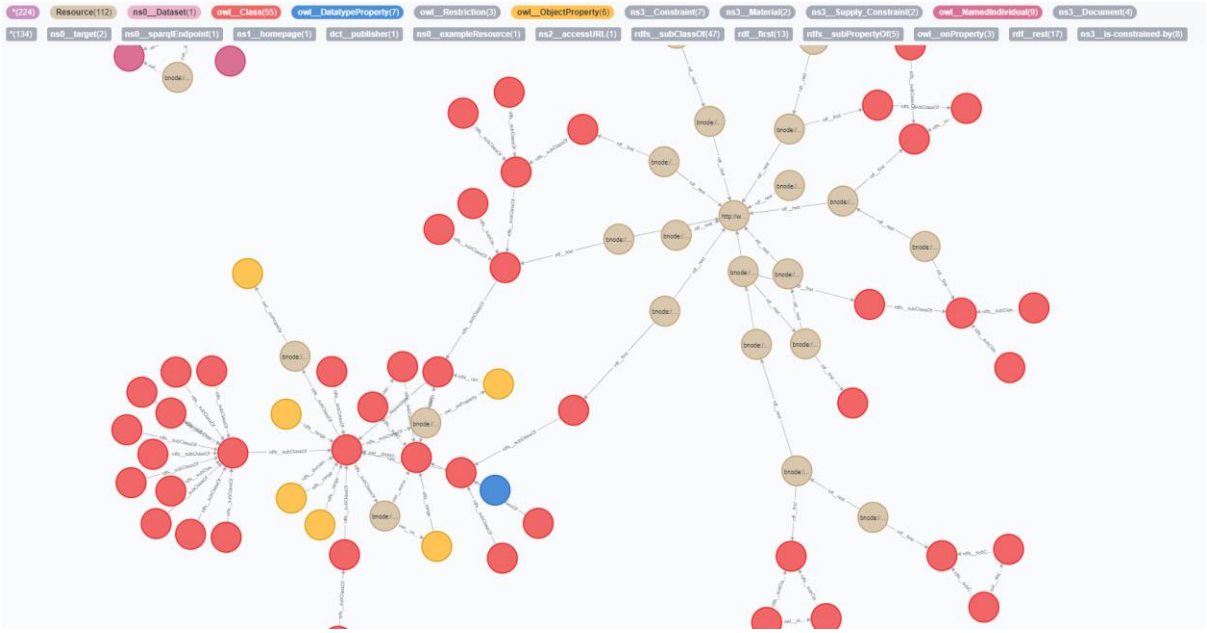


Figure 4-5 Zooming in function of Neo4j (LPGs)

4.2 Data density

The collected values from each model/dataset and the calculated density results are listed in Table 10.

Table 10 Data density comparison

Dataset	Small		Medium		Large	
Model	RDF(Rs)	LPG(Ls)	RDF(Rm)	LPG(Lm)	RDF(Rl)	LPG(Ll)
Triples	777	-	2800	-	16670	-
Vertexes	451	278	2153	937	12013	4314
Edges	676	347	3792	1866	18614	7650
Properties	-	625	-	2803	-	31140
File size (Megabyte)	0.112	0.287	0.746	1.312	2.416	4.557
Density ()	**0.030**	**0.044**	**1.097e-3**	**4.057e-3**	**6.019e-5**	**4.578e-4**

The density results are also illustrated in Figure 7. The difference in data density is clear: Although the file sizes read from computers are quite similar, the data density of the LPG approach is always higher than that of the RDF approach in all cases, and an increasing gap occurs when the scale of the dataset increases. In other words, the LPG approach performs better than the RDF approach in saving storage space when facing larger datasets. Due to the difference in the basic elements of the data structure, LPGs can present the same amount of

information with fewer vertices and edges. After transferring the ontology from the RDF to LPGs, many attached properties become embedded inside of the vertexes and edges, which can directly reduce the complexity of the data network.

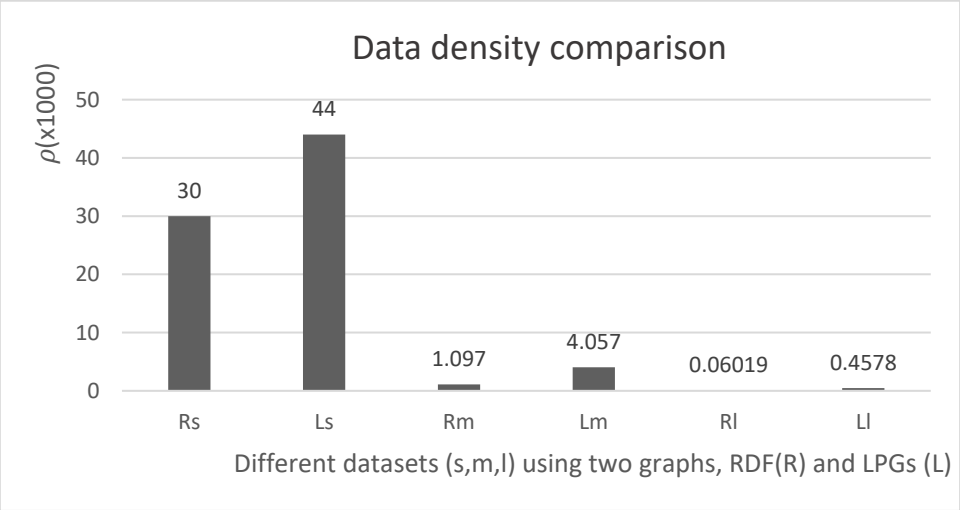


Figure 4-6 Data density comparison results.

4.3 Query efficiency

The brief query description and exaction times are listed in Table 8, and relevant figures are presented afterwards in Figures 4-7. Q1-3 represent the ability to search for general information, while Q4 and Q5 represent the search for featured information.

Table 11 Query efficiency comparison (unit: millisecond, ms)

RDF	S			M			L		
Find all vertexes (Q1R)	37	27	23	115	88	85	230	199	165
Find all relations(Q2R)	22	10	9	27	10	10	48	12	10
Find all classes (labels) (Q3R)	34	25	21	35	27	22	59	38	32
Find vertexes with certain features(Q4R)	13	10	7	16	9	8	26	13	11
Find relations with certain features (Q5R)	39	14	7	39	10	10	39	11	10
LPGs	S			M			L		
Find all vertexes (Q1L)	31	13	9	31	25	21	74	35	32
Find all relations (Q2L)	28	20	18	31	21	21	45	25	24
Find all classes (labels) (Q3L)	25	3	4	49	7	7	56	10	10
Find vertexes with certain features (Q4L)	36	4	3	46	8	7	74	15	11
Find relations with certain features (Q5L)	38	2	3	44	10	9	74	21	23

Due to the features of all structured query languages (SQLs), the time needed for queries will decrease significantly after the initial query. Little (2016) explained that this situation is based on the learning and training progress of query systems. When the query was executed three times, both of the models performed relatively stably. Thus, the first, second and third runs for each query were tested and presented.

Figure 4-9 presents the change trends of density from small to large datasets, and only the execution times for Q1 and Q5 (representing two typical types of querying: general and featured) of the third run are presented. For the same reason, Figures 9 and 10 present the internal difference of one model using only the first and third runs as effective values.

The final results indicate that LPGs require 50% less time than the RDF on average when querying information. When the data size of the RDF increases, the time cost increases almost linearly when finding all vertexes, although the effect is minor for queries associated with finding certain featured data. On the other hand, LPGs show an opposed behaviour: when the size of the dataset increases, the time needed for finding specific information changes significantly while the time needed to find all vertexes shows a relatively small fluctuation. A clear conclusion can be formed from the table: the LPG data model present a more efficient query function than the RDF data model when searching for the same information.

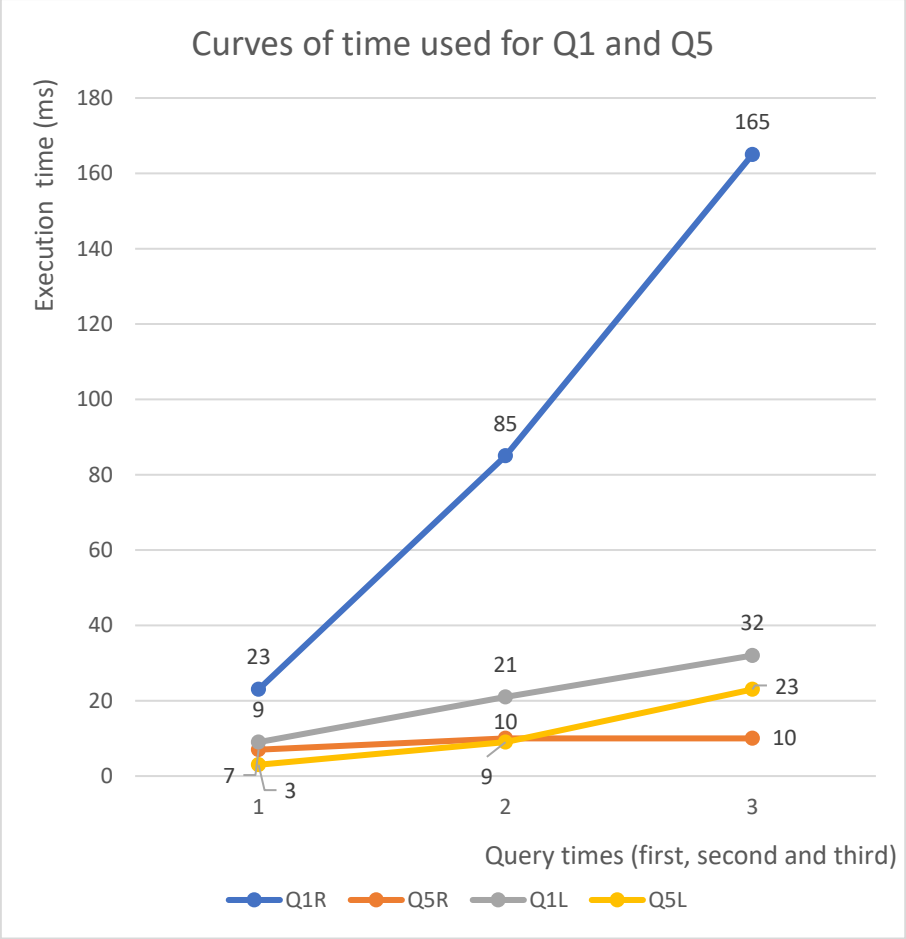


Figure 4-7 Curves of time used for Q1 and Q5.

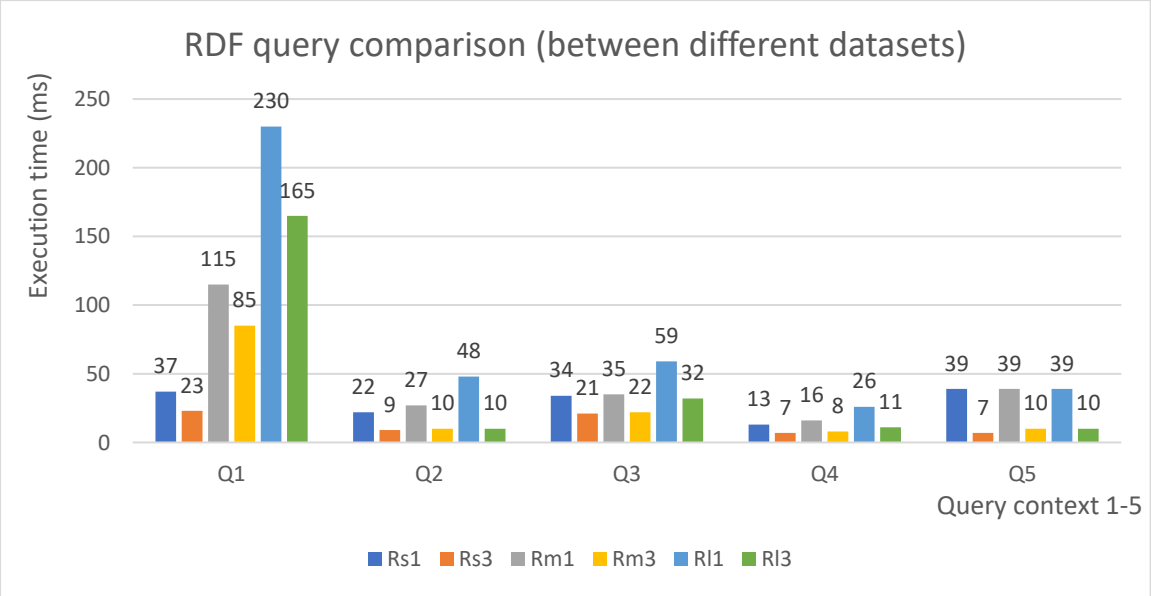


Figure 4-8 RDF internal comparison.

(Rs1: RDF-small-1st; Rs3: RDF-small-3rd; Rm1: RDF-medium-1st; and Rl1:RDF-large-1st)

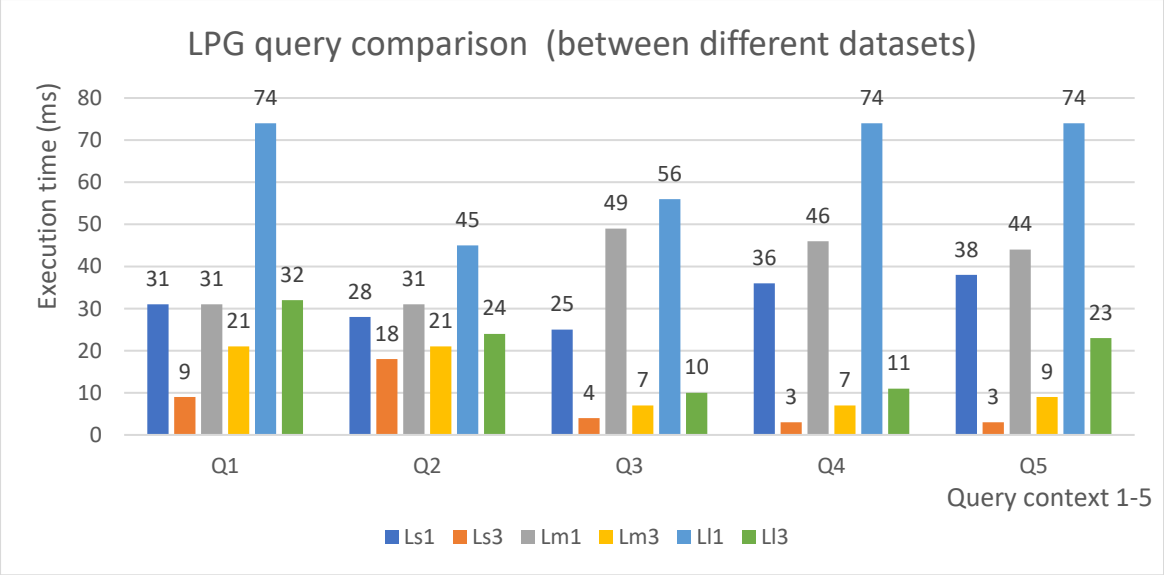


Figure 4-9 LPG internal comparison.

(Ls1: RDF-small-1st; Ls3: RDF-small-3rd; Lm1: RDF-medium-1st; and Ll1:RDF-large-1st)

The query 'finding a certain relation' presents the smallest differences compared with the other queries because of the specific data structure of LPGs. Searching the relation would still require a query of all vertexes before outputting the result, which increases the time cost. Similarly, the difference of the query 'all relations' relative to the other queries is also less than that of the query 'all vertexes'.

Other findings are also valuable. For instance, when facing complex queries, such as finding certain information, LPGs require more time than the RDF model in the first run in the majority of situations. The potential reason could be that Neo4j embeds many more functions and pre-order codes, which makes the cold start a more complex process than the simple Java API used by Jena. Subsequently, the time needed for the LPG model declines sharply, thus showing its advantage.

4.4 Reasoning

The reasoning results are listed in Table 12. For a better evaluation, the authors also manually checked the project, and the findings are listed in the row marked 'Manual' as correct outcomes. The time needed for each graph was also recorded. Then, the accuracy of reasoning can be calculated.

Table 12 Reasoning comparison

Scenario	Description	Manual		RDF		LPGs		
		Results	Time(s)	Result	Accuracy	Time(s)	Result	Accuracy
Scenario 1	Reasoning procedures for one task.	5 vertexes	11	5 vertexes	100%	15	5 vertexes	100%
Scenario 2	Reasoning problem constraints.	1 vertex	22	1 vertex	100%	40	1 vertex	100%
Scenario 3	Reasoning critical constraints.	11 vertexes 10 edges	47	11 vertexes 10 edges	100%	75	7 vertexes	64%
Scenario 4	Reasoning subsequent information for one certain task.	1 vertex and 5 following vertexes	17	1 vertex and 5 following vertexes	100%	16	1 vertex and 5 following vertexes	100%
Scenario 5	Reasoning subsequent relationships by property.	3 vertexes 6 edges	43	3 vertexes 6 edges	100%	55	1 vertex	11.1%
Scenario 6	Compute the properties.	18 days (a string/value)	77	18 days (a string/value)	100%	-	-	-

In different scenarios, the reasoning results may return to a vertex or edge or together. For instance, in scenarios 1 and 2, both graphs achieve 100% accuracy since the target and process are not complex. Another relatively easy scenario is scenario 4; although both vertex and edge need to be reasoned, both graphs also obtain 100% accuracy because of the direct relationship between the vertex and edges. In more complex reasoning scenarios, such as scenarios 3 and 5, which require internal property reasoning, LPGs have a low accuracy since multiple steps of reasoning are required. Finally, LPGs have not been sufficiently developed to address scenarios that require calculations, such as scenario 6. In addition, except for scenario 5, the RDF-based approach takes less time to obtain the reasoning result because the LPG-based approach has a specific code for the default function that finds the critical path; thus, for other temporary tasks, the RDF approach is faster.

A conclusion can be clearly made that the RDF-based approach has a much more powerful reasoning ability than the LPG-based approach.

4.5 Visualization function

The survey results from the focus group were collected and are listed in Table 13.

Table 13 Visualization comparison

Features/Techniques	RDF	Mark on average (1-5)	LPGs	Mark on average (1-5)
Visualization method	Vertex-link (2D)	3.8	Vertex-link (2D)	**4.5**
Large ontology capacity	Yes	3.4	Yes	**3.8**
List review	Yes	-	-	-
Table review	-	-	Yes	-
Zooming	Yes	4.1	Yes	**4.3**
History tracking	-	-	Yes	-
Query	Yes	3.5	Yes	**4.2**
Filter	-	-	Yes	-
Click selecting	Yes	4.0	Yes	**4.2**
Drag and drop	Yes	3.7	Yes	**4.1**

Textual editing	Yes	**4.2**	Yes	4.0
Visual editing	-	-	Yes	-
Class checking	Yes	3.7	Yes	**4.2**
Annotation	Yes	3.8	Yes	**4.5**
Property characteristics	Yes	3.9	Yes	**4.4**

Note: 'Yes' indicates that the model is able to achieve this function. Average scores from the survey are also attached, and the better score for each aspect is marked as bold text.

The results show that the LPG-based visualization method can achieve more functions (including '*history tracking*', '*filter*' and '*visual editing*') than the RDF-based approach. In addition to '*textual editing*', LPGs perform better for presenting information to audiences, such as '*zooming*', '*visual editing*' and *'property characteristics*'. '*Listing review*' is the only function that can be performed by RDF approaches but not LPG approaches. However, LPGs use *'table review*' to replace this function, thereby enabling users to review the information in tables. For the functions shared by both approaches, LPGs also do a better job compared with RDF except in 'textual editing'. Therefore, LPGs have slight advantages over the RDF method in visualizing information and can be accepted more easily by audiences.

4.6 Finding

This section makes several notable contributions to the existing body of literature. From a theoretical standpoint, it provides a comprehensive understanding of the fundamental concepts underpinning ontologies, description frameworks (RDF), and property graphs (LPGs). Ontologies serve as semantic data models that define the types of entities within a domain and the properties used to describe them. RDF, on the other hand, focuses on designing an optimal data schema and description logic, aiming to strike a balance between expressiveness, computational efficiency, and reasoning soundness. LPGs, which align more closely with graph theory, concentrate on capturing relationships between entities, enabling effective information retrieval and reasoning. The study clarifies these key theoretical distinctions, enhancing the clarity and conviction of the comparison.

Furthermore, the research conducts a comprehensive five-benchmark-based comparison between RDF and LPGs. This approach enriches the current understanding of these two dominant ontology data models. Many prior studies have primarily concentrated on general comparisons for organized ontologies, often neglecting an analysis of the core ontology building mechanisms. This research surpasses such limitations by considering a wider array of benchmarks, including storage size, query efficiency, reasoning, and data visualization. While previous comparisons have focused on individual benchmarks, this study's holistic approach considers the full spectrum of necessary factors, including the valuable attributes of reasoning and data visualization, which contribute to a more informed decision-making process regarding data model selection.

From a practical perspective, the findings in Section 4 offer valuable support to researchers in the field of data management. Rather than providing only a macro-level understanding of differences, this study offers precise mathematical insights through the lens of four distinct benchmarks. Notably, the data density benchmark reveals that LPGs can significantly reduce storage size compared to RDF, with the advantage becoming more pronounced as dataset size increases. The query benchmarks, which consider the complexity of logic, underscore that LPGs perform better in retrieving direct information but struggle when querying embedded properties. In contrast, RDF excels in reasoning benchmarks due to its simplified structure, making it easier to establish reasoning logic. The study also highlights the strengths and weaknesses of both models in the context of data visualization and reasoning, filling a critical gap in knowledge.

In summary, it advances our understanding of ontology data models and provides valuable insights for decision-makers and researchers in the field of data management. Its theoretical, practical, and methodological contributions enhance our knowledge of RDF and LPGs and their suitability for different applications.

4.7 Chapter summary

This chapter compared two popular graph data models for ontologies with an AEC project background and identified the best model in different application contexts. The theoretical differences were firstly defined and highlighted. The four most focused and important benchmarks were evaluated by both quantitative and qualitive methods,

including experiments, literature reviews and focus groups. The results showed that the LPG model has advantages in saving storage space, querying direct information and visualizing data while the RDF model presents advantages int querying complex information and reasoning out vertexes and relationships. This work strongly fills the gap in current research on ontology techniques. In the future, large datasets and other developed benchmarks can be added as extensions.

5 Development of an EIAO in RAM

5.1 Introduction

The theoretical EIAO was tested and validated using a road project in Hong Kong, which had a strict requirement on environmental impact. This project was aimed to improve the road network in the West Kowloon Reclamation Development area to meet future traffic needs. The project began in 2015 and was completed in 2019. During this period, EIA monitoring results were reported based on the EIA audit manual.

5.2 Research problems

Although EIA has been previously used in many road projects, there were several problems using traditional EIA approaches. The EIA process involves numerous decision-making actions. For instance, a single hazard can have various impacts, and different actions must be determined after the monitoring process. In the past, these decisions were made by humans through group meetings. This manual process involved searching through extensive documents for the required information; therefore, making a single decision would often take an entire day.

Additionally, mobilising and storing knowledge (including standards and project documents), and comparing them with the data recorded daily is also a very complex manual process. In the traditional model, standards and regulations are decentralised. Although each project will list precautions and requirements based on the prevailing conditions, providing immediate feedback by checking textual documents is difficult. For example, some data are collected from the EIA process, and the managers need to refer to the manual to determine whether the readings exceed the limitation. Similarly, the measures taken in the following steps also need to be confirmed and discussed manually. Commercial software can help project managers to computerise these behaviours; however, because each project has its peculiarities, and entering special requirements and data will cause delays and faces troubles.

Ontology provides the capability of high storage efficiency, fast querying and responding. Its application in an EIA project can solve the above problems by providing a smart decision-making system. Furthermore, Neo4j's reasoning ability can

make preliminary identification and judgment based on numbers and keywords, and then automatically provide default solutions to the manager.

Therefore, in this chapter, we describe a structured EIA ontology which we constructed to address the aforementioned issues. This ontology would transform the knowledge required for EIA into a unified, machine-readable, and efficiency-enhancing format. Additionally, we aimed to establish an ontology-based decision-making process to assist in making decisions and judgments using this knowledge repository, based on knowledge and on-site conditions. These scenarios include the following: 1) Several stakeholders express a keen interest in understanding the methodologies employed by the management team for the supervision and regulation of the environmental impacts. 2) The project manager is particularly concerned with locating specific work procedures characterised by specific attributes. However, the manual sorting through disorganised documents proves to be a challenging and time-consuming task. Additionally, anomalies and unexpected issues might arise within the EIA system, and the manager may be eager to swiftly identify the causes and implement solutions. The conventional manual examination of documents is viewed as inefficient and prone to inaccuracies. Moreover, the project manager aims to evaluate the performance of project participants to make informed decisions regarding future collaborations. Moreover, engineers require the ability to monitor the progress of tasks and procedures against the project plans and identify instances of work delay during the course of the project.

5.2.1 Defining the ontology structure

The fundamental concepts have been directly derived from the preceding EIA definition. The figure exclusively features these essential concepts and relationships as they constitute the foundational framework, notwithstanding the existence of additional elements in the comprehensive ontology.

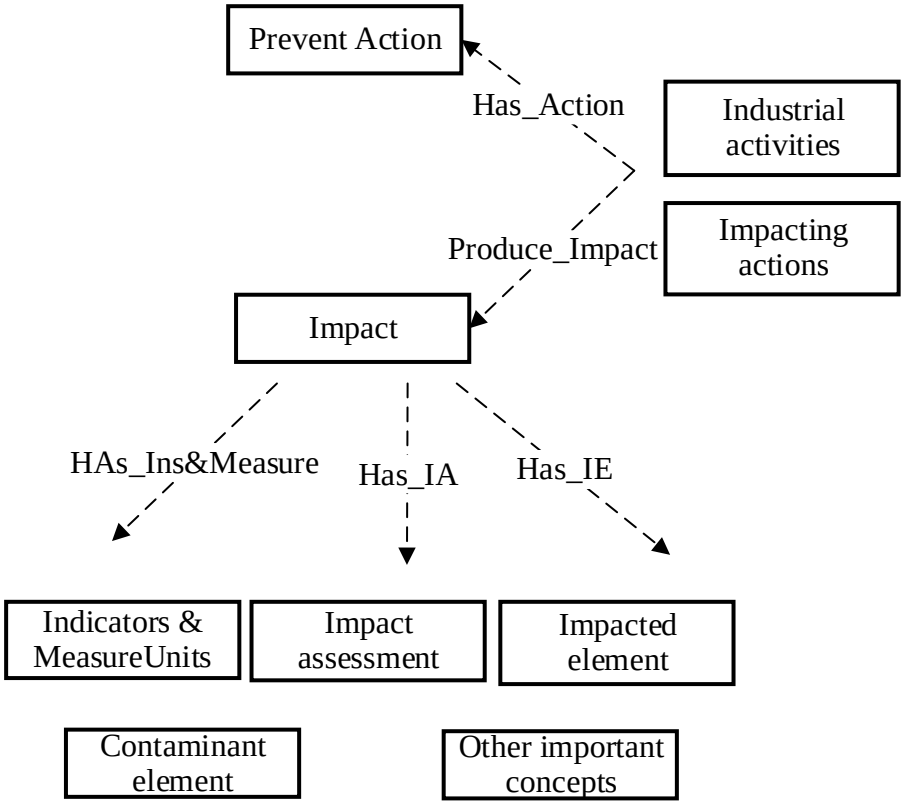


Figure 5-1 Main concepts and relationships

5.2.1.1 Impact

The standard UNE ISO 14001, defines environmental impact as any adverse or advantageous change in the environment resulting from the activities, products, or services of an organisation (Block & Markowitz, 2000). This concept encompasses various types of impacts, and they are categorised based on the environmental factors they affect. These environmental factors include the atmosphere, geophysical processes, soil, habitat, landscape, socioeconomic factors, and water (Canter & Atkinson, 2008; Casey et al., 2005).

Within the category of impact on the atmosphere, specific concepts include changes in the composition of the solid and gas phases, an increase in radioactivity, light pollution, elevated noise levels, and accumulation of odours. Additionally, phenomena, such as increased fog or precipitation and alterations in temperatures or wind circulation are under this category.

The geophysical impacts pertain to changes in geophysical processes. These include shaking, subsidence, induced seismic events, alterations in flood-prone areas,

modifications in waterway dynamics, erosion, sedimentation, hillside stability, surge propagation, coastal flows, and aquifer recharge.

Ground impacts are categorised into four main types: soil, morphology, singular elements, and mineral resources. Soil can be adversely affected by direct destruction, pollution, or alterations in its edaphic properties. Morphology impacts involve changes in topography. Singular elemental impacts are associated with geological points of interest and destruction of natural monuments. Mineral resource impacts pertain to the loss of natural resources.

Habitat impacts are categorised into three types: alteration of habitat properties, habitat direct loss, and movement interferences. Altered properties encompass changes in the composition of the biotic community, reduced vegetal coverage, critical habitat reduction, disruption of energy flow and nutrients, decreased nutriment availability, increased susceptibility to pests, decreased productivity, interference with non-hostile habitats, and short-distance movements of mobile species.

Landscape impacts encompass visual impact and changes in landscape quality.

Socioeconomic impacts encompass various aspects, such as changes in prices and taxes, economic and employment trends, demographic shifts, social and public service requirements, social communities, and alterations in land use, tourism, and leisure.

Water impacts are divided into groundwater and surface water impacts. Groundwater impacts encompass changes in the phreatic level, flow, and quality. Surface water impacts include alterations in water quality, radioactivity levels, water flows, and contributions to the basin.

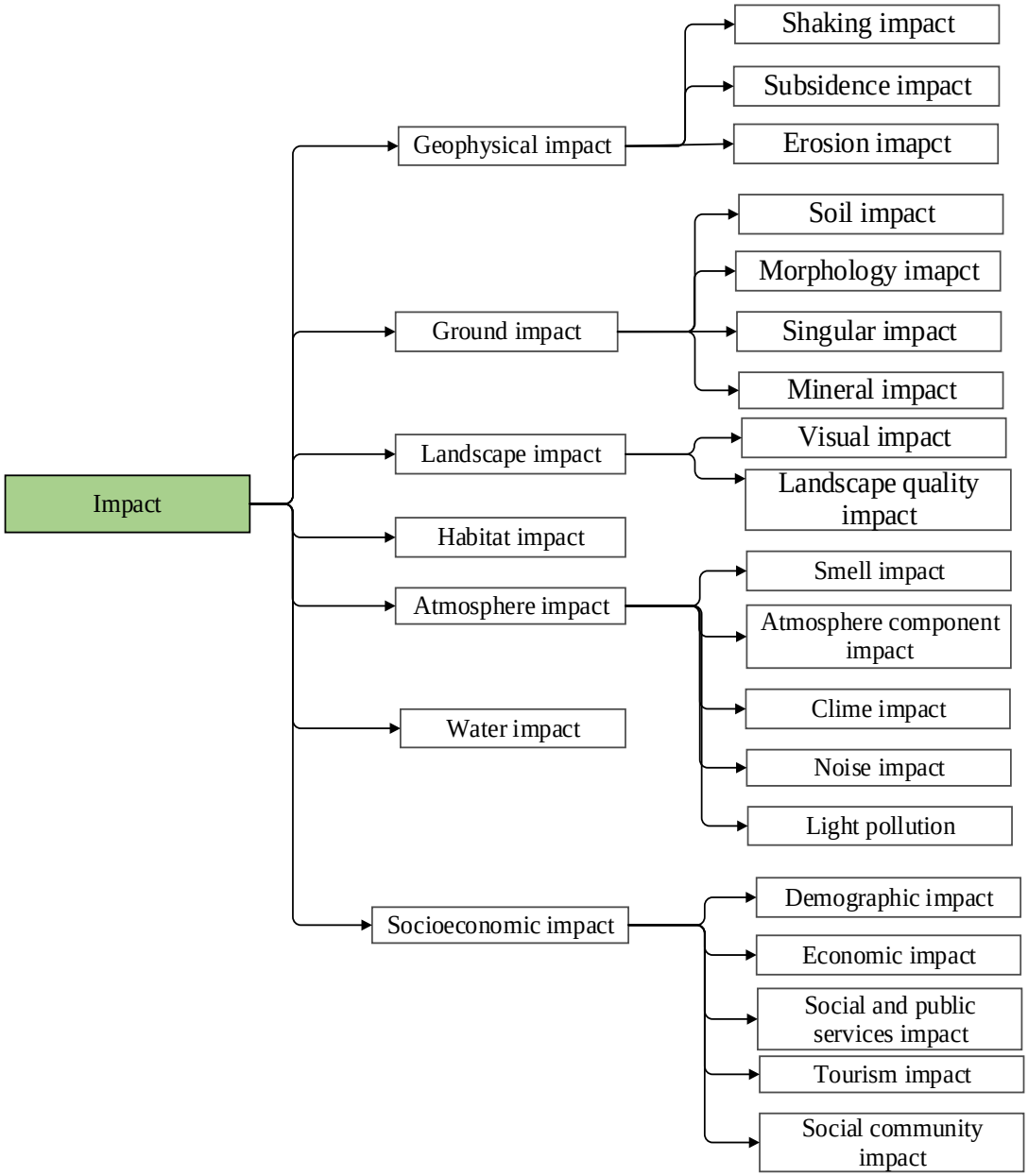


Figure 5-2 Part of the impact hierarchy.

5.2.1.2 Impacting actions

The actions affecting the environment, which serve as the causes of environmental impacts, are classified into two main categories: human actions, as described by Gómez-Sal et al. (2003), and natural processes where human intervention is not directly involved, based on the Global Change Master Directory ontology (Bermudez & Piasecki, 2004).

Owing to the widespread use of the cause-and-effect matrix in environmental assessments, various lists of actions have emerged in the literature since the 1970s.

This matrix connects project activities or actions with their corresponding environmental impacts.

Human actions, which have a direct influence on the environment, are further subdivided into specific concepts, including land alteration, traffic changes, production, soil transformation and construction, regimen modification, resource renovation, resource extraction, chemical treatment, waste treatment, and waste accumulation. Each of these concepts is elaborated below.

Land alterations encompass actions, such as erosion and terrace control, sealed mines and waste management, surface mine reclamation, dredging, and marsh drainage.

Traffic changes include alterations in the traffic patterns of railways, trucks, cable railways, fluvial and canal transport, vessels, leisure sailing, pipelines, footpaths, and communications.

Production relates to activities involving agriculture, livestock farming, vehicle and aircraft production, and the storage of products. These omitted elements have been consolidated under the broader category of industrial activities, representing a higher level of abstraction.

Soil transformation and construction encompasses a wide range of activities, including the development and construction of urban areas, land parcels, industrial buildings, airports, roads, railways, elevators, bridges, electrical infrastructure, pipelines, corridors, barriers, dredging and alignment of canals, canal lining, dam construction, port facilities, maritime structures, recreational areas, blasting and drilling, excavation, tunnel construction, and underground installations.

Regimen modification relates to actions involving exotic fauna, biological controls, changes in soil coverage, paving and smoothing, controlled burning, river management, and alterations in river flow.

Resource renovation pertains to activities, such as reforestation, conservation and nature management, use of fertilisers, and waste recycling.

Resource extraction involves actions, such as blasting and drilling, surface and underground excavation, well excavation, flow extraction, clearing and chopping, dredging, fishing, and commercial hunting.

Chemical treatment includes actions, such as chemical defrosting, soil chemical stabilisation, and weed and insect control through the use of herbicides and pesticides.

Treatment and waste accumulation corresponds to actions related to waste accumulation, emissions from exhaust pipes and chimneys, spills of liquid effluents, lubricant usage, emissions of municipal residuals, oil spills, oxidation and stabilisation ponds, refrigeration water spills, scrap disposal, underground deposits, and septic tanks.

On the other hand, natural processes themselves are not considered impacting actions; they become impacting actions when they interact with human activities. This concept is divided into various hazards, including hydrological, technological, atmospheric, geological, and biological, all of which are detailed below.

Atmospheric hazards can be categorised as either single or complex. Single atmospheric hazards include excessive rainfall, extreme temperatures, hail, heavy snowfall, radiation exposure, high wind speeds, and freezing rain. Complex atmospheric hazards encompass blizzards, glaze ice, hurricanes, heat or cold stress, thunderstorms, and tornadoes.

Biological hazards are associated with the invasion of animals and plants, epidemics, and forest or grassland fires.

Geological hazards are classified into earthquakes, mass movements (landslides), volcanic eruptions, rapid sediment movements, sedimentation, and soil erosion.

Hydrological hazards include droughts, floods, freezes, groundwater flow discharge, infiltration, land subsidence, percolation, runoff, saltwater intrusion, and thaws.

Technological hazards relate to the accidental release of toxic substances, biological, chemical, or nuclear warfare, the collapse of public buildings or other large structures, explosions, industrial fires, nuclear power plant failures, and transportation accidents.

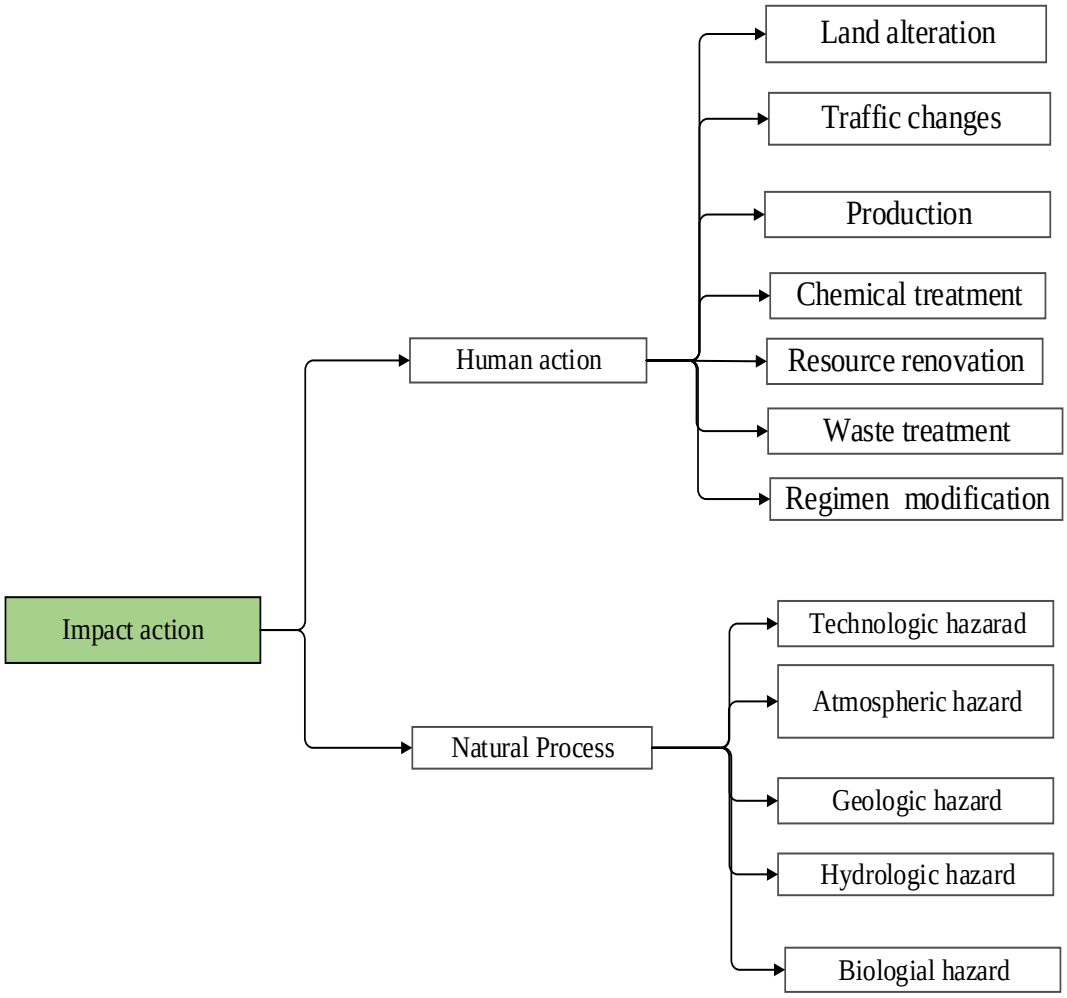


Figure 5-3 Part of the impacting actions hierarchy

5.2.1.3 Industrial activities

According to the Royal Academy of the Spanish Language, the term "industry" refers to the set of processes undertaken to obtain, transport, or transform one or more natural products. Industrial activities constitute a sector that presents its own environmental challenges and holds significant importance within EIA.

Experts recommend classifying industrial activities based on European directives, such as Directive 96/61/EC (IPPC) (O'Malley, 1999), which pertains to integrated pollution prevention and control. This classification excludes facilities or sections of facilities used for research, development, and testing of new products and processes.

As per Council Directive 96/61/EC (1996), industrial activities are categorised into those pertaining to the chemical industry, energy industry, production and metal transformation, waste management, and other industries that do not fit into the aforementioned categories. These other industries encompass activities, such as

carbon or electrographite production, milk treatment and processing, paper and paperboard production, slaughterhouses, intensive rearing of poultry or pigs, and treatment and processing of food products.

5.2.1.4 Impacted element

The elements or environmental factors are subject to the impacts generated by various activities. Several classifications exist for these environmental factors, including the European Union (EU) classification outlined in Directive 85/337 (Wood, 2000), which addresses the evaluation of the effects of specific public and private projects on the environment. This directive mandates that EIA must identify, describe, and evaluate the direct or indirect effects on various aspects, including humans, fauna, flora, soil, water, climate, air, the interplay between these factors, material assets, and cultural heritage.

While different authors may employ slightly varying classifications, the distinctions among them are not substantial. Therefore, we have amalgamated concepts from various classifications to construct a comprehensive and well-structured classification.

All environmental factors have been categorised into overarching groups, encompassing land surface, landscape, processes, living organisms, water, habitat, atmosphere, and socioeconomic elements. The classification of water has been particularly emphasised, with a more detailed focus on surface water, as it is a topic extensively discussed in the literature.

5.2.1.5 Preventive action

It refers to proactive measures and strategies taken to anticipate, mitigate, or eliminate potential problems, risks, or negative consequences before they occur. It is a fundamental component of risk management and quality assurance in various fields, including business, healthcare, and engineering. The goal of preventive action is to identify and address underlying causes or vulnerabilities that could lead to adverse events, with the aim of preventing these events from happening in the first place. This approach is often contrasted with corrective actions, which focus on addressing issues after they have already occurred.

5.2.1.6 Indicators and measure units

An environmental indicator is a metric or measure that provides information about the state of an ecosystem or its relative conditions. These indicators are used to assess

the environmental health and quality of a specific area. Biological indicators are a subset of environmental indicators and involve the presence or absence of particular plant or animal species as strong indicators of specific environmental conditions. These species are selected based on their sensitivity to or tolerance to pollution or its effects. The term indicator not only refers to the metric itself, but also encompasses the expression or formula used to calculate it. In some cases, indicators are indirectly measured using models or simulations to estimate their values. Environmental indicators play a vital role in assessing and monitoring the ecological well-being of an area.

5.2.1.7 Impact assessment

Impact assessment is an integral component of the EIA process and is typically incorporated into a technical report when conducting an environmental impact study. These environmental assessments, as outlined in Royal Decree 1131/1988 (Palerm, 1999), have been compiled and included as part of the EIA process. The purpose of these assessments is to comprehensively evaluate and document the potential impacts and effects of a particular project, activity, or policy on the environment and other relevant factors. The findings and data from these assessments play a crucial role in informed decision-making and ensuring responsible and sustainable practices.

5.2.1.8 Contaminant element

A contaminant element refers to a specific substance, compound, or chemical element that is present in an environment, substance, or material at a level that can potentially cause harm, pollution, or adverse effects. Contaminant elements are often a focus of environmental assessments and studies, for example, in the field of EIA, to evaluate their presence, concentration, and potential impacts on ecosystems, human health, or other factors. The identification and characterisation of contaminant elements are important steps in managing and mitigating environmental contamination and ensuring environmental quality and safety.

5.2.1.9 Other important concepts

Some additional essential concepts incorporated into the ontology, which are relevant to the field of EIA include methodology, environmental hazard, scene, development risk, mechanism for repairing environmental damages, environmental risk assessment, environmental impact assessment, vigilance, and control schedule.

Unlike the concepts discussed earlier, these concepts were not derived directly from the EIA definition, but are integral to EIA and its comprehensive understanding.

5.2.2 Define specific EIA decision-making by ontology

After gaining an overall understanding of the EIA knowledge of the road sector, an EIA ontology framework was established; this is presented in Figure 5-4 . Screening work was conducted to identify potential hazards. For every hazard identified, its impact on the current environment and future project was determined (e.g., the impact level). A detailed EIA plan would follow up, based on documents and project conditions. Upon the approval of the plan, EIA actions were to be taken and recorded as audit results.

Based on this flow, a linear relationship model was defined in this study, which provided a clear and convenient way to the built network (Akiho, 2002). However, they were more or less simplified for a better understanding.

Figure 5-4 demonstrates the core concepts and relationships that were directly derived from the earlier EIA definition. These foundational concepts and relationships served as the primary framework for the ontology, and although additional concepts might exist at sub-levels within the hierarchy, this figure focuses on these fundamental elements.

These relationships are crucial for enhancing the knowledge representation and enable the ontology to support queries and reasoning tasks. For instance, they facilitate inquiries related to the impacts generated by specific industrial activities or the environmental indicators employed to evaluate these activities. By structuring the ontology around these relationships, it becomes a valuable tool for comprehending and analysing the cause-and-effect relationships within the context of EIA.

Figure 5-1 presents an example of the EIAO knowledge pool. The EIA audit actions and concepts are the most comprehensive processes. Seven main auditing classes were identified, namely air quality monitoring, noise monitoring, landscape and visual impact assessment (potential impacts caused by construction activities), waste management, site inspection (environmental issues on site), mitigation action, and license management (e.g., checking of contractors' qualifications).

- Air quality monitoring: This refers to dust suppression and hazardous gas monitoring produced by activities. It includes information, such as air

monitoring equipment, location, identification number (ID), air quality parameters and quality control result.

- Noise monitoring: This refers to noise increasing levels during the project. It includes information, such as noise monitoring equipment, equipment supplier, location, and quality control results.
- Waste management: This refers to construction excavation, site demolition and general waste materials from the site. It includes information, such as normal waste (e.g., non-inert and inert waste), recycled waste, and other relevant information.
- Site inspection: This refers to regular and direct observation on-site inspection activities to ensure that the assessment activities are properly implemented. It includes information, such as location, parameters (e.g., in noise monitoring class, the decibel level is a parameter), frequency and responsibility.
- Landscape and visual impact assessment: This refers to information, such as landscape and visual impacts caused by construction activities, temporary storage of plants and materials, traffic and road diversions and dust emission. It also includes the visual category, distance, receiver, and policy.
- License management: This refers to information, such as verification of the licenses of the contractors. It includes subclasses, such as activity limitation, license status and valid period.

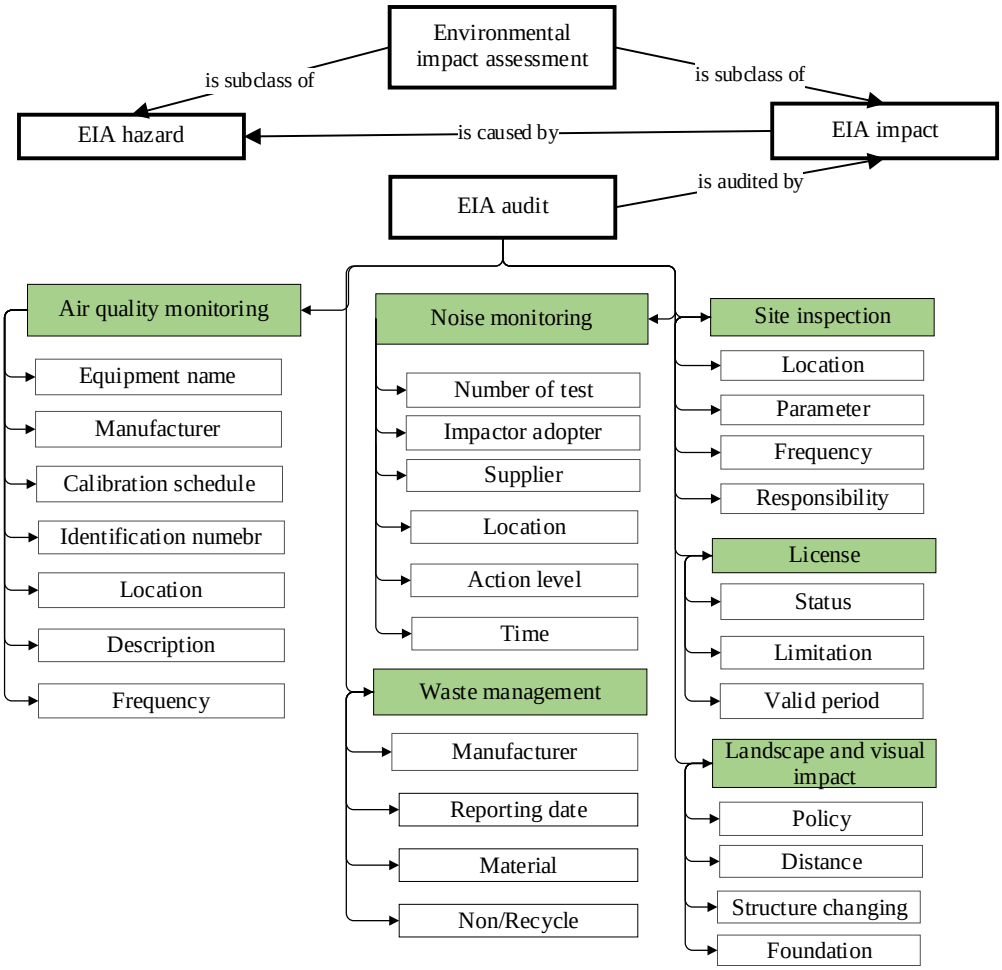


Figure 5-4 EIAO structure

5.2.3 Define ontology establishing environment and data model

Many languages and data models use different structures, such as RDF, Web Ontology Language (OWL) and LPGs. The RDF format is one of the most popular standards for establishing ontology, which is based on the expression *subject–predicate–object* (also known as *triples*) to represent relationships between instances. However, it also has limitations, such as using more storage space, lack of complex API and outdated version (Gong et al., 2018). In this research, the novel LPGs will be used as an ontology establishment model, and its popular tool, Neo4j will be the implementation environment.

LPGs are a multiple labelled graph model, which a group of Swedish computer engineers developed after the RDF was developed (Anikin et al., 2019). LPGs present the information by nodes, link those nodes by edges, and enrich them by embedded

properties. Using graph-based structures, those objects, objects, and relation types in RDF can be added to various properties more powerfully (De Abreu et al., 2013).

In LPGs, all features can be presented in one instance (node), and the link between two nodes can be named based on the relationship (edge). As an LPG can use fewer links to represent the same amount of information, it can significantly reduce querying time and deal with complex relationships (Gong et al., 2018).

Figure 5-2 shows an example of differences between RDF triples (left) and LPGs (right) for developing ontologies. An RDF model stores information separately as different instances (nodes), and every instance is linked by solid lines (which means the relationships are explicit). In this case, 'Air Quality Monitoring' has a property named 'Shortened name', 'Air monitoring location 1' has a property named 'ID', whose value is 'AM1'. Similarly, it has another property named 'Location' whose value is 'Administrative Building'. 'Air monitoring location 1' and 'Air monitoring location 2' are subclasses of 'Air Quality Monitoring', and the relationships are presented by '*Has* Subclass'. On the other hand, LPGs significantly improve the information density by coding properties into a single related node. In this model, the relationships are directly linked by dashed lines (meaning the relationships are implicit). For instance, 'Air monitoring location 1' has two properties: 'ID: AM1' and 'Location: Administrative Building', presented intuitively in one graph. A similar advantage exists in the property 'Shortened name: AM' of the instance 'Air Quality Monitoring'. The relationship '*Is Subclass of*' has the same meaning with RDF triples. However, properties can also be added to the edges of LPGs, such as the subclass level: 'Level 1'. It can be noted that, to present the same amount of information, the RDF triples use nine nodes and eight edges, while the LPGs use only five nodes (including two implicit nodes) and four edges (including two implicit edges). In addition, the query paths in LPGs are simpler and more direct. For instance, from 'Air Quality Monitoring' to 'AM1', the RDF triples need three steps, while the LPGs need only two steps (including one implicit step).

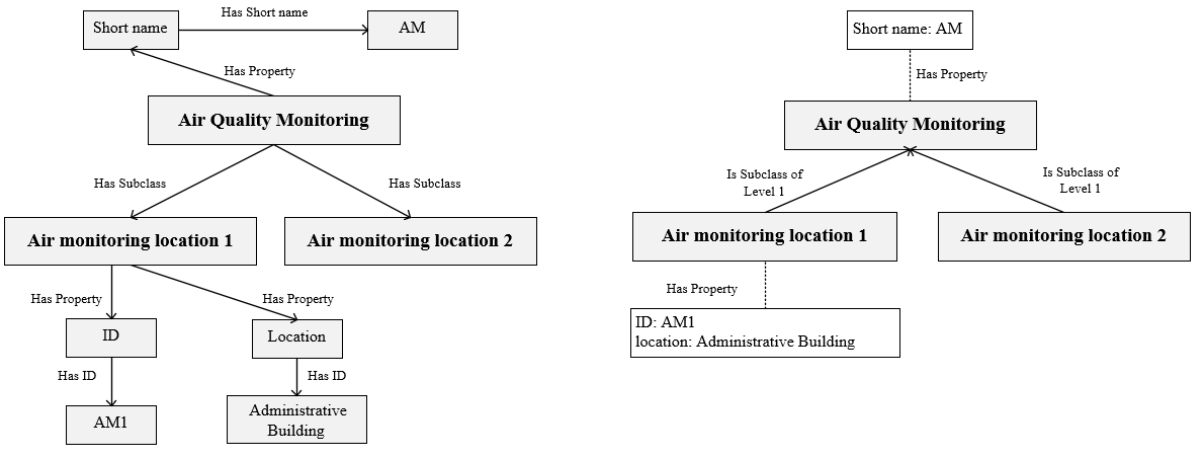


Figure 5-5 Case showing the difference between RDF triples and LPGs

5.2.4 Ontology establishment

This step aims to convert knowledge to the LPG format. In an EIA, there are four types of data: 1) drawing data, such as inspection figures; 2) tabular data, such as spreadsheets; 3) raw digital documents, such as Adobe PDF and Microsoft Word; and 4) other paper-based materials, such as maps and paper records. A four-dimensional data model is utilised to convert these four types of data into the LPG format. These four steps are related to defining the ontology instance (In), class (Cl), property (Pr) types, and relationships (R) (Zhou & Tao, 2011). The description of each dimensionality is listed below:

1) Class (Cl): All the instances shall be sorted into a complete and logic-reasonable hierarchy. It is defined from the knowledge pool and represented by navigational relationship (direction). For example, ‘Air quality location’ is a subclass of 'Air quality monitoring', which means that location is at a lower level than monitoring in this information hierarchy.
2) Instance (In): It refers to the core of ontology, also referred to as an entity in some studies. In LPGs, the concept is weakened since the richer property can be added to both instance and relationship.
3) Properties (Pr): It is an attribute that records features and properties of an instance or relationship. Features and properties are embedded within nodes, which can also be treated as implicit nodes linked to nodes. The instance would represent an EIA's main concepts and tasks, while properties would present detailed information of

each task, such as values read from monitoring equipment of each environmental issue.

4) Relationship (R): It represents the relationship between instances and shows the direction of information flow. In LPG database, the relationship (edge) links directly between instances. Property has a default relationship (*Has Property*) with its corresponding instance.

An example which can present the four dimensionalities with LPGs is shown in Figure 5-6. In this example, air quality was highlighted. The grey rectangles represent instances (nodes); the solid lines represent relationships (edges); the white rectangles represent properties contained in instances; and the dashed lines represent default relationships between properties and instances. Most of the instances, properties and relationships have been explained and listed under Figure 5-6. Compared with the last example, more detailed information has also been added to the air monitoring domain. One air quality monitoring point has these features: The property 'description' refers to a further description of the air monitoring location. The property 'parameter' refers to the measurement parameter of this location, which is the total suspended particulates per hour (1-hr TSP). The property 'Location' refers to the special geographical location on this monitoring process. Moreover, Neo4j allows one property to be presented as the prior label of the node, which to be shown in the map of nodes. For instance, the property 'ID' has been chosen; therefore, the node 'Air monitoring location 1' will be shown as 'AM1' in the overview of ontology, which helps a manager to quickly check the information he/she needs.

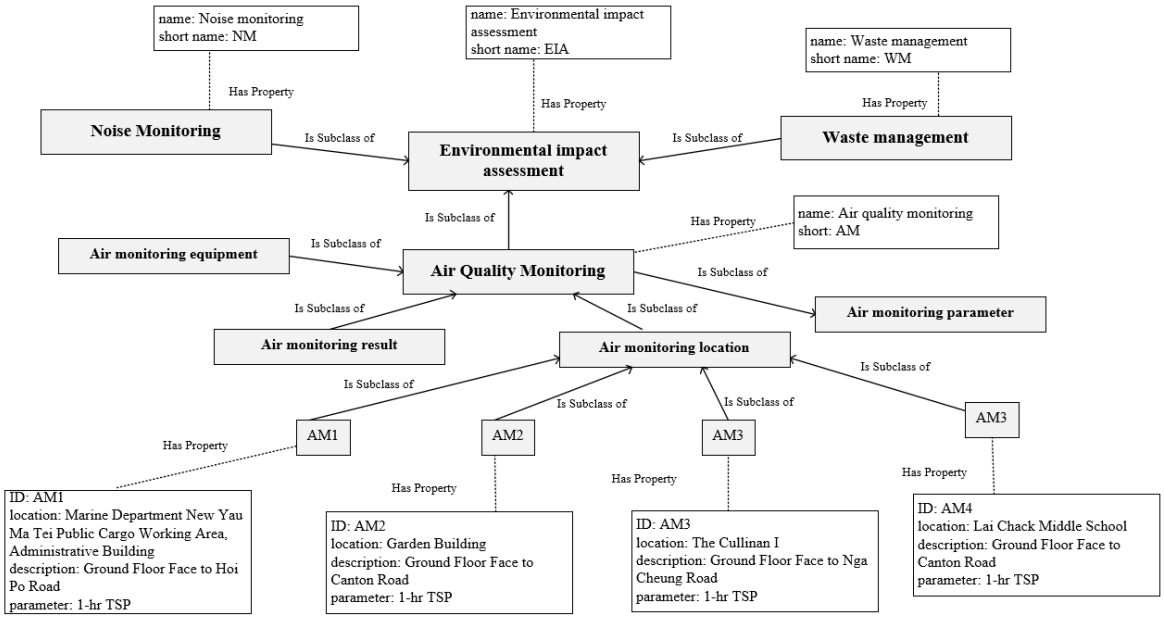


Figure 5-6 Partial case of air quality monitoring and its relevant information in EIAO

Table 14 shows the coding rules based on Cypher query language for each of the steps.

Table 14 Examples of code and description

Sector	Code	Description
Cl	Create (n: Air_Quality_Monitoring)	Create a label for the class 'Air Quality Monitoring'.
In	Create (n: Air_Quality_Monitoring {name: 'Air monitoring location'})	Create instance named 'Air monitoring location' under the class 'Air Quality Monitoring'.
	Match (n: Air_Quality_Monitoring {name: 'Air monitoring location'}) Return n	Find an instance named 'Air monitoring location' under the class 'Air Quality Monitoring'.
Pr	Match (n: Air_Quality_Monitoring {name: 'Air monitoring location'}) Set n.short =AML	Find an instance named 'Air monitoring location', and then give a property type as 'short' which means short name; the value is AML.
R	Create (n: Air_Quality_Monitoring {name: 'Air monitoring location'})-[r: is_subclass_of]-> (n: EIA {name: 'Air Quality Monitoring')	Find an instance named 'Air monitoring location' under the class 'Air Quality Monitoring' and an instance named 'Air Quality Monitoring' under the class 'EIA'; then create a relationship named 'is_subclass _of'. 'Air monitoring location' is a subclass of 'Air Quality Monitoring'. -> means the direction is from the left to the right, while r is a code to differ code n.

The framework should also have basic automatic query functions to reduce the amount of manual work, e.g., finding the shortest path from one node to the other within ms.

5.2.5 Validation and improvement

The EIAO was implemented in a standard Neo4j environment and used Cypher as the query language (Zhang et al., 2015). A case study was conducted to test the development, using two main functionalities, namely searching and reasoning information, which are the most accepted factors when assessing an ontology (Scholer et al., 2002). Table 15 lists some of the query and reasoning functions that the EIAO could provide.

Table 15 Description and used codes for evaluation

Description	Project application	LPGs codes	Number
Find all nodes	Find all information points of the project.	*MATCH (n) RETURN n*	Query 1
Find all relationships	Find all relationships between project elements.	*MATCH p=()-->() RETURN p*	Query 2
Find all classes	Find defined categories of EIA.	*CALL db.labels ()*	Query 3
Find nodes with a certain feature	E.g., Find the works monitored by One Hour Total Suspended Particulate (1-hr TSP)	*MATCH (n:{parameter:*'1-hr TSP*'}) RETURN n*	Query 4
Find relationships with a certain feature	E.g., Find the sub-class of 'Air Quality Monitoring'	*MATCH p=()-[r: is subclass of]->(n: Air Quality Monitoring) RETURN p*	Query 5
Reasoning requires nodes from an existing ontology	E.g., Find the unusual records in noise quality monitoring.	*Match (n: {result:}) AS one, (n: {limitation:}) AS two* *RETURN one > two AS result* *RETURN n, where result: 'true'*	Query 6
Reasoning requires nodes from existing ontology	The manager tries to get the critical hazards, impact and auditing information of 'Air Monitoring Location 1'	*MATCH (n:Air monitoring location {name:* Air Monitoring Location 1*}),* *p = shortestPath((n)-[*]-(Al))* *WHERE length(p) > 1* *RETURN p*	Query 7

The above query functions can help find three types of information: 1) standard, guidance, and manual book, which is the existing knowledge in EIA; 2) project record, which is flexible knowledge collected during monitoring and decision making is often involved; 3) implicit result that can be reasoned from the existing knowledge by EIAO.

Additionally, the improvement in the query time by the EIAO in Neo4j was measured and compared with that checked manually and RDF-based ontology. For each query, the time of finding the specific information in printed documents was manually recorded. As for the RDF-based ontology, the Neosemantics (N10s) plug-in enabled the LPGs to be transferred into RDF-based data, and SPARQL could achieve most of the queries for the RDF. The times in Apache Jena (a platform to implement RDF and SPARQL) were collected for both of the above.

Detailed information on the case demonstration process, data preparation and evaluation method will be presented in Section 5.3. Important criteria, such as clarity, correctness and complexity were measured in the tasks separately (Gómez-Pérez, 2001). For instance, clarity could be defined by the feedback from the application and interviews (i.e., score or word comments from managers and customers on improving the behaviour and user-friendliness). The results should be passed back to the ontology development process to optimise it. Correctness and complexity could be collected from a comparison of the ontology information with the original documents, which suggested if the ontology was missing out some data after the transfer. The final EIAO could automatically identify a related resource for a certain activity, suggest a fast-responding path, and visualise all the relevant information.

5.3 Ontology implementation

The EIAO was implemented following steps 1-6 outlined in Section 3. Nodes, classes, and relationships were defined and linked by knowledge pool and available project documents. This ontology scenario had six main EIA aspects: air quality monitoring, noise monitoring, waste management, site inspection, license management, landscape, and visual impact assessment. The partial view of the EIAO is shown in Figure 6. For example, air quality monitoring was decomposed into four work aspects: monitoring locations, equipment, and control results. The relationship between different levels was simply defined as ‘composition’. In total, 136 nodes and 135 edges were established in Neo4j.

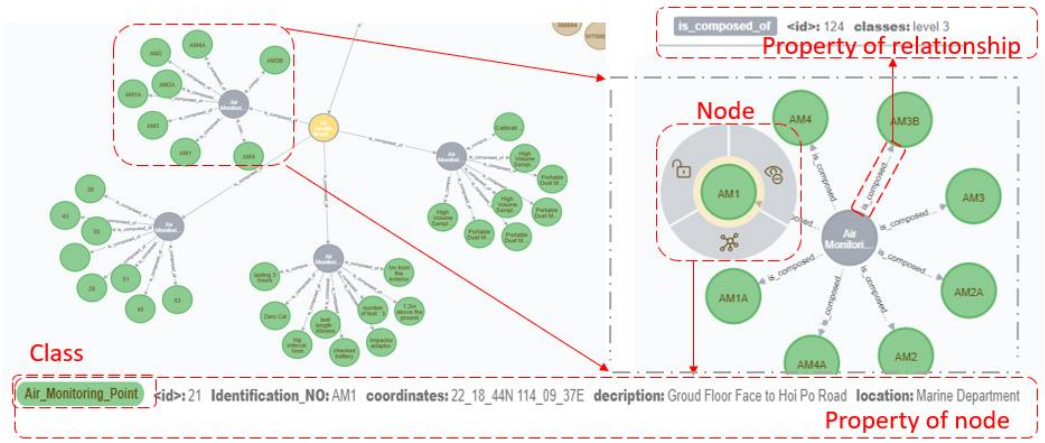


Figure 5-7 Partial view of the EIAO in Neo4j

5.3.1 Evaluation of EIA in a project:

Rule-based evaluation

The integration of the OWL API with the ontological reasoning rules served to facilitate three crucial management functions: the assessment of work progress, the evaluation of constraint statuses, and the appraisal of the performance of the participants. This integration enabled the API to export data from the ontology, allowing for the execution of intricate computations that went beyond the capabilities of conventional OWL syntax. Subsequently, the results were imported back into the ontology, where SWRL and SQWRL rules could be applied.

The decision support for the recommended strategies related to the reusability of parts relied on the reusability level of these parts. The primary objective was to provide manufacturers with information about the degree of reusability of the end-of-life (EOL) construction machinery parts. To determine the recommended actions based on the relationships among various concepts, such as *Has Part*, *Has Cause*, *Has Action*, and *Has Level*, a Jena reasoner was used in combination with custom rules expressed in the semantic web rule language (SWRL).

SWRL rules were formulated as pairs of antecedents and consequents, represented as "antecedents → consequents." Each SWRL rule stipulated that if the conditions outlined in the antecedents were met, then the statement in the consequents must also be true. The antecedents and consequents in the SWRL rules could consist of multiple elements, and a set of elements could be expressed as **e1** ⋀ **e2 ...**⋀ **en**. Variables in the SWRL rules were generally instances of classes or values of their data properties, and they were prefixed with a question mark (?).

Two fundamental elements of SWRL rules are as follows:

1. **C(?x)**: If variable *x* is an instance of class C or a value of its data property, then C(?*x*) holds true.
2. **P(?*x*, ?*y*)**: If variable *x* is related to variable *y* through property P, then P(?*x*, ?*y*) is true.

Building upon the description of the SWRL rules and analysis of classes and properties within the ontology model, the SWRL rules created for the Jena reasoner are as follows:

Rule 1: If equipment *x* has evaluation features *y*, and *y* can lead to a reusability degree *z*, then equipment *x* has a reusability degree *z*.

These SWRL rules, in conjunction with the semantic properties and the interconnections among various concepts as defined in the ontology model, facilitated the determination of suggested strategies for reusing, remanufacturing, recycling, or disposing of reuse parts. These strategies resulted from a comprehensive consideration of the ontology model and reasoning rules explained above. It is important to note that most of the rules could be applied to both procedures and tasks. However, for the sake of clarity, the examples of rules presented in the following sections are procedure-level rules.

5.3.2 Evaluation of work progress

This function is designed to assess the progress of various entities, including procedures, tasks (comprising multiple procedures), and projects (comprising multiple tasks). In practical scenarios, the duration and progress of tasks are often tracked using their start and end dates. As SWRL and SQWRL do not support temporal calculations, the OWL API was employed to extract date information from the datatype properties associated with task and procedure entities. It identified the most recent ongoing task or procedure and computed crucial metrics, such as actual and planned durations, along with the current progress performance, denoting the specific number of days ahead or behind the planned schedule for each task or procedure.

The extracted information was subsequently reintegrated into the EIAO. Within the EIAO, rules were applied to infer additional insights related to the schedule. This included identifying potentially delayed work and evaluating the total delay for a task or project. The critical rules required to execute this function are listed in Table 16.

It is important to note that within the ontological reasoning process, two relations, namely *'has-total-progress*' and '*has-current-progress*' were employed to indicate progress performance. The values associated with these relations could be either positive, signifying progress ahead of schedule, or negative, indicating a delay.

Table 16 Rules for work progress evaluation

Rule	Rule body	Explanation
1	*has-actual-duration(?p, ?ad) ^ has-current-progress(?p, ?cp) ^ start-procedure-of(?p, ?t) -> has-total-progress(?p, ?cp) ^ has-actual-duration-from-start(?p, ?ad)*	This rule computes the duration and delay of the starting procedure of a task as its total duration and delay.
2	*Procedure(?p1) ^ Procedure(?p2) ^ is-succeeded-by(?p1, ?p2) ^ has-actual-duration-from-start(?p1, ?adfs) ^ has-actual-duration(?p2, ?ad) ^ swrlb:add(?y, ?adfs, ?ad) -> has-actual-duration-from-start(?p2, ?y)*	The procedures of a task are sequential. The rules traverse them to sum the duration and progress values.
3	*latest-procedure-of(?p, ?t) ^ is-preceded-by(?p, ?pp) ^ has-actual-duration-from-start(?pp, ?adfs) ^ has-current-duration(?p, ?cd) ^ swrlb:add(?y, ?adfs, ?cd) -> has-actual-duration-from-start(?t, ?y)*	The rules evaluate the total duration and delay of the latest procedure of a task and then assign the values to the task.
4	*Procedure(?p) ^ is-finished(?p, true) ^ (has-current-progress some xsd:integer[<0])(?p) -> Delayed_Procedure(?p)*	The rules identify delayed procedures and tasks based on the progress values.

5.3.3 Evaluation of the performance of project participants

The evaluation of participants' performance primarily hinged on their proficiency in swiftly resolving constraints and successfully completing tasks/procedures. Rules were devised to execute this function, effectively identifying individuals accountable for delays in tasks/procedures and constraint removal. Moreover, these rules had the capability to rank participants based on their performance. To facilitate these rules, the API monitored the delays linked to tasks/procedures and constraint removal for each project participant, calculating their respective performance. This process is depicted

in Figure 5-5(d), where the delay was computed utilizing the functions delineated earlier. Subsequent rules were then established to enable the selection of participants based on specific performance criteria. The critical rules needed to achieve this function are outlined in Table 17.

Table 17 Rules for participant performance evaluation

Rule	Rule body	Explanation
6	*is-supervised-by(?p, ?pp) ^ Delayed_Procedure (?p) -> Participant_With_Delayed_Procedure(?pp) ^ sqwrl:select(?pp, ?p)*	The rules find participants who fail to complete work.
7	*Constraint(?c) ^ to-be-removed-by(?c, ?pp) ^ is-timely-removed(?c, false) -> Participant_With_Delayed_Constraints(?pp)*	
8	*has-constraints-removal-performance(?pp, ?cp) -> sqwrl:select(?pp, ?cp) ^ sqwrl:orderBy(?cp)*	The rules compare the delay of participants in delivering the work; then rank the participants based on their performance.
9	*has-constraints-removal-performance(?pp, ?cp) ^ swrlb:largerThan(?cp, 0.9) -> Good_Participant(?pp)*	The rules select participants based on their performance and certain thresholds.
10	*has-work-performance(?pp, ?wp) ^ swrlb:largerThan(?wp, 0) -> Good_Participant(?pp)*	

5.4 Scenarios

To validate querying and searching functions discussed in Section 3.7, we investigated by five scenarios typically encountered in real-world situations with the EIAO.

5.4.1 Scenario 1

Some stakeholders wanted to know how the management team monitors and controls all environmental impacts. However, they had limited time and engineering knowledge to read professional documents manually. Managers could use the EIAO to query the required information easily by using Cypher. Moreover, graph-based information in Neo4j Browser also provided a clearer format.

- Query 1 (Figure 5-8a) shows all the monitoring activities as nodes in Neo4j, such as air monitoring locations and equipment information. In addition, the manager can simply click each node to see the detailed properties (e.g., Identification number and Calibration Date) in the interface.
- Query 2 (Figure 5-8b) shows all relationships among the information in the project, which gives a hierarchical presentation to the reader. The level of sub-class can be reviewed as the property. It should be noted that, when querying relationships, Neo4j would return a graph containing both the relationships and linked nodes. A table review function was also provided for the manager to review the relationship separately, if needed, as shown in Figure 13b.
- Query 3 (Figure 5-8c) shows all classes (which are named 'label' in Neo4j) identified and sorted during the EIA process. It helps the stakeholders to understand variouf works in the project.

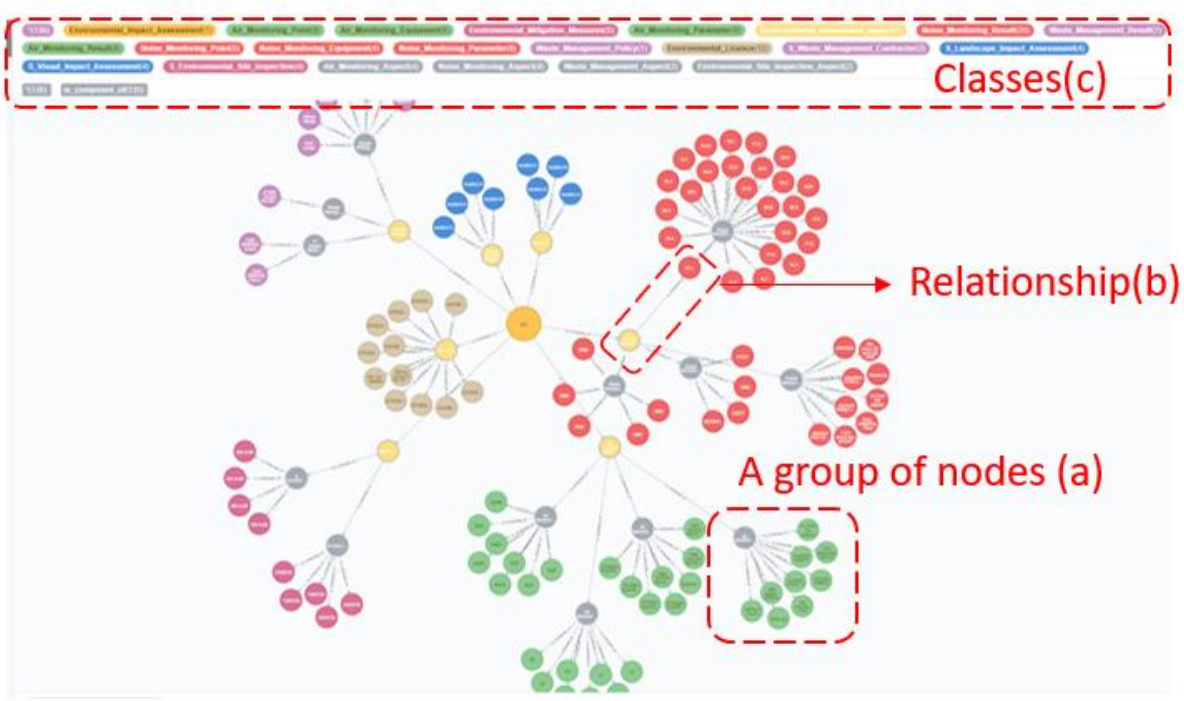
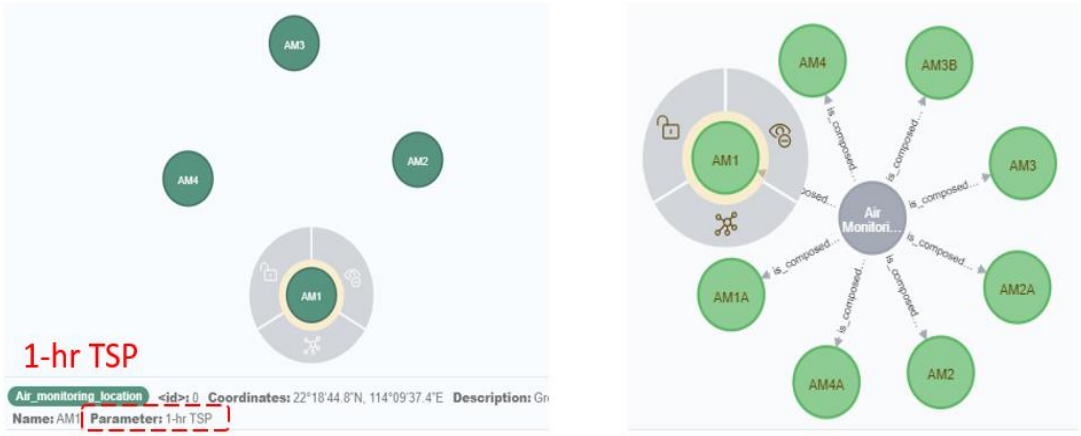


Figure 5-8 Querying results for Scenario 1

5.4.2 Scenario 2

The project manager wanted to find a work procedure with a certain feature; however, it was hard to find out in the mass of unsorted documents manually. Thus, the EIAO was implemented and used to search for the required information.

- Query 4 (Figure 5-9a) shows all the air monitoring works monitored by ‘One Hour Total Suspended Particulate’ (1-hr TSP) during the projects. The assessment methods were defined by the HK EIA manual, which set different monitoring frequencies by distance and site conditions. In total, four locations were found as 1-hr TSP measuring positions, which were labelled AM1–4.
- Query 5 (Figure 5-9b) shows all locations for air monitoring in ‘Air Monitoring Location’ class, and the director can have an overview of these locations and find out who is responsible for the site. There were eight monitoring locations during the whole project, and properties, such as road name, building name, floor number and measurement parameter were also available.



a b

Figure 5-9 Querying results for Scenario 2

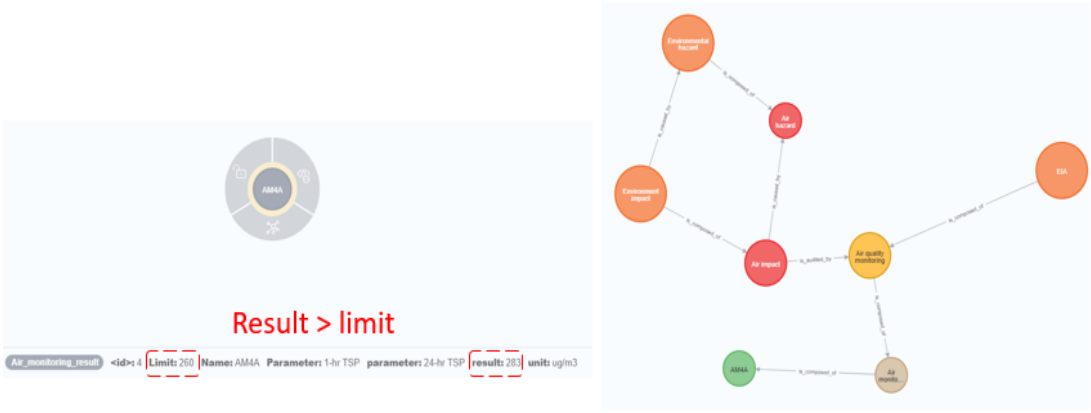
5.4.3 Scenario 3

Some unusual issues were recorded during the EIA system, and the manager wanted to find the reason immediately and solve the problem. Traditional manual checking could be inefficient and inaccurate. The EIAO could provide a computer-based information searching approach and enable the software to reason out some information from the existing databases.

- Query 6 (Figure 5-10a) shows all the observed records that exceeded the air quality limit. This limitation was defined by the EIA manual book and set in the nodes as a property. If the reading was higher than limited, the assessment point would have an impact on the surrounding environment. By this code, Neo4j could automatically compare values under ‘results’ and ‘limitation’, then show the node

whose ‘results’ value was higher than ‘limitation’ value. ‘AM4A’ is found to fail to keep the air quality within the standard level, and immediate response was taken on the construction site, for example, enhancing the cleaning activity and monitoring frequency.

- Query 7 (Figure 5-10b) shows critical steps from the beginning node to the found node with the problem in the last query. The manager could use this code to find the shortest path from one node to node 'AM4A', which could help the team find the fastest way to solve the problem because the causes might exist in all the relevant nodes.



a b

Figure 5-10 Querying results for Scenario 3

5.4.4 Scenario 4

During the project's execution, it became necessary for the engineer to monitor the progress of tasks and procedures in comparison to the original plans, particularly in identifying any delays. On September 15, 2019, the engineer focused on checking the progress of the ongoing task, which was deck paving. The initial EIAO solely contained static information and could not support progress tracking.

To address this limitation, the OWL API was used to extract date-related information from the tasks and procedures. This information was employed to calculate the duration and progress values, which were then integrated back into the EIAO. Rules 1–4 in Table 16 were executed, leading to the inference of additional progress information for the tasks and procedures, highlighted in yellow in the table.

As a result of this process, the engineer determined that the monitoring point AM3 had a delay of 0.5 h per day, which caused the total project to go behind the schedule

by seven days. Notably, the first three procedures contributed seven days each to this delay, resulting in a total task duration of 14 days. Furthermore, any delayed tasks or procedures were automatically identified through the execution of Rule 5 in Table. The process is visualised in Figure 5-11, with information computed by the EIAO highlighted in red boxes.

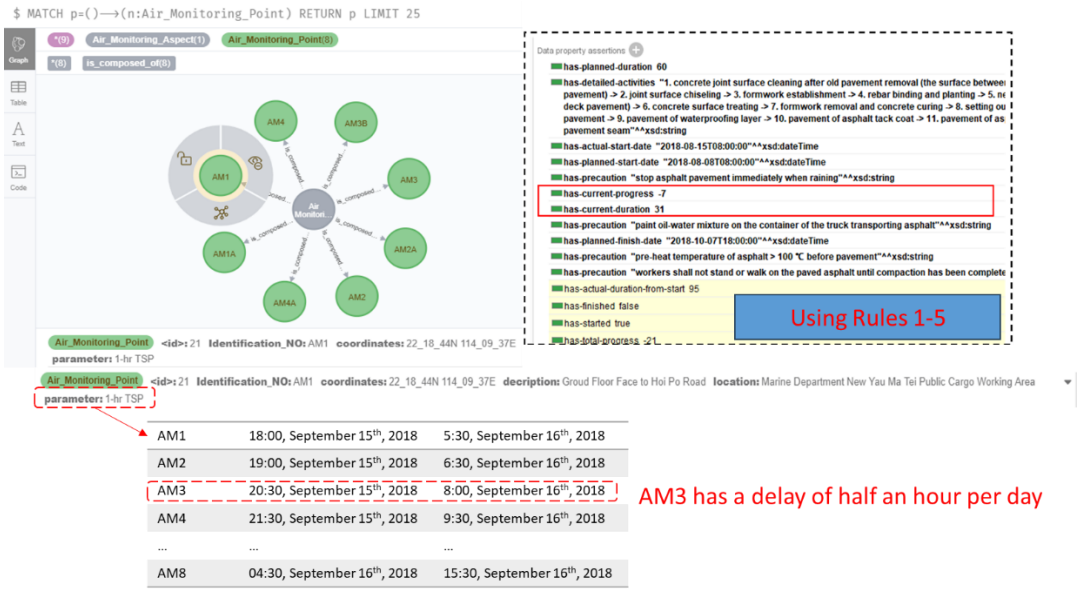


Figure 5-11 Evaluation and inference from procedure's progress

5.4.5 Scenario 5

The manager owner sought to evaluate the performance of project participants to inform future collaborations. To achieve this, the owner could utilise Rules 6-10 from Table 17 to identify participants associated with delayed tasks/procedures or constraint removal. Additionally, to assess specific participant performance, the OWL API computed two crucial metrics: the total delay in delivering tasks/procedures and the ratio of timely constraint removal (i.e., the number of constraints removed on time compared to the total number of constraints for which the participant was responsible). The outcomes enabled the application of Rules 8 and 9, as listed in Table. Conversely, government agencies, such as the building and construction authority, responsible for granting construction approval, and the Department of Transportation (DoTs), responsible for granting road closure approval, exhibited poorer performance in constraint removal. This suggested that additional buffer time should be allocated to these participants.

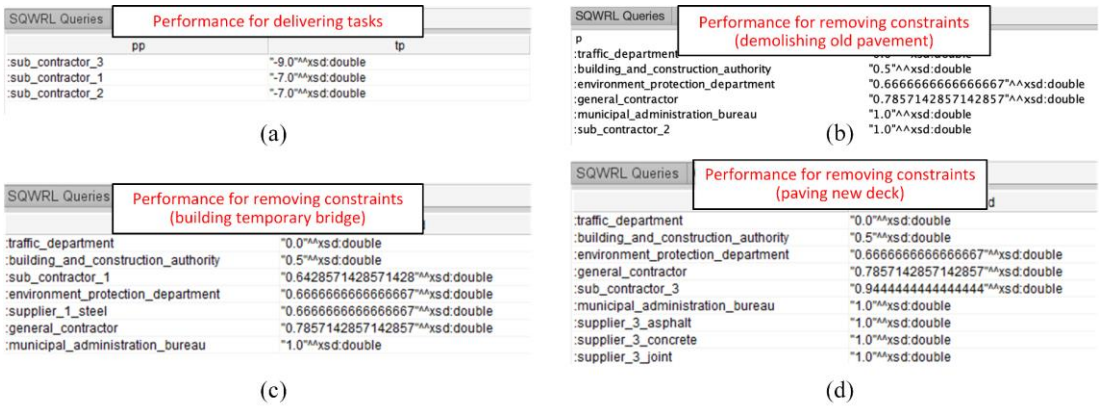


Figure 5-12 Comparison of participant performance

5.5 Results

From the system level, EIAO solved several problems encountered by traditional methods. Firstly, it integrated the existing standards and specifications and could even connect the requirements of multiple countries, which had stronger reference and flexibility. At the same time, it could also add the experience of experts as part of the knowledge pool, which helped the manager make more professional decisions. At the same time, this knowledge pool also saved cumbersome steps, such as printing and storing documents. In a traditional EIA project, the number of pages of information runs into thousands, and the electronic version of the documents also reaches several GB. Moreover, fewer human resources could be used to collect and search for the required information. In this case, only one person was needed to operate the EIAO in neo4j, and the occupied space was very small. A simple comparison in Table 18 shows that EIAO had greatly improved the knowledge storage.

Table 18 Information resources and comparison of searching time.

Method	Pages used for paperwork	Electronic storage size	Human resources used	Time for tiding up
Traditional	10,000+	10 GB+	10 persons	1 month
EIAO	100-150	60 MB	1 person	3 days

Secondly, EIAO made it smarter for the decision-making process by quickly providing the relevant information so that the decision-maker could deliver a reasonable judgment immediately. In addition, its reasoning ability based on numbers and keywords could automatically output some recommended actions from the default action pool in nodes. Thus, it reduced thinking time for managers, and workers could even act directly according to the action given by the EIAO without instructions.

Different amounts of knowledge and efficiency provided in each scenario and query are presented in Table 19.

Table 19 Information resources and comparison of searching time.

Scenario	Query	Manual book	Project record	Implicit knowledge	Manual checking (s)	EIAO (s)
1	Q1	√	-		1152	0.03
	Q2	√	-	-	5489	0.04
	Q3	√	-	√	3567	0.01
2	Q4	√	-	-	156	0.06
	Q5	√	-	√	1278	0.08
3	Q6	√	√	√	834	24
	Q7	√	√	√	1592	6
4	-	-	√	-	3152	35
5	-	-	√	-	3024	34

In the EIAO, querying general information was much faster than manually checking the result. For example, finding the monitoring parameter with 1-hr TSP (Q4) would take 156 s to find them in daily reports, while in the EIAO it only needed 0.06 s to run the prepared code and return the same results. On an average, the EIAO largely reduced the searching time from 1827.3 to 4.3 s, while retaining 100% accuracy compared with human effort. On the contrary, the time consumed by the BRMO was much more stable, as information was integrated in the KBs. Besides, Scenarios 3–5 involved semantic reasoning based on domain knowledge not explicitly mentioned in the documents. Therefore, in some cases (e.g., Scenario 5), it was impossible to obtain the information merely using manual searching.

5.6 Finding

This investigation offers a multidimensional contribution to the existing body of knowledge. Firstly, from a theoretical perspective, it delved into the origins and essence of ontology, RDF, and LPGs. Ontologies function as semantic data models defining the types of entities within a domain and the attributes used to describe them. The focus of the RDF centres on devising an optimal approach (data schema and description logic) to strike a balance between expressiveness, computational efficiency, and logical soundness. In contrast, the LPGs, closely aligned with graph theory, aim to depict relationships among entities explicitly, enabling an effective and efficient information retrieval and reasoning, rather than standardising the concepts. While both

the RDF and LPGs employ graph-based information structures, the key distinctions lie in the LPGs' capacity to include additional information in edges and properties within the nodes without necessitating an additional structure. This comprehensive exposition of the fundamental differences from theoretical sources enhances the clarity and persuasiveness of the comparative analysis.

Secondly, this research undertook a benchmark-based comparison, enhancing the current understanding of the RDF and LPG methods as the predominant data models for ontologies. Previous studies have often focused on general comparisons of organised ontologies, overlooking core ontology construction mechanisms. Additionally, some improved studies have only considered one or two benchmarks (e.g., storage size and query efficiency) in their experiments and analyses, failing to account for all essential factors when selecting an ontology model. As discussed in Section 2, reasoning and data visualisation are valuable features that facilitate information flow for both machines and human audiences. While some prior work has explored the implementation of functions, others have compared these two indicators and delivered direct conclusions on model selection. Moreover, this systematic comparison augments the existing knowledge of data model disparities and extends the visualisation capacity for ontology data model comparison methods, thereby providing a comprehensive approach by cataloguing the most valuable benchmarks.

From a practical perspective, the results in Section 4 offer valuable insights for data management researchers. Rather than providing a macro-level understanding of differences, this study furnishes mathematical insights from five benchmarks for two prominent ontology techniques. In the data density benchmark, LPGs demonstrate the potential to save over two-thirds of storage space compared to RDF, with this advantage increasing as dataset size grows. While previous research has arrived at similar conclusions, this study introduces a uniform factor (ρ) for straightforward and professional measurement. In querying benchmarks, the study delves beyond basic queries and introduces varying levels of logic difficulty, enabling a deeper analysis. LPGs excel in searching for direct information, such as all vertices. However, they encounter challenges when seeking embedded properties, as this necessitates additional logical analyses. In contrast, the RDF exhibits a notable advantage in reasoning benchmarks. Its simple structure may offer lower efficiency in other contexts, but it facilitates the establishment of reasoning logic. A series of qualitative

analyses confirmed that the LPG-based approaches offered enhanced functions and performance in visualising data, while RDF-based approaches necessitate the use of multiple tools to achieve similar functionalities. By comparing both reasoning and visualisation, this study filled a critical gap, as singular analyses struggled to provide a comprehensive support. Moreover, the distribution of dataset sizes was thoughtfully designed, enhancing the experiment's logicality and integrity. Consequently, the experimental results inspired confidence and yielded valuable evidence for selecting the most suitable model.

Thirdly, this study was conducted within the context of a bridge maintenance project, providing valuable insights into the application of ontology in the architecture, engineering, and construction (AEC) industries. Prior research has predominantly focused on computer science performance, overlooking the ultimate goal of ontology—to enhance data management and information processing for practical, everyday use. Consequently, the engineering scenarios considered in this study represented diverse and complex real-world situations, offering a holistic perspective. While the work involved in EIA for RAM is substantial, it currently lacks a computer-based approach to streamline management processes. This research successfully encompassed domain knowledge related to environmental conditions in road projects, encompassing knowledge, tasks, procedures, and project information of participants. By enabling computers to replace labour-intensive manual work, this approach not only conserves storage space and reduces costs but also enhances accuracy.

Furthermore, this research selected LPGs as the data model, challenging the conventional use of RDF/OWL for ontology development. The findings illustrated that the LPG-based ontologies offered superior querying speed (with an average improvement of 30%) and superior data visualisation capabilities. This choice diversifies the landscape of ontology development methods and underscores the specific strengths of LPGs.

Ultimately, this research presented a comprehensive approach to environmental impact assessment, addressing challenges related to knowledge integration, real-time data updates, and decision-making. While the study has made significant contributions, it also has its limitations. The knowledge pool utilised in building the ontology was manually collected and organised. Future research could focus on automating the ontology development process using machine learning or big data applications.

Additionally, while the ontology exhibits reasoning capabilities, these could be further enhanced through the incorporation of supplementary computer programming languages and APIs. An individualised user interface, whether web-based or software, could minimise the training required for end-users, such as project managers, to effectively utilise the ontology.

5.7 Summary

An EIAO was designed to integrated comprehensive domain knowledge by reviewing relevant standards, guidance, project documents and studies. The EIA management progress covered environmental hazards, impact and audit stages. By establishing an ontology with the LPG data model, information was stored in a digital environment as nodes and linked by edges to represent their relationships. Thus, the EIAO could support not only static information searching but also continuous integration, reasoning, and updating information in ongoing projects. The EIAO was validated in a real-world case, which proved that the proposed ontology could efficiently search for information (e.g., target step of work) along the project progress. Moreover, when new project information is automatically reasoned out by ontology, it can realise essential functions for project manager such as finding the critical workflow and calculating basic delay, which provides necessary resources for smart decision-making. The LPGs-based ontology also improved the storage structure and querying efficiency comparing with traditional management approach or the RDF-based ontology. The EIAO extended the current EIA ontology research in the RAM field. Besides, it proposed an LPG-based ontology, which provided efficient storing, searching and reasoning information.

6 Design of automatic information extraction model for special data in RAM

6.1 Introduction

This chapter provides an overview of the experimental results of the ATIEM for table data. ATIEM was specifically developed for identifying table information in ontological RAM KB. The experimental results are presented to demonstrate the actual behaviour of the RoBERTa method in tabular data semantics. Additionally, a case study has been conducted to showcase the practical usefulness of the ATIEM. The model was implemented using Python 3.7 and Neo4j 4.5. The training, validation, and testing of the model were performed on the Google Colab cloud computing platform.

6.2 Overall design of the ATIEM model

ATIEM is a model that processes an input table with M rows and N columns (and generates embeddings for each cell of the table, denoted by Xi,j, where i and j represent the row and column indices, respectively. Additionally, the model produces embeddings for each column (Cj) and row (Ri) of the table.

Initialisation: the embeddings for each cell in a given M × N table are initialised using a pretrained RoBERTa model, where the contents of each cell are fed into RoBERTa and the dimensional [CLS] token representation is extracted. This is important because many tables contain cells with long-form text, and RoBERTa is able to encode some degree of numeracy, making it useful for representing cells with numerical content. The RoBERTa encoder is kept fixed during training to save computational resources. Additionally, learned positional embeddings are added to each [CLS] vector to form the initialisation of Xi,j.

Contextualising the cell embeddings: The uncontextualised cell embeddings obtained from RoBERTa are computed in isolation from all the other cells in the table. Some methods such as TaBERT and TaPaS contextualise cell embeddings by linearising the table into a single long sequence (Lee et al., 2017). However, in this study, a different approach is taken for computationally manageable results. A row Transformer is defined to encode cells across each row of the table, while a column

Transformer does the same for columns. This allows for contextualisation of the embeddings while avoiding the computational burden of linearising the table.

At the core of ATIEM 's cell contextualisation approach is the use of row and column Transformers. Suppose we have a table with rows containing cell embeddings, $X1,j$, $X2,j$, … , Xm,j and columns containing cell embeddings , $C1,j$, $C2,j$, . . . , Cm,j. We apply self-attention to the embeddings of each row and column, producing contextualised output representations. This results in a row embedding and a column embedding. for each cell (i,j) in the table. To enable contextualisation across the whole table, we combine these embeddings by averaging them at each layer of ATIEM. By doing so, subsequent layers of the model have access to information from both rows and columns, allowing for a more comprehensive representation of the table. Specifically, the cell embeddings at layer L of TABBIE are computed by averaging the row and column embeddings.

Extracting representations of an entire row or column: In order to capture the contents of entire rows or columns in ATIEM, the row and column Transformers are modified to include special tokens. Specifically, [CLSROW] and [CLSCOL] tokens are prepended to the beginning of each row and column, respectively, during the preprocessing step. By doing so, the Transformers generate representations that incorporate the overall information of the row or column. During pretraining, the final-layer cell representations of these [CLS] tokens are extracted. These representations can then be utilised in downstream tasks, such as retrieving similar columns from a vast dataset of tables based on a query column. By incorporating these special tokens and extracting the final-layer cell representations, ATIEM enables the utilisation of comprehensive embeddings that capture the content of entire rows or columns in various table-related tasks.

6.3 Pretraining

Moving on to ATIEM 's training objective, we adopt the self-supervised ELECTRA objective proposed by Clark et al. (2020) for text representation learning. This objective involves applying a binary classifier to each word in a text and determining whether the word is part of the original text or has been corrupted. While the ELECTRA objective was initially developed for more efficient training compared

to RoBERTa's masked language modelling objective, it is particularly well-suited for tabular data.

RoBERTa is a language model introduced by Liu et al. (2019), which based on the BERT (Bidirectional Encoder Representations from Transformers) architecture, which is a popular model for NLP tasks. RoBERTa builds upon BERT by applying various modifications and training techniques to improve its performance.

The key differences between RoBERTa and BERT lie in the training methods. RoBERTa is trained on a larger corpus of unlabelled text data, using dynamic masking patterns and longer sequences, resulting in a more comprehensive language representation. It also benefits from advanced pre-training techniques such as using larger batch sizes, training for more iterations, and removing the next sentence prediction objective.

RoBERTa has achieved state-of-the-art performance on various natural language understanding benchmarks and tasks, demonstrating its effectiveness in tasks such as text classification, named entity recognition, sentiment analysis, and question answering. It has become widely adopted in both academia and industry for a wide range of NLP applications.

Based on these features and requirements, RoBERTa was selected as model framework. It involves randomly masking a word in a table and then using the remaining known words to predict the masked word. The transformer model parameters are updated through backpropagation and gradient descent based on the difference between the predicted and actual words.

In the context of tabular data, detecting corrupted cells is a fundamental task in table structure decomposition pipelines (Raja et al., 2020; Tensmeyer et al., 2019). Incorrectly predicted row or column separators, as well as cell boundaries, can lead to corrupted cell text. ATIEM extends the ELECTRA objective to tables by employing a binary classifier that takes a final-layer cell embedding as input to determine whether the cell has been corrupted.

6.4 Automatic ontology establishing

After obtaining the triples in the table, we need to import them into the original EIAO. Here, we use the py2neo API, which allows users to use Python as a

programming language in the neo4j database, and then automatically write the table data to Neo4j.

6.5 Model experiments

6.5.1 Experiment data collection and pre-processing

Triples stored in EIAO model were gathered for training and testing the ATIEM model. However, in this section, triples need to be extracted from tables in road asset materials. To ensure effective training, tables with and without lines drawn are both simplified into Microsoft Excel format.

6.5.2 Training and validation

Before experiments, the ATIEM model was also trained using the training and validation datasets before it was evaluated in the testing dataset. There are many mature pre-trained transformer models from an open-source repository, which has been trained on a billion-scale corpus (i.e., BERT and RoBERTa). This type of pre-training is typically performed using self-masking for unsupervised learning.

The core task of this step is to: 1) Automatically determine whether the text in the table is a header or a value, and to 2) Automatically establish semantics based on the order of header and table values. A case to run the prompt is showed as follows:

```
'''processing data and developing the prompt'''
samples = prompting(data_df, 'the term ', ' is a ', ' in a Table', bert_tokenizer.mask_token)
```

In this case, the term "time average" serves as a placeholder or mask for a specific word in the input text. This masked word is then predicted by a pre-trained model. For binary classification, the model predicts whether the masked word corresponds to a header or not. For multi-class classification, the model predicts the specific type of header or table value that the masked word belongs to. This scenario represents a typical Named Entity Recognition (NER) prompt problem, where the goal is to identify and classify specific entities within the table based on the predicted values for the masked words.

A simple validation is adopted to check the accuracy for identify the header and corresponding values after applying RoBERTa and pre-trained data.

6.5.3 Automatic triples inputting

After obtaining triples from tables, the last step is inputting them into EIAO and linked with the previous triples if they are describing the same term. In this experiment, py2neo was used as bridge between text type and LPGs type. By running certain code in Neo4j, entities would be automatically created and linked by relations. In this case, a simplified relationship definition was adopted, and the experiment only considered one type of relationship between nodes: The 'header' 'has_a_value_of''value', where 'header' and 'value' were automatically extracted from last step.

The evaluation method adopted a comparison with other automatic information extraction model by F1 score. The F1 score is a measure of a model's accuracy, particularly for classification problems. It combines precision and recall into a single value, providing a balanced assessment of a model's performance. The F1 score is especially useful when dealing with imbalanced datasets, where one class significantly outnumbers the other.

6.6 Data collection

To ensure consistency and convenience, the ATIEM employed the same data from Chapter 5. In total 78 tables have been collected and used in the model. The provided table exemplifies one of these charts, specifically depicting air pollution indicators. Through pre-training, the model gained the capability to automatically classify the headers and values within the chart, effectively mapping them to triples. Moreover, an additional experiment was conducted to assess the model's ability to differentiate between distinct headers. Furthermore, a test was performed within the EIAO system to evaluate whether the model could autonomously import pairs of data from the chart and establish connections with the original ontology, thereby enhancing the scalability and comprehensiveness of EIAO. There are total of 81 tables served as input data for the pre-training process.

Table 20 Table for Hong Kong Air Quality Monitoring Objectives

Pollutants	1 h	8 h	24 h	3 months	1 year
Sulphur Dioxide (SO2)	800	-	350	-	80

Total Suspended Particulates (TSP)	500	-	260	-	80
Respirable Suspended Pariculates (RSP)	-	-	180	-	55
Nitrogen Dioxide (NO2)	300	-	150	-	80
Carbon Monoxide (CO)	30000	10000	-	-	-
Lead (Pb)	-	-	-	1.5	-

6.6.1 Dataset generation

In the context of ELECTRA, a distinct generator model is trained using RoBERTa's masked language modelling objective. This generator model is responsible for producing candidate corrupted tokens. For example, given the sentence "Jane went to the [MASK] to check on her experiments," the generator model might generate corrupted candidates like "lab" or "office." It should be noted that simpler corruption strategies, such as randomly sampling words from the vocabulary, are not effective in inducing strong text representations. This is because local syntactic and semantic patterns are typically sufficient for identifying obvious corruptions.

However, in the case of tabular data, we demonstrate that simple corruption strategies, which leverage the intra-table structure, can indeed generate powerful representations without the need for a separate generator network (as shown in Figure 6-1). Specifically, we employ two different corruption strategies:

Frequency-based cell sampling: In this strategy, corrupt candidates are sampled from the training cell frequency distribution. Cells that occur more frequently in the training data are sampled more often compared to rare cells. However, it is worth noting that this method can sometimes generate samples that violate the specific column type. For example, it may sample a textual cell as a replacement for a cell in a numeric column. Despite this drawback, our analysis in Section 4 demonstrates that this strategy alone yields strong performance on most downstream table-based tasks. However, it does not result in a comprehensive understanding of intra-table semantics.

Intra-table cell swapping: To promote the learning of fine-grained distinctions between topically-similar data, we employ a second corruption strategy. In this strategy, corrupted candidates are generated by swapping two cells within the same table. This task is more challenging compared to the frequency-based sampling

strategy, particularly when the swapped cells are within the same column. Although this strategy may not perform as well as frequency-based sampling on downstream tasks, it qualitatively leads to a greater semantic similarity among the nearest neighbours of column and row embeddings. By incorporating this strategy, we aim to enhance the model's ability to capture nuanced relationships within the table data.

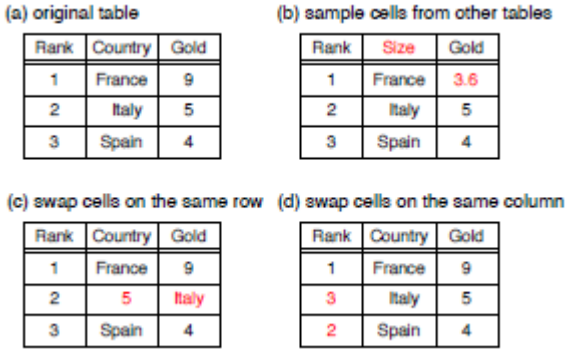


Figure 6-1 Different cell corruption strategies used in the experiments

6.7 Results

6.7.1 Fine-tuning ATIEM

In all of the downstream experiments, the same fine-tuning strategy is applied to both ATIEM and TABERT. A subset of their final-layer representations, specifically the cell or column representations corresponding to the tabular structures used in the downstream task, is selected. These representations are then used as input for a classifier to predict the training labels. Task-specific hyperparameters are selected based on the size of each dataset, and the test performance of the best-performing validation checkpoint is reported. It is important to note that the downstream error signal is back-propagated into all parameters of the model, and the pretrained model is not "frozen".

6.7.2 Column Population

In the column population task, which was utilised for attribute discovery, tabular data augmentation, and table retrieval (Das Sarma et al., 2012), the remaining column headers were predicted by a model given the first N columns of a "seed" table. Zhang and Balog, 2017 have compiled a dataset for this task, consisting of 1.6 million tables from Wikipedia, with a test set of 1,000 tables. The task was formulated as a multi-

label classification problem with a total of 127,656 possible header labels. It is important to note that all the tables in the column population test set were removed from our pretraining data to prevent any potential inflation of our results, in case TABBIE memorised the missing columns during pretraining.

To perform fine-tuning on the column population task, the column embeddings of the seed table, denoted as [CLSCOL], were concatenated into a single vector. This concatenated vector was then passed through a linear layer followed by a softmax layer. The model was trained using a multi-label classification objective, as described by Mahajan et al. (2018). Our baselines for comparison include the generative probabilistic model (GPM) proposed by Li et al. (2017)hang and Balog, 2017, as well as a word embedding-based extension called Table2VecH (TH) developed by Deng et al. (2019)Deng et al., 2019.

Owing to the high computational cost of fine-tuning on the entire dataset for both ATIEM and TaBERT, we selected a random subset of 100,000 training examples for fine-tuning. Additionally, unlike GPM and GPM+TH, we did not utilise the table captions during the training. Despite these limitations, Table 21 demonstrates that TABBIE and TaBERT significantly outperform both baselines. Notably, TABBIE consistently outperformed TaBERT, regardless of the number of seed columns provided. This finding suggested that TABBIE captured a richer semantic understanding of the headers and columns compared to TaBERT.

Table 21 MAP and MRR results for columns

Method	MAP	MRR
GPM	24.4	32.3
TaBERT	35.1	39.4
ATIEM(FREQ)	**37.4**	**58.4**
ATIEM(MIX)	37.1	54.5

6.7.3 Row Population

The row population task posed a greater challenge compared to the column population task. In this task, the model was provided with the first N rows of a table, where the first column contained entities (e.g., "Country"), and it was required to predict the remaining entries in the first column. Making accurate predictions for filling the column necessitated a comprehensive understanding of the contextual

information provided by the seed table. We evaluated our models using the dataset provided by Li et al. (2017) Zhang and Balog, 2017, which included a specific split for the row population task. However, owing to the size of the dataset and the resource requirements of our large embedding models, we randomly sampled a subset of tables for fine-tuning purposes.

For the row population task, our label space consisted of 300,000 entities that occurred at least twice in Wikipedia tables. Similar to the column population task, we formulated this task as a multi-label classification problem. However, this time we focussed on the representation of the first column's [CLSCOL] to predict the labels accurately.

In the row population task, both TaBERT and ATIEM demonstrated superior performance compared to the baseline EntiTables model, which utilised the table captions as external information. TaBERT performed slightly better than ATIEM when provided with only one seed row. However, as the number of seed rows increased, ATIEM showed consistent improvements over TaBERT, highlighting its ability to effectively capture and utilise contextual information for accurate predictions in this task.

Table 22 MAP and MRR results for rows

Method	MAP	MRR
Entitables	28.8	43.2
TaBERT	36.1	52.4
ATIEM(FREQ)	**42.3**	**58.4**
ATIEM(MIX)	42.2	58.6

6.7.4 Overall performance

First, we evaluated the performance of RoBERTa, a pre-trained language model, as a baseline for comparison. Second, we demonstrate the superiority of our proposed model through multiple comparison experiments and operational performance evaluations. Next, we analysed the enhanced performance of each module in our proposed model using ablation experiments. Finally, we investigated the effects of learning rate and sentence vector strategy on the experimental results by comparing multiple groups. Detailed analysis and results of these experiments are presented in the following subsections.

To evaluate the performance of the RoBERTa model compared to the TaBERT model, we conducted a comparison on the EIAO datasets. The pre-trained models were utilised to extract semantic information from the text, which was then fed into a fully connected layer for prediction. The experimental results, as shown in Table 4, demonstrated that the RoBERTa model outperformed the TaBERT model. These results confirmed the superior performance of the RoBERTa model in extracting semantic information for entity linking tasks.

Table 23 F1 score results.

Model	**F1/%**
TaBERT	88.7
RoBERTa	**88.6**

To validate the effectiveness of our proposed model, we conducted a comparison with several DL methods on the EIAO dataset using the F1 score as the evaluation metric. The following is a description of the comparative models used.

The comparison experiment results are presented in Table 24, and the following conclusions can be drawn from the analysis of the results. First, the effectiveness of our flow and label-embedding modules was demonstrated in the BERT-based model, Second, our proposed model outperformed the RoBERTa model combined with the TextRank technique, as the F1 score of RoBERTa-TextRank was 1.3% lower than that of our model. Lastly, the proposed approach outperformed commonly used DL models in entity linking. Specifically, ATIEM achieved F1 scores that were 0.9% and 0.8% higher than those of RoBERTa-Attention and RoBERTa-BiLSTM, respectively. These results confirmed the superior performance of our proposed model compared to the baselines and DL models commonly used in entity linking.

Table 24 F1 score results.

Model	**F1/%**
BERT-TextRank	87.5
RoBERTa-BiLSTM	88.6
RoBERTa-Attention	89.5
RoBERTa-TextRank	89.1
ATIEM	90.4

6.8 Applied in EIAO

Next, we selected table data from the same case and the ATIEM was made to read and achieve two objectives. The first objective was to perform a simple classification, allowing the model to have a basic understanding of the headers of all the charts in the case. This step aimed to confirm that the model could correctly comprehend the correspondence between the headers and values in this particular case. The second objective is to utilise py2neo to import the triples identified from the tables into the original ontology.

6.8.1 Model pre-train

To handle the high complexity of the information contained in the table and improve subsequent processes while minimising training requirements and early-stage errors, a manual basic classification method was implemented as the initial step. Given the abundance of numerical values and the presence of numerous attributes in the header (over 100 in this particular case), a 'basic eight-classification' was performed that included person, number, general information, space, concentration, time, compound, monitoring level, and place. The classification aimed to categorise the attributes based on their intended meaning, resulting in the identification of eight distinct categories. For instance, a sample input pair could consist of 'Andy is a people', where *people* is the corresponding category. This process was repeated for various input samples, aligning them with the eight predefined categories. Figure 6-2 shows the prediction level of this task.

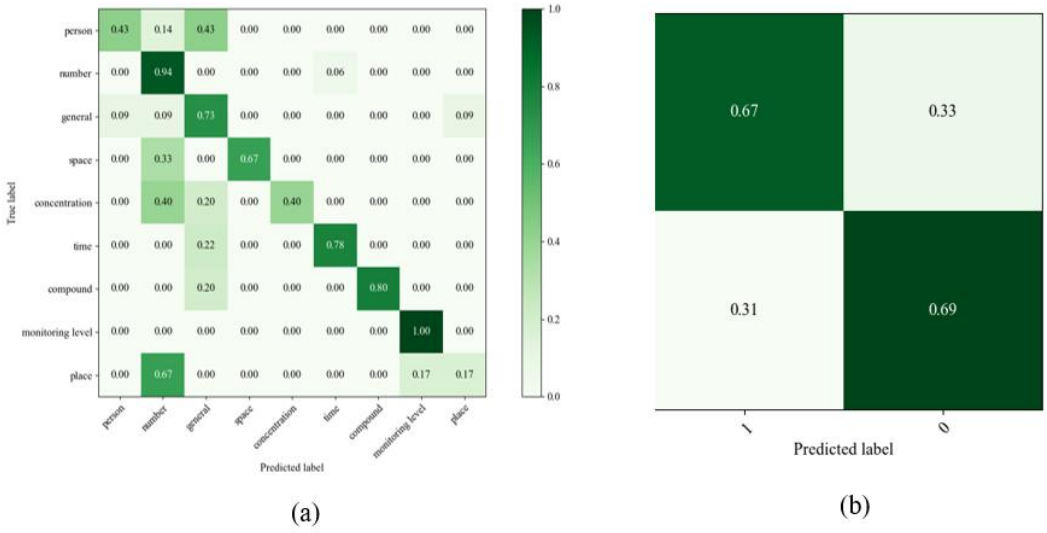


Figure 6-2 Prediction accuracy

To assess the accuracy of the model's predictions for all table headers, the diagonal values were observed, as they indicated the correct predictions. Higher values closer to 1 indicated better accuracy. Figure 6-2(b) presents the accuracy scores, which were 0.67 and 0.69. According to the criteria specified in the article, a score of 0.67 is considered good, indicating that the model exhibited a high level of recognition for the table data in this particular case. Based on this satisfactory result, the model could confidently proceed to the next step of the task.

6.8.2 Triples automatic inputting

The efficacy of the ATIEM model was demonstrated through EIAO based results of the first task. After verifying the model's capabilities and obtaining some headers and numerical corresponding triples, this step used tools to import them into the EIAO in neo4j, preferably connected. The extracted triples were then encoded into a Neo4j graph database for visualisation and information retrieval, following the approach described by Gong et al., 2018.

The py2neo plug-in was used to automatically encode triples into neo4j. It could read the pair data and connect them by certain relationship (in this case, the link between the header and value were simplified). Moreover, the node, which already

existed in EIAO could also be scanned and determined to add a new value into it. The run code of this test is presented below.

```
from py2neo import Graph, Relationship, Node
import xlrd

g = Graph("xxx", username="neo4j", password="neo4j123")
readbook = xlrd.open_workbook(r'xxx')

sheet1 = readbook.sheets()[0]
sheet_rows = sheet1.nrows
for i in range(sheet_rows):
    if i == 0:
        continue    #
    start_node = Node("Person", name=sheet1.row_values(i)[0])
    end_node = Node("Person", name=sheet1.row_values(i)[1])
    relation = Relationship(start_node, sheet1.row_values(i)[2], end_node)
    g.merge(start_node, "Person", "name")
    g.merge(end_node, "Person", "name")
    g.merge(relation, "Person", "name")
```

Table 25 and Figure 6-3 show one of the process to input triples. The initial data was extract from air pollution material. Three air monitoring places existed that could accumulate NO_2. The previous EIAO only recorded 'Air monitoring' at places 'OLP-1', 'RC-1' and 'WG-1'. After the extraction by the model, the values of NO_2 in the three place were extracted from the table data. The code py2neo then automatically read both the new triples and old ontology, finding that those three nodes had existed. Instead of creating another three nodes called 'OLP-1', 'RC-1', and 'WG-1', it linked three nodes, which presented the values '305.2', '304.9', and '301.9', respectively, directly. By this approach, the storage space and processing time could be saved.

Table 25 Reasoning results

Place	Cumulative impact of NO_2
OLP-1	305.2
RC-1	304.9
WG-1	301.9

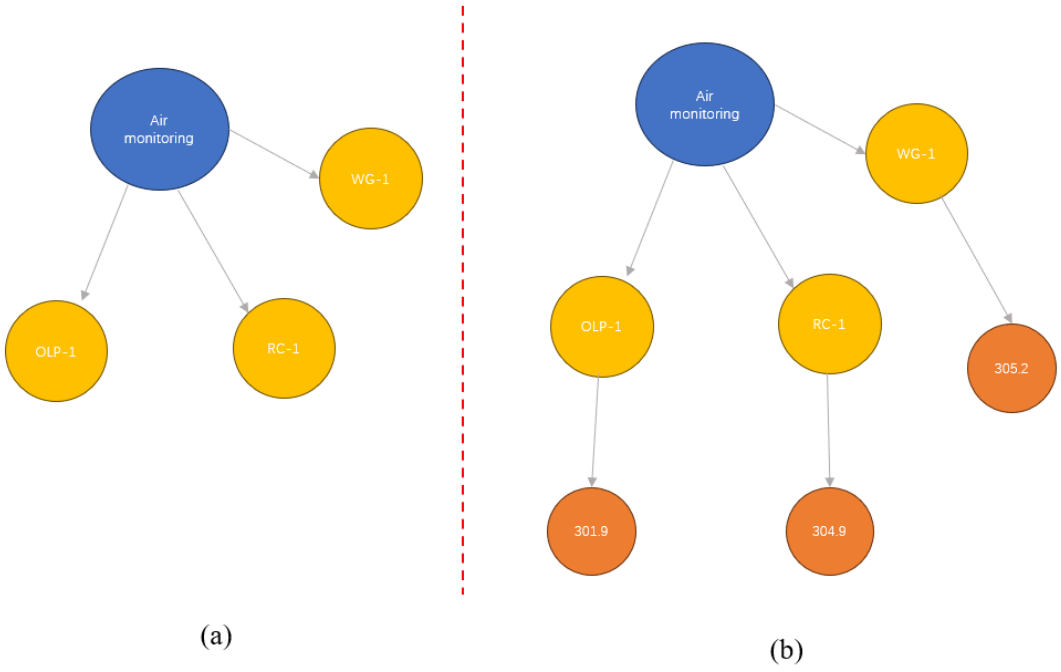


Figure 6-3 Reasoning in ontology

6.9 Finding

The ATIEM process, including the manual insertion of work package entities and link setup, was completed within 45 s. In contrast, it required the researcher's colleague 15 min to manually construct the graph without the assistance of the hybrid model. These results highlighted the effectiveness of the model in facilitating the EIAO and RAM by reducing the manual effort and accelerating the construction of the RAM graph with enhanced information.

The previous section highlighted the superior performance of ATIEM as a table representation method, surpassing TaBERT in numerous downstream task configurations and maintaining competitiveness in others. In this section, we delve deeper into the analysis of TABBIE's representations by conducting a comprehensive comparison with TaBERT. This comparison encompasses various quantitative and qualitative analysis tasks, including our custom pretraining task of corrupt cell classification, as well as tasks, such as embedding clustering and nearest neighbours. Through this extensive analysis, we aim to shed light on TABBIE's ability to capture intricate table semantics more effectively than TaBERT.

To assess TaBERT's performance on ATIEM 's pretraining task of corrupt cell detection, we conducted an experiment to evaluate its ability to detect errors in table structure decomposition. This task was particularly useful as a postprocessing step, as mistakes in predicting row/column/cell boundaries, often compounded by OCR errors, could lead to inaccurate data extraction. Table 25 presents the results of this evaluation. It is evident that all other models performed worse than ATIEM across all types of corrupt cells, including random corruption and intra-table swaps. Notably, both models struggled the most with intra-column swaps, with ATIEM achieving an F1 score of 90.4% on this subset.

Interestingly, while the MIX corruption strategy underperformed the FREQ strategy in the TABBIE models, evaluated in the previous section for downstream tasks, it demonstrated better performance in detecting more challenging corruptions. It performed almost as well in detecting random cells sampled by the FREQ strategy. This finding suggested that more complex table-based tasks might be necessary to fully exploit the representations derived using the MIX corruption strategy.

Experimental results demonstrated that the number of triples and relation types in the triples could be significantly increased using this enrichment technique, from 341 to 396, and the relation types from 39 to 40. Leveraging the enriched data, the model was effectively trained to identify missing triples in ontology, thereby improving the completeness and semantic richness of the triples used in RAM.

Overall, ATIEM marginally outperforms other automatic tabular data extraction approaches in classifying headlines on the road asset dataset. It is evident that this method is better suited to the current range and types of data. Through analysis of data trends and comparisons, ATIEM demonstrates superior performance when handling chart data characterized by simple structural relationships but large volumes. Its ability to iteratively learn methods for extracting information autonomously means that it necessitates less pre-training in the initial stages, rendering it more suitable for engineering projects with shorter time constraints. Naturally, due to its requirement for self-learning, it is better suited for documents featuring standard charts and less complex content..

6.10 Summary

This chapter presented the experimental results of the novel ATIEM model, which addressed the issue of incomplete EIA graphs by automatically identifying missing triples and completing the KBs. The model was developed based on RoBERTa, and consisted of two parts: a pre-trained information extraction model and a triple inputting module. The proposed model introduced two key improvements compared to the existing studies. First, a table data enriching module was developed, which leveraged ontological reasoning rules to enhance the semantics of triple data and facilitate training. This allowed for more accurate and comprehensive completion of information in RAM, especially for table data. Second, the model used an API (Py2Neo) for Neo4j to easily input triples, which formed from last step into the ontology. Moreover, the model could link the existing node with the formed node if they had the same character. These enhancements significantly improved the model's performance. The model demonstrated its effectiveness in identifying various types of missing triples in ontology. The completed ontology can serve as a valuable resource for engineers, enabling them to identify constraints and tasks/procedures that require attention and make informed decisions regarding constraint removal.

7 Discussion

7.1 Introduction

This chapter discusses the results of the research to identify further implications for academia and decision-makers. Sections 7.1, 7.2, and 7.3 focus on the results for the selection of ontology establishment environment, proposed ontological RAM and decision making method, and the automatic information extraction and ontology integration for tabular data, respectively. Section 7.4 states the overall contribution of this thesis to RAM and the transport sector, while Section 7.5 summarises the implication of this research.

7.2 Systematic comparison of ontology establishment techniques

This study contributes to the literature in many ways. First, from a theoretical perspective, the research explained the origination and core of ontology, description framework (RDF) and LPGs. Ontologies are semantic data models that define the types of things that exist in the domain and the properties that can be used to describe them (Zhao & Ichise, 2014). It is used to standardise concepts, using unambiguous and sound logic languages. Thus, the focus of RDF is to design an optimal way (data schema and description logic) to reach the trade-off between expressiveness, computational efficiency, and reasoning soundness (Zhao & Ichise, 2014). The LPGs are closer to the graph theory, which aim to describe relationships among entities (Alocci et al., 2015). The focus is to explicitly record that relationship to enable effective and efficient information searching and reasoning, instead of standardising the concepts (Anikin et al., 2019). RDF and LPGs are two data models, both of which have graph-based information structure. The RDF graphs are triples with standardised rules to present, while the LPGs have the richness of detail of the given data (Alocci et al., 2015). One key difference between them is that the LPGs include the possibility to add information in edges, and add properties inside the nodes without any additional structure. The explanation of key differences from a theoretical source makes the comparison clear and convictive.

Second, a four-benchmark-based comparison was conducted, which can improve the current understanding of the RDF and LPG methods as the two dominant data models for ontology. Some of the previous studies only focused on general comparison

for organised ontologies, which missed out analysis from the core building mechanism for ontology (Quinn & McArthur, 2021). Other improved studies only considered one or two benchmarks (e.g., storage size and query efficiency) in their experiments and analyses, and thus did not consider all the necessary factors when choosing a model for ontology development (Gong et al., 2018). As we mentioned in Section 2, reasoning and data visualisation from ontologies are also valuable features that provide easier information flow for both machines and audiences (Dudáš et al., 2018). Several previous works discussed the implementation of functions, and others compared these two indicators and provided a direct conclusion on model selection (Arndt et al., 2017; Dudáš et al., 2018). Moreover, a systematic comparison improves the current knowledge on the difference between data models and extends the visualisation capacity for ontology data model comparison methods, thus providing a more comprehensive approach by listing the most valuable benchmarks.

Third, the study was conducted for a road maintenance project, which helped understand and apply ontology in AEC industries. Previous works (e.g., Angles (2012); Das et al. (2014)) primarily focused on the performance from a computational perspective. However, the final aim of an ontology should be improving data management and information processing for daily use. Therefore, engineering scenarios that represent various potential situations in real-world projects were also attached and the model that performed better in each context was identified. In this study, typical engineering scenarios were simulated, and both graphs were implemented for the analysis. From the experiments, the RDF-based approaches were good at complex engineering contexts, such as automatically obtaining valuable features of subsequent tasks and calculations. On the other hand, the LPG-based approaches could provide rich functionality by visualising the information, which helped the involvers quickly gain knowledge about the project without additional explanations (Few & Edge, 2017). In addition, some subjective questions were raised in the survey, and they could provide initial information on how engineers treat ontologies as novel tools to manage information.

7.3 Development of an EIAO in RAM

The professional knowledge pool for the EIA process was hard to integrate, as the information was scattered on various websites and documents. Previous research on

EIA was limited either to improving the project management plans or reducing the engineering steps. Ontology has been implemented into EIA from different aspects, such as construction safety, design, and knowledge management (Garrido & Requena, 2011; Zhang et al., 2018; Zhou & Tao, 2011). However, these attempts were not adequately comprehensive and have not been considered to contribute to a smart decision-making process. The EIAO could help the current EIA engineering system in three stages: 1) forming a novel ontology with EIA knowledge integration; 2) updating EIA data in real-time, which can direct project managers to take immediate and proper actions; 3) improving the decision-making by providing querying, reasoning, and visualising data. The whole ontology-basic management system is illustrated in Figure 10.

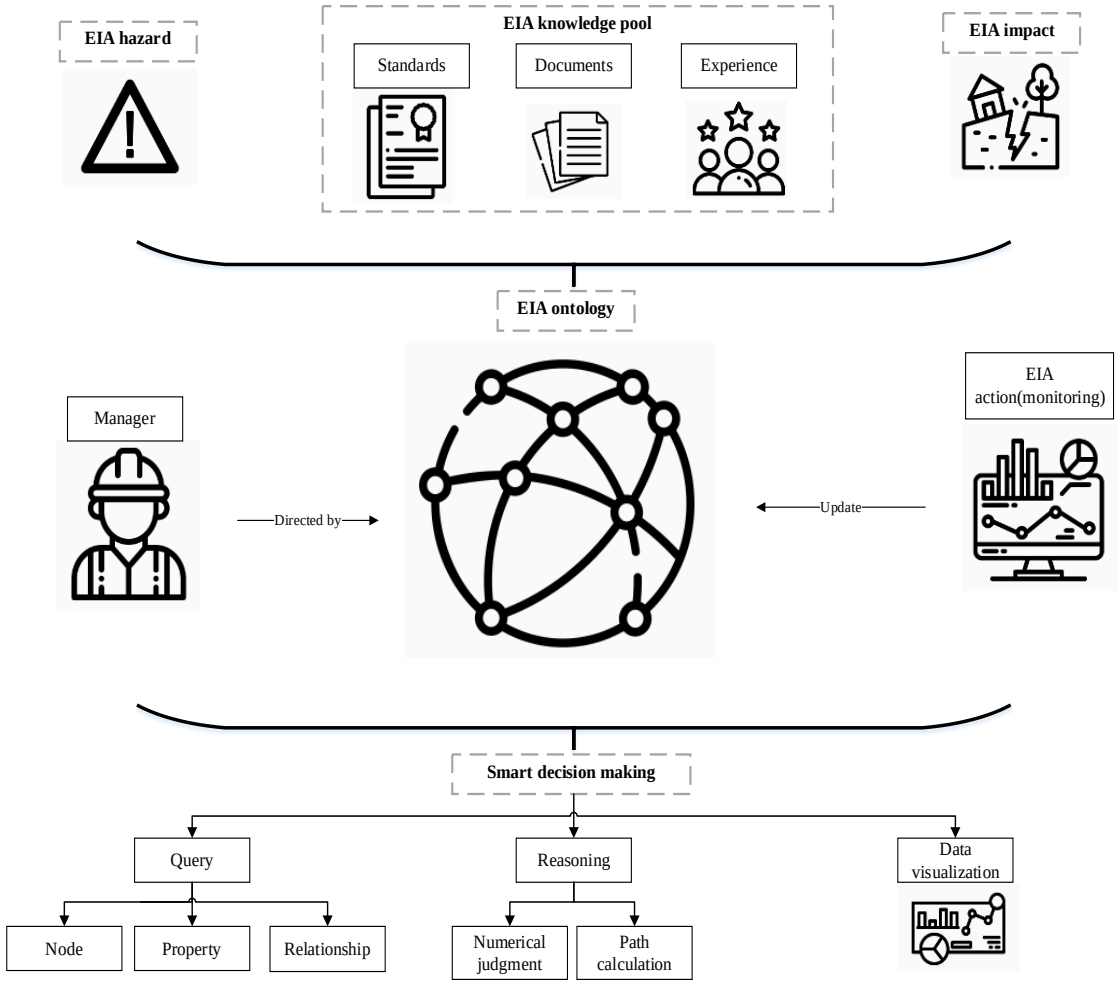


Figure 7-1 Role of EIAO in smart decision-making in a project

The proposed LPG-based ontology makes three significant contributions: 1) it extended the current ontology research in EIA, especially in relation to the projects in

the RAM field; 2) it proposed a novel ontology presentation method by the LPGs, which would enable easier sorting of the EIA knowledge; and 3) it provided efficient searching and reasoning function, using simple codes to get targeted information in a reasonable amount of time.

Firstly, environmental issues are a critical problem in any society, especially for engineering projects (Morgan, 2012). In RAM, project planning and implementation require increasingly detailed EIA knowledge and approaches to avoid these issues (Fernandes, 2000). In a traditional method, many activities, such as maintenance rely significantly on manual work to collect knowledge from paperwork, human ideas, and experience to make decisions. Thus, an EIA Ontology can improve knowledge collection by providing a structured knowledge for both humans and machines.

However, the existing environmental impact assessment ontologies only focused on general information integration or searching function evaluation (Garrido & Requena, 2011). EIA projects need to not only integrate specific information, such as specific hazards, impacts and audit actions, but also improve the decision-making process (Konys, 2018). Therefore, the previous ontologies could not be applied directly. This EIAO was established by comprehensively collecting environmental knowledge from various sources, such as standards, manuals, project documents and previous research. Thus, it can be used in most of the EIA projects without any additional efforts from the managers. Additionally, previous studies have not considered specific application situations in engineering fields, which are diverse and complex. The work of an EIA in RAM is considerable; however, it has no computer-based approaches to manage them (France-Mensah & O'Brien, 2019). This EIAO covers sufficient domain knowledge of environmental conditions in road projects, integrating knowledge, tasks and procedures, and project information of participants. Furthermore, it allows a computer to replace massive amount of manual work, thus saving storage space, cost, and improving the accuracy.

Secondly, this study chose LPGs as the data model and established a novel ontology in Neo4j instead of RDF, which enriched the cases demonstrating the use, advantages, and disadvantages of different ontology development methods. Majority of ontologies in this filed were built by RDF/OWL (Das et al., 2014). For instance, the ontologies in environmental energy management, sustainability assessment, and highway environment domain were all implemented in RDF-based tools to store and

present knowledge (Arce & Gullón, 2000; Konys, 2018; Park et al., 2016). However, previous studies often did provide adequate justifications related to the selection of the ontology establishing method (Le & Jeong, 2016). This is probably due to a lack of understanding of the specific advantages and disadvantages of these methods. With the advances in recent ontology development, traditional ontology tools have demonstrated some limitations, such as slow query time when facing large amount of data and poor visualisation functions (Vicknair et al., 2010). This study has taken the first step in examining the use of the most adopted ontology development methods, i.e., LPGs and RDF. The results show that the LPG-based ontology demonstrated more efficient querying speed with an improvement of 30%, on an average. The LPGs also demonstrated a better data visualisation system which could be seen from this case and relevant studies.

Thirdly, the current EIA process was affected by manual information searching and transferring, which would delay the information flow and cause errors (Morgan, 2012). It is difficult to find information by manual search, even in digital documents (Tang et al., 2016). The EIAO improved the time cost to find featured information significantly compared with the manual work and RDF-based methods in two levels. It embedded relevant property information into nodes, which can automatically show key information related to a target without further querying. Additionally, previous EIA related ontologies did not consider automatic information creation from existing knowledge pool (Das et al., 2014). The LPGs in Neo4j provide basic reasoning functions, such as finding errors after comparison. For instance, the EIAO can achieve basic numerical calculations under rules, and managers can get processed results to make quick decisions. Therefore, the EIAO is an early attempt in this filed to improve the current management method by automating the information search steps. It can save a significant amount of time for the project team to conclude. Although these functions can be implemented in traditional tools (such as Microsoft Project), the EIAO can manage information in the knowledge database to easily explore implicit links (such as finding the key route of a workflow).

7.4 Automatic information extraction model

The ATIEM model introduced in this study contributes significantly to the enhancement of the EIAO by addressing the existing knowledge graph limitations.

These limitations primarily involve incompleteness and a lack of rich semantics in the existing knowledge triples. Incompleteness is a critical concern, given that the current information extraction methods in the industry cannot comprehensively extract all the necessary data (Chi et al., 2019; Xu & Cai, 2020). This incompleteness can negatively impact the functionality of RAM, particularly in tasks, such as information retrieval and graph analysis. Additionally, the manual validation process and reasoning rules fall short of completing the triples owing to the extensive number of entities and intricate relations in modern projects, thereby rendering many triples unrecoverable through rule-based reasoning alone. The second limitation revolves around the sparse semantics within the current triples. Often, these triples only encompass relations with simplistic semantics, failing to specify specific relation types, synonyms, hypernyms, and essential constraint management relations, such as 'constrains' and 'has-attribute' (Qu & Tang, 2019; Yang et al., 2017). Thus, the missing information in the triples can affect the management functions in RAM (e.g., information searching and progress analysis).

Second, the main computational novelty of the ATIEM model is that it improves the current automate model such as TaBERT by utilising domain information to increase the model performance. Effective retrieval of valuable information from incomplete knowledge tables can pose a formidable challenge. Consequently, the principal objective of the ATIEM model is not to excel in information retrieval, but rather to assist engineers in identifying crucial missing information essential for the implementation of automated information extraction and input systems. This missing information may encompass constraint and task statuses. As demonstrated in Section 6.3.3, The ATIEM substantially reduces the time required to complete a KB compared to manual validation, with a significant reduction in the processing time, while maintaining a higher level of accuracy. In practical applications, this model can complement existing information retrieval methods, such as SPARQL, thereby enhancing the comprehensiveness and accuracy of the search results. Notably, the ATIEM excels in detecting intra-column swaps, achieving an F1 score of 90.4% for this subset. The experimental results demonstrate a substantial increase in the number of triples and relation types by leveraging the enrichment technique. The number of triples increased from 341 to 396, and the relation types expanded from 39 to 40. By harnessing this enriched data, the model effectively learns to identify missing triples

in the ontology, thereby ameliorating the completeness and semantic richness of the triples used in RAM.

Thirdly, the suitability of this method becomes apparent within the current scope and variety of data. Through analysis of data trends and comparisons, ATIEM exhibits superior performance when handling chart data characterized by simple structural relationships but vast volumes. Its capacity for iterative self-learning in information extraction obviates the need for extensive pre-training during initial stages, rendering it particularly apt for engineering projects with constrained timelines. However, given its reliance on self-learning, it is best suited for documents featuring standard charts and less intricate content. The primary computational innovation of theATIEM lies in its enhancement of the original models which based on BERT and RoBERTa through the incorporation of domain information to enhance model performance (Liu et al., 2019). Specifically, this entails considering information regarding domain classes and working contexts. Domain classes are identified as additional nodes and integrated into the CNN and GNN encoder. Working contexts are designated as tasks/procedures associated with entities. A constraint entity may be linked to multiple tasks/procedures, and the information from various task/procedure combinations is amalgamated to form the working context embedding of a constraint entity, which is then incorporated into the input matrix of the decoder. Both strategies serve to cluster entities from two distinct perspectives: domain classes and project stages. This approach enables the model to discern patterns among triples involving entities, relations between entities and clusters, and interactions within clusters. Consequently, the model is less susceptible to the influence of entities with disparate names.

7.5 Towards construction 4.0

The construction industry, despite its historical reluctance to adopt new information and communication technologies, is currently undergoing a transformative process reminiscent of the Industrial Revolution. This progress can be categorised into several phases:

Construction 1.0: This phase marks the transition from labour-intensive construction methods to the introduction of machinery, such as cranes, to facilitate construction processes.

Construction 2.0: During this phase, the industry shifted from non-standard construction practices to more standardised methods, such as off-site construction techniques.

Construction 3.0: Here, the industry moved from document-based construction processes to computer-based approaches, exemplified by computer-aided design.

In recent years, the construction sector has embraced various intelligent technologies associated with Industry 4.0. These technologies include building information modelling (BIM), artificial intelligence (AI), and Internet of Things (IoT) (Thakare et al., 2022). Their integration into construction projects has yielded numerous benefits, including enhanced productivity, safety, and quality. Consequently, experts in the field concur that the construction industry is progressively evolving towards **Construction 4.0** or intelligent construction (Schönbeck et al., 2020).

This advancement falls within the purview of computer and BIM-based construction management as part of the digitalisation pillar. It contributes to greater automation within the construction sector. One of the central challenges of Construction 4.0 is the establishment and upkeep of the connection between physical and digital projects. In this context, a common implementation approach involves the creation of n-D BIM models to represent various facets of a project, such as schedules (4D) and costs (5D), even before the physical construction begins. The linkage, represented by the BIM environment, is maintained through the utilisation of IoT and computer vision systems. These systems gather real-time data related to structures, labour, materials, and equipment, which is then integrated into the BIM platform (Dave et al., 2018). Nevertheless, these systems are centred around structured data such as sensor readings and geometries of defects measured in images. They lack the ability to automatically capture crucial information essential for contemporary project management approaches, such as ontologies. Specifically, they struggle to comprehend the intricate semantics and interconnections among project entities (Wu et al., 2021). As such, RAM is heavily dependent on inefficient manual approaches, such as the manual extraction and updating of interconnections among entities. In contrast, the proposed approach excels in handling unstructured data, enabling the automation of ontology modelling and triples checking. This automation liberates engineers from strenuous and repetitive manual tasks, allowing them to dedicate more time to essential management duties, such as monitoring and recording. Furthermore, this can serve as

a valuable supplement to BIM-based management. This can also supplement BIM-based management. For instance, a data link can be set up by the proposed approach and BIM systems, so that data from both sides can be automatically integrated to enable more sophisticated functions. For instance, certain studies focusing on the restoration of historical buildings suggest the following three steps: 1) retaining non-geometric information, such as historical events and intricate material properties, in ontologies; 2) transferring geometric information from BIM to ontologies for reasoning purposes (e.g., identifying inconsistencies between various inspection activities); and 3) presenting the outcomes visually in BIM for effective communication (Niknam & Karshenas, 2017; Simeone et al., 2019; Werbrouck et al., 2020). The approach can be adopted in RAM projects (e.g., storing geometries and defects of components in BIM while storing condition evaluation rules in ontologies to assist structure health assessment) to leverage the strengths of different information management tools.

7.6 Implication

The proposed information management approach addresses practical challenges in implementing ontology in RAM projects. It includes the ontology technique comparison model, ontology implementation, and the ATIEM model. The approach offers three practical implications: efficient constraint modelling, improved KBC, and enhanced information exchange among project participants.

Lined-data management approaches, such as ontology, rely on three pivotal stages: triples modelling, triples monitoring/analysis, and triples removal. However, the initial phase, triples modeling, is predominantly executed through manual scrutiny of project documents. Furthermore, the resultant project data from triples modeling are frequently incomplete. Given that practical ontology management is an iterative and resource-intensive process, the inefficiencies in modeling and the incompleteness of triples can impede subsequent constraint management steps. Moreover, linked data necessitates efficient information exchange. Nevertheless, in road asset projects, multiple project participants manage disparate databases or file systems, lacking a unified schema for information integration and sharing. To address these challenges, this research proposes the ontology comparison model, a novel ontology and ATIEM model to facilitate the implementation of ontology in road projects. Consequently, the

proposed information management approach offers the following three practical implications.

7.6.1 Ontology technique comparison model

Instead of macro-level understanding the differences, this study achieved mathematical insights from five benchmarks for two outstanding ontology techniques. In the data density benchmark, the LPGs could save more than two-thirds of the storage space compared with the RDF, and the difference became even larger when the size of the dataset increased. Although previous works have reached similar conclusions (e.g., De Abreu et al. (2013)), this study provided a uniform factor for a straightforward and professional measurement. For querying benchmarks, most previous studies compared basic queries of models and found small differences. In this study, the logic difficulty was set from 'simple to complex' to provide for a deeper analysis. The LPGs performed better when searching for direct information (e.g., all vertices). However, the embedded properties were hard to find, as additional logical analyses were required in the LPGs, which made it less efficient in querying featured data relative to the RDF. Next, the RDF method showed a dominant advantage in reasoning benchmarks. The simple structure may have behaved less efficiently in other contexts, and it represented a good feature, as the reasoning logic could be easily set up. A series of qualitative analyses proved that the LPG-based approaches had more functions and better performance in visualisation data, while the RDF-based approaches needed to combine multiple tools to achieve the same functionality. A comparison of both reasoning and visualisation can fill in the gap because the supporting information was difficult to obtain by singular analyses. Moreover, the distribution of the size of the datasets was set reasonably, which could contribute to the logicality and integrity of the experiment. Thus, the experimental results can be trusted and they provide a valuable basis for selecting between the two models.

7.6.2 Ontology-based project information integration platform

The EIAO efficiently integrates, infers, and searches static and dynamic information in road rehabilitation projects. It offers a software-neutral platform for accessing project information, facilitating collaboration among participants. The EIAO supports important management functions such as evaluating work progress,

monitoring constraint removal, assessing participant performance, and identifying critical constraints. The previous ontologies could not be applied directly. The EIAO has been meticulously crafted by aggregating environmental knowledge from diverse sources, including standards, manuals, project documents, and prior research. Consequently, it proves to be applicable in the majority of EIAprojects without necessitating additional effort from managers. Notably, previous studies often overlooked the nuanced application scenarios within engineering fields, characterized by their diversity and complexity. Given the substantial volume of EIA work in RAM, the absence of a computer-based approach for efficient management is conspicuous. The EIAO not only encapsulates extensive domain knowledge related to environmental conditions in road projects but also integrates knowledge, tasks, procedures, and project information of participants. Moreover, it empowers a computer to supplant labor-intensive manual processes, leading to savings in storage space and costs, alongside enhanced accuracy.

The OWL API functions by exporting information from the EIAO and then programmatically executes all necessary computations (e.g., determining task/procedure delays and the ratio of unresolved constraints). Subsequently, the computed results are reintegrated back into the EIAO via the API. Conversely, SWRL and SQWRL excel in deducing new knowledge within ontologies. Leveraging the updated information, rules are applied to infer additional insights (i.e., triples) that reflect the project's performance across three key dimensions: work progress, constraint removal progress, and participant performance. This approach can be viewed as an extension of existing information management methodologies in ontologies, allowing for the continuous updating of the EIAO to incorporate both static and dynamic project information.

Compared to manual approaches, the EIAO significantly reduces the time required for integrating, inferring, and searching project information, especially when it is scattered across multiple sources. By leveraging the EIAO, stakeholders can access valuable project insights and navigate its KBs to uncover implicit information that may not be readily apparent in traditional project management tools.

7.6.3 Automatic table information extraction model

This research has two key contributions. First, it improved automatic ontology extraction and establishment that relies on high-quality knowledge graphs. Current automatic approaches have two limitations: 1) They suffer from incompleteness because information extraction methods in the industry cannot extract all the needed information. 2) They lack rich semantics as they often only consider relations with simple semantics, e.g., existence of relations (i.e., no relation type is identified), synonyms, hypernyms (Chi et al., 2019; Xu & Cai, 2020), as well as basic management relations (‘constrains’ and ‘has-attribute’) (Wu et al., 2021; ZhongXing et al., 2020). These two limitations can hinder ontology extraction as follows: Given the large number of entities and complex relations in modern projects, it is difficult to complete KBs using manual checking or reasoning rules (many triples cannot be reasoned by rules) (Qu & Tang, 2019; Yang et al., 2017). Thus, the missing information in KBs can affect management functions in AWP (e.g., information searching and graph analysis). On the other hand, current KBC models cannot be directly applied to ATIEM, because the lack of semantics in KBs can largely hurt model performance. Incomplete KBs can hinder the discovery of valuable information for supporting RAM functions, through either manual or automated approaches. Especially for table data in management material, few studies have been conducted to find an automated approach to create knowledge and store them. To address this challenge, this research proposed an ATIEM model that aids engineers in identifying critical missing information in ontological triples, such as constraint and task status updates. The model demonstrated its effectiveness in reducing the time required for checking and completing a triple compared to manual approaches, achieving time reductions ranging from 1/6 to 1/40, while maintaining a high completion accuracy.

The primary function of the ATIEM model, when implemented in practice, is to enhance the quality of ontology by identifying and incorporating missing information. It serves as a valuable supplementary tool for improving completeness and accuracy, particularly for constraint and task status updates. The integration of the KBC model with information searching tools, such as SPARQL for ontology querying, further enhances the comprehensiveness and accuracy of information retrieval. By leveraging the proposed model as a supplementary tool and integrating it with information

searching tools, practitioners can overcome the challenges posed by incomplete ontology and facilitate more effective management of tabular data.

8 Conclusions, contributions, implications, and future work

8.1 Introduction

This research study has focussed on improving knowledge collection and integration in RAM projects. It has proposed an innovative approach based on novel ontology models for information extraction and completion, as well as ontologies for information integration.

The proposed information management approach includes three key components: 1) a comprehensive comparison of popular ontology techniques; 2) an LPGs-based ontological KB (i.e., EIAO) to integrate information and support project management functions; and 3) an ATIEM model to identify table data and automatically form triples in ontology. The hybrid comparison model can help manager choose a suitable data model for ontology; then, the EIAO extracts information from documents and integrates such information into the ontological KB; finally, the ATIEM model is used to enrich the triples and improve the quality of ontology. Both DL model experiments and controlled experiments have been carried out to validate the capacity and usefulness of each component in the proposed approach. The results show that the approach can attain high performance in terms of entity/relation extraction and completion; it can also integrate dynamic project information, i.e., constraints, tasks, procedures, attributes of constraints, and project participants. The approach can largely automate information extraction modelling, while enabling effective information integration. Hence, it can contribute to project success by saving significant amount of time for RAM tasks.

8.1.1 Research findings for Objective 1

Objective 1: To obtain an in-depth understanding of ontology approaches and their implementations in RAM areas.

Summary of findings: A critical review was carried out based on 117 and 16 articles on ontology and table information extraction and integration approaches, respectively. All documents were taken from the Web of Science and databases of RAM implementors. The review has clearly revealed the following research gaps:

- As an emerging technique, ontology has been implemented in many RAM fields since 2006. However, most of the research has focussed on traffic service sand road assets, while other knowledge areas have not been comprehensively studied. In terms of the life-cycle stage, over half of the studies have focussed on the operation stage of RAM.
- From an engineering perspective, the adoption of standard techniques (e.g., RDF and OWL) in ontology has been increasing, and various models and languages have been developed. Among the tools, Protégé is the most frequently used, as it has many functions, such as creating, editing, and presenting ontologies.
- Finally, five main gaps were identified in the review process. Ontology development in RAM fields should consider automatic mechanisms, multiple techniques, sharing and linking ontologies coordination with other technologies, and considering user-friendliness

8.1.2 Research findings for Objective 2

Objective 2: To establish a systematic comparison model for ontology establishment techniques to choose the proper one for implementing RAM ontology.

Summary of findings: This objective performed a series of experiments to compare the two models based on four benchmarks in the current research field that have been predominantly focussed on, namely, data density, query efficiency, reasoning and data visualisation. The aim of this study was to find the best choice between the RDFs and LPGs in different implementation contexts through a comprehensive analysis.

- Although the file sizes read from the computers were quite similar, the data density of the LPG approach was always higher than that of the RDF approach in all cases; furthermore, an increasing gap occurred as the scale of the dataset increased. In other words, the LPG approach performed better than the RDF approach in saving storage space when faced with larger datasets.
- The final results indicated that LPGs required 50% less time than the RDF, on average, when querying information. As the data size of the RDF increased, the time cost increased almost linearly when finding all the vertices,

although the effect was minimal for queries associated with finding certain featured data.

- RDF-based approach had a much more powerful reasoning ability than the LPG-based approach. The LPGs-based approach could achieve similar reasoning function when faced with general conditions.
- LPGs had slight advantages over the RDF method in visualising information and hence, can be accepted more easily by audiences.

8.1.3 Research findings for Objective 3

Objective 3: To develop ontological KBs to integrate the information of environmental impact assessment in RAM.

Summary of findings: This objective aimed to investigate a framework to organise, transfer, and digitise the EIA information flow. LPGs-based on the Neo4j graph database was proposed to formalise critical EIA management knowledge. Three real-life scenarios were applied to the EIAO to validate improved information searching and implicit reasoning knowledge.

- EIAO integrated the existing standards and specifications and could even connect the requirements of multiple countries, which had stronger reference and flexibility. At the same time, it could also add the experience of experts as part of the knowledge pool, which could help the manager make more professional decisions. At the same time, this knowledge pool also save on cumbersome steps, such as printing and storing documents.
- The EIAO made it smarter for the decision-making process by quickly querying the relevant information so that the decision-maker could provide a reasonable judgment immediately. Additionally, its reasoning ability based on numbers and keywords could automatically output some recommended actions from the default action pool in nodes. Thus, it reduced the thinking time for managers, and workers could even act directly according to the action specified by the EIAO without instructions.
- In the EIAO, querying both general information and specific project condition was much faster than manual checking the result. On an average, the EIAO significantly reduced the searching time from 1827.3 to 4.3 s, while maintaining 100% accuracy compared with human effort.

8.1.4 Research findings for Objective 4

Objective 4: To realise automatic information extraction and ontology establishment for RAM from tabular data by DL model.

Summary of findings: The ATIEM model has two essential parts: a RoBERTa based pre-trained model to learn embeddings of entities/relations, and a Py2Neo-based automatic triples inputting module. Some controlled experiments have been conducted to check the ability of ATIEM in the EIAO.

- The proposed ATIEM model could effectively identify tabular data in the RAM. Compared with other ML model, the maximum performance had a better MAP and MRR. Overall, F1 was 88.9% (0.2% more than that of TaBERT)
- It was evident that all other performance measures were worse than ATIEM across all types of corrupt cells, including random corruption and intra-table swaps. Notably, both models struggled the most with intra-column swaps, with ATIEM achieving an F1 score of 90.4% on this subset.
- Experimental results demonstrated that the number of triples and relation types in the triples could be significantly increased using this enrichment technique, from 341 to 396 and relation types from 39 to 40.
- The ATIEM model could reduce the time to check and complete the ontology to 1/6–1/40 of the manual checking, while obtaining a higher F1 in terms of identifying the missing information.

8.2 Contribution

8.2.1 Summary of theoretical contributions

First, it expanded the existing domain ontologies, broadening their scope and applicability. Second, it introduced a novel approach for integrating dynamic information in ontologies, overcoming their limitations and enabling effective representation and processing of changing data. Last, the research developed innovative computational models for automatic information extraction and KBC in the AEC industry. These models leverage advanced techniques, such as DL and NLP to enhance the acquisition and organisation of information.

8.2.2 Expansion of domain ontologies

This research addressed a gap in existing ontologies for RAM, and primarily focussed on inspection, evaluation, and decision-making stages. However, RAM requires specific domain knowledge, including specialised constraints, tasks, and their relationships. The existing ontologies lack this specific knowledge, thus making it challenging to effectively integrate information related to road rehabilitation. To overcome this limitation, this research proposed the EIAO, which was specifically designed to capture and integrate the RAM knowledge. The EIAO expands the coverage of ontologies by incorporating information on rehabilitation tasks/procedures, project participants, and three types of constraints, namely engineering, supply-chain, and site constraints. Importantly, the EIAO can also be seamlessly integrated with other road ontologies, such as those modelling road components, to support informed maintenance decisions without requiring significant modifications.

8.2.3 Critical Examination and evaluation of ontology methodologies

This research explained the origination and core of ontology, RDF and LPGs. Ontologies are semantic data models that define the types of entities that exist in the domain and properties that can be used to describe them. It is used to standardise concepts, using unambiguous and sound logic languages. Thus, the focus of RDF is to design an optimal way (data schema and description logic) to reach a trade-off among expressiveness, computational efficiency, and reasoning soundness. A four-benchmark-based comparison was conducted, which could improve the current understanding of the RDF and LPG methods as two dominant data models for ontology. Some of the previous studies only focused on a general comparison for organised ontologies, and hence missed out analysis from the core building mechanism for ontology. Other improved studies only considered one or two benchmarks (e.g., storage size and query efficiency) in their experiments and analyses, and thus did not consider all the necessary factors when choosing a model for ontology development. Moreover, a systematic comparison conducted in this study improved the current knowledge on the difference between data models and extended the visualisation capacity for ontology data model comparison methods, thus providing a more comprehensive approach by listing the most valuable benchmarks.

8.2.4 A novel approach for automatic information extraction in ontologies

Previous ontologies in the architecture, engineering, and construction (AEC) sector have primarily focused on integrating static information and facts, such as geometries and reasons for defects or accidents. However, these ontologies often overlooked the dynamic project information that changes frequently, such as task progress and constraint removal. The main limitation with this approach is that conventional ontologies lack the necessary support for critical computations required to update such dynamic information.

To address this limitation, the proposed EIAO combined the use of SWRL, SQWRL, and OWL API to overcome syntax limitations and enable effective information updating. The OWL API was used to export information from the EIAO and perform the required computations programmatically, such as calculating task/procedure delays and ratio of unremoved constraints. The computed results were then imported back into the EIAO using the API.

This study chose LPGs as the data model and established a novel ontology in Neo4j instead of RDF, which enriched cases demonstrating the use, and advantages and disadvantages of different ontology development methods. A majority of ontologies in this filed were built by RDF/OWL (Das et al., 2014). This was probably due to a lack of understanding of the specific advantages and disadvantages of these methods. With the advances in recent ontology development, traditional ontology tools have demonstrated some limitations, such as slow query time when faced with large amounts of data and poor visualisation functions (Vicknair et al., 2010). This study has taken the first step to examine the use of the most commonly adopted ontology development methods, i.e. LPGs and RDF. The results showed that the LPG-based ontology had more efficient querying speed with an improvement of 30%, on average. The LPGs also have a better data visualisation system which can be seen from this case and relevant studies.

8.2.5 Novel computational models for automatic information extraction and KBC

This research proposed a DL model, namely the ATIEM, to automate constraint modelling and provide comprehensive information for tabular data in the AEC sector. These models leverage state-of-the-art NLP techniques; however, as the existing NLP

models focus on general knowledge and lack domain-specific information, they may not perform well when directly applied to RAM.

To address this limitation, this research introduced domain-specific information to modify the structures of the DL models, thereby improving their performance. These domain-specific details help cluster constraint entities and reduce the model's distraction caused by heterogeneous entity names during training and testing. The research proposed two ways to utilise domain information effectively. First, for the ATIEM model, embeddings of domain classes (header) of a triple are horizontally stacked at both sides of the input matrix. Second, for the Py2Neo inputting module, the domain triples are inserted into the EIAO as additional nodes that are processed by the encoder.

Through detailed model experiments, the proposed model demonstrated an average increase of 0.2%–3.4% in the F1 score for triple extraction, while the proposed model demonstrated an increase of 3.4%–9.7% in MAP and MRR. Moreover, this research represents an early attempt to extract both entities and semantic-rich relations in the AEC sector. Therefore, the model training and validating protocols, optimal hyperparameters, and model performance metrics presented in this research serve as valuable baselines for future IE or NLP studies in the sector.

8.3 Limitations and future work

In this section, limitations of the proposed information extraction and integration approach for RAM ontology modelling are identified. Accordingly, potential future research directions are proposed.

First, the original knowledge pool has been collected, sorted and transferred through manual effort. To make the ontology more practical and intelligent, automating the ontology development process using technologies, such as machine learning or big data applications is necessary. For example, Wu et al. (2021) applied NLP in bridge maintenance ontology to automatically create new constraints. Although EIAO has reasoning capability, in the future such a capability can be significantly improved by other supplementary computer programming languages through various application programming interfaces (API) (e.g., UiO by Java and Session by C#) (Hartig, 2019). Additionally, the decision-making process in EIAO

follows a process, by which managers retrieve key information and then make relevant decisions. In future studies, such a decision tree can be embedded into EIAO to further automate the lifecycle process. Finally, this study has not designed a unique and friendly interface for end users. The use of EIAO and relevant tools require some basic coding skills, which warrant that extra training lessons be imparted to the project managers. An individual operation interface, either web-based or software could minimise the training required (Hu et al., 2019).

Second, the proposed model has a limitation in that it does not utilise all the data present in the triples. This is due to the high variance and sparsity of such data, as well as the focus of RoBERTa models on interpreting connections among nodes rather than predicting specific attribute values. To address this limitation, future research will explore training additional ML models specifically designed to predict missing attributes by considering various factors, such as task type, quantities, and constraint removal progress. The current performance of the ATIEM model, as indicated by its F1 score of 88.9%, is not exceptionally high, and human intervention is required in tasks, such as selecting one entity from multiple candidates. To improve the model's performance, more data will be collected for training purposes, aimed at increasing the accuracy and efficiency of the model. Furthermore, as mentioned in Section 2.4, unsupervised methods similar to association rule mining and DL models that automatically create reasoning rules show promise in inferring implicit knowledge and creating table information in ontology. By combining DL models with Markov logic networks, information searching and reasoning capabilities of the EIAO, can be enhanced in addition to improving the performance of the ATIEM model. While these methods are still in their early stages and have limited application in the AEC sector, it is worthwhile to explore their potential for improving RAM projects.

Third, the evaluation method can be improved by more complex approaches. For example, the proposed evaluation for visualisation benchmarks is quality-based, and it could be improved by adding more participants and deeper experience (e.g., forming a study group to teach participants these two data models with the visualisation function). In addition, visualisation comparison of the two approaches is restricted to the default plug-ins. There are other visualisation tools developed for both RDF and LPGs, and the comparison of these could be conducted in future studies for a deeper understanding of visualisation. Finally, the comparison selected two of the most

popular and representative data models for the study. However, there are many other data models that may have advantages for certain benchmarks. Moreover, with rapid updates in storage and cloud techniques, more data models can be tested and identified in terms of optimal implementations in the future.

Reference

Abdelaziz, Harbi, Khayyat, & Kalnis. (2017). A survey and experimental comparison of distributed SPARQL engines for very large RDF data. *Proceedings of the VLDB Endowment, 10*(13), 2049-2060.

Akiho. (2002). Overview of total productive maintenance, case studies of best practice of the Japanese manufacturing industry. *ChuSanRen (Central Japan Industries Association), Nagoya*.

Akrivi, Elena, Constantin, Georgios, & Costas. (2006). *A comparative study of four ontology visualization techniques in protege: Experiment setup and preliminary results.* Paper presented at the Tenth International Conference on Information Visualisation (IV'06).

Ali, Kwak, Khan, El-Sappagh, Ali, Ullah, Kim, & Kwak. (2019). Transportation sentiment analysis using word embedding and ontology-based topic modeling. *Knowledge-Based Systems, 174*, 27-42.

Aliyu, Singhry, Adamu, & AbuBakar. (2015). *Ontology, epistemology and axiology in quantitative and qualitative research: Elucidation of the research philophical misconception.* Paper presented at the Proceedings of the Academic Conference: Mediterranean Publications & Research International on New Direction and Uncommon.

Alocci, Mariethoz, Horlacher, Bolleman, Campbell, & Lisacek. (2015). Property graph vs RDF triple store: A comparison on glycan substructure search. *PLoS One*, *10*(12), e0144578. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4684231/pdf/pone.0144578.pdf

Anderson, Arlitt, Morrey III, & Veitch. (2009). DataSeries: an efficient, flexible data format for structured serial data. *ACM SIGOPS Operating Systems Review, 43*(1), 70-75.

Angjeliu, Coronelli, & Cardani. (2020). Development of the simulation model for Digital Twin applications in historical masonry buildings: The integration between numerical and experimental reality. *Computers & Structures, 238*, 106282.

Angles. (2012). *A Comparison of Current Graph Database Models*. Paper presented at the 2012 IEEE 28th International Conference on Data Engineering Workshops. https://ieeexplore.ieee.org/document/6313676/

Angles, & Gutierrez. (2018). An introduction to graph data management. In *Graph Data Management* (pp. 1-32): Springer.

Angles, Thakkar, & Tomaszuk. (2019). *RDF and Property Graphs Interoperability: Status and Issues.* Paper presented at the AMW.

Anikin, Borisenko, & Nedumov. (2019). *Labeled Property Graphs: SQL or NoSQL?* Paper presented at the 2019 Ivannikov Memorial Workshop (IVMEM).

Arce, & Gullón. (2000). The application of strategic environmental assessment to sustainability assessment of infrastructure development. *Environmental impact assessment review, 20*(3), 393-402.

Arndt, De Meester, Dimou, Verborgh, & Mannens. (2017). *Using rule-based reasoning for RDF validation.* Paper presented at the International Joint Conference on Rules and Reasoning.

Ashraf, Chang, Hussain, & Hussain. (2015). Ontology usage analysis in the ontology lifecycle: A state-of-the-art review. *Knowledge-Based Systems, 80*(may), 34-47.

Asim, Wasim, Khan, Mahmood, & Abbasi. (2018a). A survey of ontology learning techniques and applications. *Database, 2018*.

Asim, Wasim, Khan, Mahmood, & Abbasi. (2018b). A survey of ontology learning techniques and applications. *Database, The Journal of Biological Databases and Curation, 2018*, 1-24.

Austroads. (2016). Guide to asset management. In. Australia.

Baken. Linked Data for Smart Homes: Comparing RDF and Labeled Property Graphs.

Baldwin. (1990). *Naming and Grouping Privileges to Simplify Security Management in Large Databases.* Paper presented at the IEEE Symposium on Security and Privacy.

Ban, El Karoui, & Lim. (2018). Machine learning and portfolio optimization. *Management Science, 64*(3), 1136-1154.

Barrachina, Garrido, Fogue, Martinez, Cano, Calafate, & Manzoni. (2012). VEACON: A Vehicular Accident Ontology designed to improve safety on the roads. *Journal of Network and Computer Applications, 35*(6), 1891-1900. doi:10.1016/j.jnca.2012.07.013

Baton, & Van Bruggen. (2017). *Learning Neo4j 3. x: Effective data modeling, performance tuning and data visualization techniques in Neo4j*: Packt Publishing Ltd.

Beetz, & Borrmann. (2018). *Benefits and limitations of linked data approaches for road modeling and data exchange.* Paper presented at the Workshop of the European Group for Intelligent Computing in Engineering.

Bennett, Chamorro, Chen, Solminiihac, & Flintsch. (2007). *Data collection technologies for road management*. Retrieved from

Bennett, De Solminihac, & Chamorro. (2006). Data collection technologies for road management.

Berdier. (2011). Road System Ontology: Organisation and Feedback. In *Ontologies in Urban Development Projects* (pp. 211-216): Springer.

Berges, Ramírez-Durán, & Illarramendi. (2021). A Semantic Approach for Big Data Exploration in Industry 4.0. *Big Data Research, 25*, 100222.

Bermejo, Villadangos, Astrain, & Cordoba. (2013). Ontology based road traffic management. In *Intelligent Distributed Computing VI* (pp. 103-108): Springer.

Bermejo, Villadangos, Astrain, Cordoba, Azpilicueta, Garate, & Falcone. (2014). Ontology Based Road Traffic Management in Emergency Situations. *Ad Hoc & Sensor Wireless Networks, 20*(1-2), 47-69. Retrieved from <Go to ISI>://WOS:000324972600004

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4505179/pdf/epih-37-e2015027.pdf

Bermudez, & Piasecki. (2004). *Role of ontologies in creating hydrologic metadata.* Paper presented at the International Conference on HydroScience and Engineering, Brisbane, Australia.

Berners-Lee, Hendler, & Lassila. (2001). The semantic web. *Scientific american, 284*(5), 28-37.

Bilal, Oyedele, Qadir, Munir, Ajayi, Akinade, Owolabi, Alaka, & Pasha. (2016). Big Data in the construction industry: A review of present status, opportunities, and future trends. *Advanced Engineering Informatics, 30*(3), 500-521.

Bizer. (2009). The Emerging Web of Linked Data. *Ieee Intelligent Systems, 24*(5), 87-92. doi:Doi 10.1109/Mis.2009.102

Bizer, Heath, & Berners-Lee. (2011). Linked data: The story so far. In *Semantic services, interoperability and web applications: emerging concepts* (pp. 205-227): IGI Global.

Block, & Markowitz. (2000). *The flawless consulting fieldbook and companion: A guide to understanding your expertise*: John Wiley & Sons.

Canter, & Atkinson. (2008). Environmental indicators, indices and habitat suitability models. *International Association for Impact Assessment*.

Carbon, Ireland, Mungall, Shu, Marshall, Lewis, Hub, & Group. (2009). AmiGO: online access to ontology and annotation data. *Bioinformatics, 25*(2), 288-289. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2639003/pdf/btn615.pdf

Casey, Jones, Xiao, Anders, Christie-Blick, Sharp, Masson, Lucchini, Holland, & Legrand. (2005). Interesting Papers in Other Journals. *AMERICAN JOURNAL OF SCIENCE, 305*(1).

Ceausu, & Despres. (2006). Case based reasoning to analyze road accidents. *International Journal of Computers Communications & Control, 1*, 118-123. Retrieved from <Go to ISI>://WOS:000203014800018

Chen, Lin, Meng, Liang, & Tan. (2023). Named Entity Identification in the Power Dispatch Domain Based on RoBERTa-Attention-FL Model. *Energies, 16*(12), 4654.

Cheng, & Ugrinovskii. (2016). Event-triggered leader-following tracking control for multivariable multi-agent systems. *Automatica, 70*, 204-210.

Chi, Jin, & Hsieh. (2019). Developing base domain ontology from a reference collection to aid information retrieval. *Automation in Construction, 100*, 180-189.

Chi, Wang, & Jiao. (2015). BIM-enabled structural design: impacts and future developments in structural modelling, analysis and optimisation processes. *Archives of Computational Methods in Engineering, 22*(1), 135-151.

Clark, Luong, Le, & Manning. (2020). Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Consoli, Presutti, Recupero, Nuzzolese, Peroni, & Gangemi. (2017). Producing linked data for smart cities: The case of Catania. *Big Data Research, 7*, 1-15.

Constantinov, Mocanu, & Poteras. (2015). Running complex queries on a graph database: A performance evaluation of neo4j. *Annals of the University of Craiova, 12*(1), 38-44.

Cordoba, Astrain, Villadangos, Azpilicueta, Lopez-Iturri, Aguirre, & Falcone. (2017). SesToCross: Semantic Expert System to Manage Single-Lane Road Crossing. *Ieee Transactions on Intelligent Transportation Systems, 18*(5), 1221-1233. doi:10.1109/tits.2016.2604079

Corry, Pauwels, Hu, Keane, & O'Donnell. (2015). A performance assessment ontology for the environmental and energy management of buildings. *Automation in Construction, 57*, 249-259.

Corsar, Markovic, Edwards, & Nelson. (2015). *The transport disruption ontology.* Paper presented at the International Semantic Web Conference.

Czarnecki. (2018). Operational world model ontology for automated driving systems–part 1: Road structure. *Waterloo Intelligent Systems Engineering Lab (WISE) Report, University of Waterloo*.

Dao, Ng, Yang, Zhou, Xu, & Skitmore. (2021). Semantic framework for interdependent infrastructure resilience decision support. *Automation in Construction, 130*, 103852.

Darejeh, & Singh. (2013). A review on user interface design principles to increase software usability for users with less computer literacy. *Journal of computer science*, *9*(11), 1443.

Das, Cheng, & Law. (2015). An ontology-based web service framework for construction supply chain collaboration and management. *Engineering Construction and Architectural Management, 22*(5), 551-572. doi:10.1108/Ecam-07-2014-0089

Das, Srinivasan, Perry, Chong, & Banerjee. (2014). *A Tale of Two Graphs: Property Graphs as RDF in Oracle.* Paper presented at the EDBT.

Dave, Buda, Nurminen, & Främling. (2018). A framework for integrating BIM and IoT through open standards. *Automation in Construction, 95*, 35-45.

De Abreu, Flores, Palma, Pestana, Pinero, Queipo, Sánchez, & Vidal. (2013). *Choosing Between Graph Databases and RDF Engines for Consuming and Mining Linked Data.* Paper presented at the Cold.

Decker, Melnik, Van Harmelen, Fensel, Klein, Broekstra, Erdmann, & Horrocks. (2000). The semantic web: The roles of XML and RDF. *IEEE Internet computing, 4*(5), 63-73.

Delir Haghighi, Burstein, Zaslavsky, & Arbon. (2013). Development and evaluation of ontology for intelligent decision support in medical emergency management for mass gatherings. *Decision Support Systems, 54*(2), 1192-1204. doi:10.1016/j.dss.2012.11.013

Deng, Guo, Xue, & Zafeiriou. (2019). *Arcface: Additive angular margin loss for deep face recognition.* Paper presented at the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.

Donkers, Yang, & Baken. (2020). *Linked Data for Smart Homes: Comparing RDF and Labeled Property Graphs.* Paper presented at the LDAC 2020 Linked Data in Architecture and Construction: Proceedings of the 8th Linked Data in Architecture and Construction Workshop Dublin, Ireland, June 17-19, 2020.

Drakopoulos, Kanavos, Mylonas, Sioutas, & Tsolis. (2017). *Towards a framework for tensor ontologies over Neo4j: Representations and operations.* Paper presented at the 2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA).

Du, Anand, Alechina, Morley, Hart, Leibovici, Jackson, & Ware. (2012). Geospatial Information Integration for Authoritative and Crowd Sourced Road Vector Data. *Transactions in Gis, 16*(4), 455-476. doi:10.1111/j.1467-9671.2012.01303.x

Dudáš, Lohmann, Svátek, & Pavlov. (2018). Ontology visualization methods and tools: a survey of the state of the art. *The Knowledge Engineering Review, 33*.

El-Gohary, & El-Diraby. (2010). Domain ontology for processes in infrastructure and construction. *Journal of Construction Engineering and Management, 136*(7), 730-744.

Fernandes. (2000). Landscape ecology and conservation management—Evaluation of alternatives in a highway EIA process. *Environmental impact assessment review, 20*(6), 665-680.

Fernandez, Hadfi, Ito, Marsa-Maestre, & Velasco. (2016). Ontology-Based Architecture for Intelligent Transportation Systems Using a Traffic Sensor Network. *Sensors, 16*(8), 17. doi:10.3390/s16081287

Fernandez, & Ito. (2017). Semantic Integration of Sensor Data with SSN Ontology in a Multi-Agent Architecture for Intelligent Transportation Systems. *IEICE TRANSACTIONS on Information and Systems, 100*(12), 2915-2922.

Few, & Edge. (2017). Data Visualization Effectiveness Profile. *Perceptual Edge*, 1-11.

France-Mensah, & O'Brien. (2019). A shared ontology for integrated highway planning. *Advanced Engineering Informatics, 41*. doi:10.1016/j.aei.2019.100929

Freyne, Coyle, Smyth, & Cunningham. (2010). Relative status of journal and conference publications in computer science. *Communications of the ACM, 53*(11), 124-132.

Fu, Zhang, & Li. (2016). *Using LSTM and GRU neural network methods for traffic flow prediction.* Paper presented at the 2016 31st Youth academic annual conference of Chinese association of automation (YAC).

Garrido, & Requena. (2011). Proposal of ontology for environmental impact assessment: An application with knowledge mobilization. *Expert Systems with Applications, 38*(3), 2462-2472.

Gennari, Musen, Fergerson, Grosso, Crubézy, Eriksson, Noy, & Tu. (2003). The evolution of Protégé: an environment for knowledge-based systems development. *International Journal of Human-computer studies, 58*(1), 89-123.

Gómez-Sal, Belmontes, & Nicolau. (2003). Assessing landscape values: a proposal for a multidimensional conceptual model. *Ecological modelling, 168*(3), 319-341.

Gómez-Pérez. (2001). Evaluation of ontologies. *International Journal of intelligent systems, 16*(3), 391-409.

Gong, Ma, Gong, Li, Li, & Yuan. (2018). Neo4j graph database realizes efficient storage performance of oilfield ontology. *PLoS One, 13*(11).

Gorawski, & Grochla. (2020). Performance tests of smart city IoT data repositories for universal linear infrastructure data and graph databases. *SN Computer Science, 1*(1), 31.

Gould, & Cheng. (2016). A prototype for ontology driven on-demand mapping of urban traffic accidents.

Gregor, Toral, Ariza, & Barrero. (2012). An ontology-based semantic service for cooperative urban equipments. *Journal of Network and Computer Applications, 35*(6), 2037-2050.

Grubic, & Fan. (2010). Supply chain ontology: Review, analysis and synthesis. *Computers in Industry, 61*(8), 776-786. doi:10.1016/j.compind.2010.05.006

Guba, & Lincoln. (1994). Competing paradigms in qualitative research. *Handbook of qualitative research, 2*(163-194), 105.

Guia, Soares, & Bernardino. (2017). *Graph Databases: Neo4j Analysis.* Paper presented at the ICEIS (1).

Haase, Broekstra, Eberhart, & Volz. (2004). A Comparison of RDF Query Languages. In *The Semantic Web – ISWC 2004* (pp. 502-517).

Halfawy. (2008). Integration of municipal infrastructure asset management processes: challenges and solutions. *Journal of Computing in Civil Engineering, 22*(3), 216-229.

Halfawy, Newton, & Vanier. (2006). Review of commercial municipal infrastructure asset management systems. *Electronic Journal of Information Technology in Construction, 11*, 211-224.

Hartig. (2019). *Foundations to Query Labeled Property Graphs using SPARQL*. Paper presented at the SEM4TRA-AMAR@ SEMANTICS.

Hoang, & Tjoa. (2006). *The state of the art of ontology-based query systems: A comparison of existing approaches*: Citeseer.

Holzschuher, & Peinl. (2013). *Performance of graph query languages*. Paper presented at the Proceedings of the Joint EDBT/ICDT 2013 Workshops on - EDBT '13.

Hood, & Wilson. (2001). The literature of bibliometrics, scientometrics, and informetrics. *Scientometrics, 52*, 291-314.

Hornsby, & King. (2008). Modeling motion relations for moving objects on road networks. *Geoinformatica, 12*(4), 477-495. doi:10.1007/s10707-007-0039-7

Horrocks, Patel-Schneider, & Van Harmelen. (2003). From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics, 1*(1), 7-26.

Houda, Khemaja, Oliveira, & Abed. (2010). *A public transportation ontology to support user travel planning.* Paper presented at the 2010 Fourth International Conference on Research Challenges in Information Science (RCIS).

Hu, Liu, Sugumaran, Liu, & Du. (2019). Automated structural defects diagnosis in underground transportation tunnels using semantic technologies. *Automation in Construction, 107*, 102929.

Hülsen, Zöllner, & Weiss. (2011). *Traffic intersection situation description ontology for advanced driver assistance.* Paper presented at the 2011 IEEE Intelligent Vehicles Symposium (IV).

Innovation. (2009). National guidelines for digital modelling. *Cooperative Research Centre for Construction Innovation, Brisbane, Australia*.

ISO. (2014). 55001: Asset management In *Overview, principles and terminology*.

Jelokhani-Niaraki, Sadeghi-Niaraki, & Kim. (2012). An ontology-based approach for managing spatio-temporal linearly referenced road event data. *Road & Transport Research, 21*(4), 38-49. Retrieved from <Go to ISI>://WOS:000320286700004

Jiang, & Wu. (2019). Estimation of environmental impacts of roads through life cycle assessment: a critical review and future directions. *Transportation Research Part D: Transport and Environment, 77*, 148-163.

Jiang, Zhu, Li, & Ji. (2020). Co-embedding of nodes and edges with graph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Jouili, & Vansteenberghe. (2013). *An empirical comparison of graph databases.* Paper presented at the 2013 International Conference on Social Computing.

Karimi, & Iordanova. (2021). Integration of BIM and GIS for construction automation, a systematic literature review (SLR) combining bibliometric and qualitative analysis. *Archives of Computational Methods in Engineering*, 1-22.

Kaza, & Hopkins. (2007). Ontology for land development decisions and plans. In *Ontologies for urban development* (pp. 47-59): Springer.

Khondoker, & Mueller. (2010). *Comparing ontology development tools based on an online survey*.

Killam. (2013). *Research terminology simplified: Paradigms, axiology, ontology, epistemology and methodology*: Laura Killam.

Kineber, Mohandes, ElBehairy, Chileshe, Zayed, & Fathy. (2022). Towards smart and sustainable urban management: A novel value engineering decision-making model for sewer projects. *Journal of cleaner production, 375*, 134069.

Kiritsis. (2013). Semantic technologies for engineering asset life cycle management. *International Journal of Production Research, 51*(23-24), 7345-7371. doi:10.1080/00207543.2012.761364

Konys. (2018). An ontology-based knowledge modelling for a sustainability assessment domain. *Sustainability, 10*(2), 300.

Koschmann. (1996). *Revolution and subjectivity in postwar Japan*: University of Chicago Press.

Koukias, & Kiritsis. (2015). Rule-based mechanism to optimize asset management using a technical documentation ontology. *IFAC-PapersOnLine, 48*(3), 1001-1006.

Koukias, Nadoveza, & Kiritsis. (2015a). *Towards Ontology-Based Modeling of Technical Documentation and Operation Data of the Engineering Asset*: Springer International Publishing.

Koukias, Nadoveza, & Kiritsis. (2015b). Towards Ontology-Based Modeling of Technical Documentation and Operation Data of the Engineering Asset. In *Engineering Asset Management-Systems, Professional Practices and Certification* (pp. 983-994): Springer.

Kremer, Lersy, De Sèze, Ferré, Maamar, Carsin-Nicol, Collange, Bonneville, Adam, & Martin-Blondel. (2020). Brain MRI findings in severe COVID-19: a retrospective observational study. *Radiology, 297*(2), E242-E251.

Kupriyanovsky, Pokusaev, Klimov, & Volodin. (2020). BIM on the way to IFC5-alignment and development of IFC semantics and ontologies with UML and OWL for road and rail structures, bridges, tunnels, ports, and waterways. *International Journal of Open Information Technologies, 8*(8), 69-78.

Lampoltshammer, & Wiegand. (2015). Improving the computational performance of ontology-based classification using graph databases. *Remote Sensing, 7*(7), 9473-9491.

Le, & Jeong. (2016). Interlinking life-cycle data spaces to support decision making in highway asset management. *Automation in Construction, 64*, 54-64. doi:10.1016/j.autcon.2015.12.016

Lécué, Tallevi-Diotallevi, Hayes, Tucker, Bicer, Sbodio, & Tommasi. (2014). Smart traffic analytics in the semantic web with STAR-CITY: Scenarios, system and lessons learned in Dublin City. *Journal of Web Semantics, 27*, 26-33.

Lécué, Tucker, Bicer, Tommasi, Tallevi-Diotallevi, & Sbodio. (2014). *Predicting severity of road traffic congestion using semantic web technologies.* Paper presented at the European semantic web conference.

Lee, Lee, & Kwan. (2017). Location-based service using ontology-based semantic queries: A study with a focus on indoor activities in a university context. *Computers Environment and Urban Systems, 62*, 41-52. doi:10.1016/j.compenvurbsys.2016.10.009

Li, Shen, Wu, & Yue. (2019). Integrating building information modeling and prefabrication housing production. *Automation in Construction, 100*, 46-60.

Li, Zhang, Shadmand, & Balog. (2017). Model predictive control of a voltage-source inverter with seamless transition between islanded and grid-connected operations. *IEEE Transactions on Industrial Electronics, 64*(10), 7906-7918.

Lim, Porras-Alvarado, & Zhang. (2019). Pricing of highway infrastructure for transportation asset management. *Built Environment Project and Asset Management*.

Little. (2016). Why is My Query Faster the Second Time it Runs? . *Dear SQL DBA.*

Liu, & Chetal. (2005). Trust-based secure information sharing between federal government agencies. *Journal of the American society for information science and technology, 56*(3), 283-298.

Liu, & El-Gohary. (2017). Ontology-based semi-supervised conditional random fields for automated information extraction from bridge inspection reports. *Automation in Construction, 81*, 313-327.

Liu, Hagedorn, & König. (2021a). *BIM-Based Organization of Inspection Data Using Semantic Web Technology for Infrastructure Asset Management.* Paper presented at the International Conference of the European Association on Quality Control of Bridges and Structures.

Liu, Hagedorn, & König. (2021b). AN ONTOLOGY INTEGRATING AS-BUILT INFORMATION FOR INFRASTRUCTURE ASSET MANAGEMENT USING BIM AND SEMANTIC WEB.

Liu, Ott, Goyal, Du, Joshi, Chen, Levy, Lewis, Zettlemoyer, & Stoyanov. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Lom, Pribyl, & Svitek. (2016). *Industry 4.0 as a part of smart cities.* Paper presented at the 2016 Smart Cities Symposium Prague (SCSP).

Mahajan, Girshick, Ramanathan, He, Paluri, Li, Bharambe, & Van Der Maaten. (2018). *Exploring the limits of weakly supervised pretraining.* Paper presented at the Proceedings of the European conference on computer vision (ECCV).

Malgundkar, Rao, & Mantha. (2012). GIS driven urban traffic analysis based on ontology. *International Journal of Managing Information Technology, 4*(1), 15.

McGuinness, & Van Harmelen. (2004a). OWL web ontology language overview. *W3C recommendation, 10*(10), 2004.

McGuinness, & Van Harmelen. (2004b). OWL Web Ontology Language Overview. W3C Recommendation, 2004. *URL http://www. w3. org/tr/2004/rec-owl-features-20040210*.

Merdan, Koppensteiner, Hegny, & Favre-Bulle. (2008). *Application of an Ontology in a Transport Domain.* Paper presented at the 2008 IEEE International Conference on Industrial Technology.

Mohammad, Kaloskampis, Hicks, & Setchi. (2015). Ontology-based framework for risk assessment in road scenes using videos. *Procedia Computer Science, 60*, 1532-1541.

Möller, & Beer. (2008). Engineering computation under uncertainty–capabilities of non-traditional models. *Computers & Structures, 86*(10), 1024-1041.

Morgan. (2012). Environmental impact assessment: the state of the art. *Impact Assessment and Project Appraisal, 30*(1), 5-14.

Morris, & Therivel. (2001). *Methods of environmental impact assessment* (Vol. 2): Taylor & Francis.

Motik, & Horrocks. (2006). *Problems with OWL Syntax.* Paper presented at the OWLED.

Murphy. (2012). *Machine learning: a probabilistic perspective*: MIT press.

Najafi, & Bhattachar. (2011). Development of a culvert inventory and inspection framework for asset management of road structures. *Journal of King Saud University - Science, 23*(3), 243-254. doi:10.1016/j.jksus.2010.11.001

Neumann, & Weikum. (2010). x-RDF-3X: fast querying, high update rates, and consistency for RDF databases. *Proceedings of the VLDB Endowment, 3*(1-2), 256-263.

Nguyen, & Nguyen. (2019). Fuzzy Ontology Based Model for Supporting Safe Driving. *International Journal of Computer Science and Network Security, 19*(7), 111-115. Retrieved from <Go to ISI>://WOS:000487275600014

Niestroj, McMeekin, & Helmholz. (2019). Introducing a Framework for Conflating Road Network Data with Semantic Web Technologies. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-2/W5*, 231-238. doi:10.5194/isprs-annals-IV-2-W5-231-2019

Niestroj, McMeekin, Helmholz, & Kuhn. (2018). A PROPOSAL TO USE SEMANTIC WEB TECHNOLOGIES FOR IMPROVED ROAD NETWORK INFORMATION EXCHANGE. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences, 4*(4).

Niknam, & Karshenas. (2017). A shared ontology approach to semantic representation of BIM data. *Automation in Construction, 80*, 22-36. doi:10.1016/j.autcon.2017.03.013

Noy, Crubézy, Fergerson, Knublauch, Tu, Vendetti, & Musen. (2003). *Protégé-2000: an open-source ontology-development and knowledge-acquisition environment.* Paper presented at the AMIA... Annual Symposium proceedings. AMIA Symposium.

Noy, & McGuinness. (2001). Ontology development 101: A guide to creating your first ontology. In: Stanford knowledge systems laboratory technical report KSL-01-05 and ….

Nyulas, O'connor, & Tu. (2007). *Datamaster–a plug-in for importing schemas and data from relational databases into protege.* Paper presented at the 10th international Protégé conference.

O'Malley. (1999). The Integrated Pollution Prevention and Control (IPPC) Directive and its implications for the environment and industrial activities in Europe. *Sensors and Actuators B: Chemical, 59*(2-3), 78-82.

Palerm. (1999). Public participation in environmental impact assessment in Spain: three case studies evaluating national, Catalan and Balearic legislation. *Impact Assessment and Project Appraisal, 17*(4), 259-271.

Park, Lee, & Kim. (2016). A study on analysis of the environmental load impact factors in the planning stage for highway project. *KSCE Journal of Civil Engineering, 20*(6), 2162-2169.

Parundekar, Knoblock, & Ambite. (2010). *Linking and building ontologies of linked data.* Paper presented at the International Semantic Web Conference.

Pauwels, Zhang, & Lee. (2017). Semantic web technologies in AEC industry: A literature overview. *Automation in Construction, 73*, 145-165.

Piyatrapoomi, Kumar, & Setunge. (2004). Framework for Investment Decision-Making under Risk and Uncertainty for Infrastructure Asset Management.

Research in Transportation Economics, 8, 199-214. doi:10.1016/s0739-8859(04)08010-2

Pokusaev, Kupriyanovsky, Klimov, Namiot, Kupriyanovsky, & Zarechkin. (2020). BIM, Ontology and Asset Management Technologies on European Highways. *International Journal of Open Information Technologies, 8*(6), 108-135.

Qu, & Tang. (2019). Probabilistic logic neural networks for reasoning. *Advances in neural information processing systems, 32*.

Quinn, & McArthur. (2021). A case study comparing the completeness and expressiveness of two industry recognized ontologies. *Advanced Engineering Informatics, 47*, 101233.

Rahman, Medhi, & Hussain. (2023). DynO-IoT: a dynamic ontology for provisioning semantic interoperability in internet of things. *International Journal of Sensor Networks, 41*(2), 114-125.

Raja, Mondal, & Jawahar. (2020). *Table structure recognition using top-down and bottom-up cues.* Paper presented at the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16.

Reddy, & Veeraragavan. (2011). Application of Highway Development and Management Tool in Rural Road Asset Management. *Transportation Research Record: Journal of the Transportation Research Board, 2204*(1), 29-34. doi:10.3141/2204-04

Saikaew, Asawamenakul, & Buranarach. (2014). Design and evaluation of a NoSQL database for storing and querying RDF data. *Engineering and Applied Science Research, 41*(4), 537-545.

Scheuermann, & Leukel. (2014). Supply chain management ontology from an ontology engineering perspective. *Computers in Industry, 65*(6), 913-923.

Schmidt, Hornung, Küchlin, Lausen, & Pinkel. (2008). *An experimental comparison of RDF data management approaches in a SPARQL benchmark scenario.* Paper presented at the International Semantic Web Conference.

Scholer, Williams, Yiannis, & Zobel. (2002). *Compression of inverted indexes for fast query evaluation.* Paper presented at the Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval.

Schönbeck, Löfsjögård, & Ansell. (2020). Quantitative review of construction 4.0 technology presence in construction project research. *Buildings, 10*(10), 173.

Schram, & Anderson. (2012). *MySQL to NoSQL: data modeling challenges in supporting scalability*. Paper presented at the Proceedings of the 3rd annual conference on Systems, programming, and applications: software for humanity.

Sfoungari. (2021). *Automatic maintenance of COVID-19 related Knowledge Graphs based on large-scale information extraction in scientific literature.*

Shaaban, Schmittner, Gruber, Mohamed, Quirchmayr, & Schikuta. (2020). An Automated Ontology-Based Security Requirements Identification for the Vehicular Domain. *J. Data Intell., 1*(4), 401-418.

Shaaban, Schmittner, Gruber, Mohamed, Quirchmayr, & Schikuta. (2021). Ontology-Based Security Requirements Framework for Current and Future Vehicles. In *Data Science and Big Data Analytics in Smart Environments* (pp. 197-217): CRC Press.

Sharma, Sharma, & Bundele. (2018). *Performance Analysis of RDBMS and No SQL Databases: PostgreSQL, MongoDB and Neo4j*. Paper presented at the 2018

3rd International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE).

Shi, Zhu, & Li. (2022). *Research on automatic text summarization technology based on ALBERT-TextRank.* Paper presented at the 2022 5th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE).

Simeone, Cursi, & Acierno. (2019). BIM semantic-enrichment for built heritage representation. *Automation in Construction, 97*, 122-137.

Smith. (2012). Ontology. In *The furniture of the world* (pp. 47-68): Brill Rodopi.

Song, Price, Guvenen, Bloom, & Von Wachter. (2019). Firming up inequality. *The Quarterly journal of economics, 134*(1), 1-50.

Stephan, Pascal, & Andreas. (2007). Knowledge representation and ontologies logic, ontologies and semantic web languages. *CiteSeerX*.

Stocker, Rönkkö, & Kolehmainen. (2012). Making sense of sensor data using ontology: A discussion for road vehicle classification.

Svetel, & Pejanović. (2010). The role of the semantic web for knowledge management in the construction industry. *Informatica, 34*(3).

Tang, Fan, Ni, & Shen. (2016). Environmental impact assessment in Hong Kong: a comparison study and lessons learnt. *Impact Assessment and Project Appraisal, 34*(3), 254-260. doi:10.1080/14615517.2016.1177934

Tensmeyer, Morariu, Price, Cohen, & Martinez. (2019). *Deep splitting and merging for table structure decomposition.* Paper presented at the 2019 International Conference on Document Analysis and Recognition (ICDAR).

Thakare, Khire, & Kumbhar. (2022). Artificial intelligence (AI) and Internet of Things (IoT) in healthcare: opportunities and challenges. *ECS Transactions, 107*(1), 7941.

Thakkar, & Lohiya. (2020). A Review on Machine Learning and Deep Learning Perspectives of IDS for IoT: Recent Updates, Security Issues, and Challenges. *Archives of Computational Methods in Engineering*, 1-33.

Thakkar, Punjani, Lehmann, & Auer. (2018). *Two for one: Querying property graph databases using SPARQL via gremlinator.* Paper presented at the Proceedings of the 1st ACM SIGMOD Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA).

Toner. (2009). Small is not too small: Reflections concerning the validity of very small focus groups (VSFGs). *Qualitative Social Work, 8*(2), 179-192.

Toulni, Nsiri, Boulmalf, & Sadiki. (2015). An ontology based approach to traffic management in urban areas. *International Journal of Systems Applications, Engineering & Development, 9*.

Urbieta, Nieto, García, & Otaegui. (2021). Design and Implementation of an Ontology for Semantic Labeling and Testing: Automotive Global Ontology (AGO). *Applied Sciences, 11*(17), 7782.

Vallejo, Albusac, Jimenez, Gonzalez, & Moreno. (2009). A cognitive surveillance system for detecting incorrect traffic behaviors. *Expert Systems with Applications, 36*(7), 10503-10511.

Van de Vyvere, Colpaert, Mannens, & Verborgh. (2019). *Open traffic lights: a strategy for publishing and preserving traffic lights data.* Paper presented at the Companion Proceedings of The 2019 World Wide Web Conference.

Vicknair, Macias, Zhao, Nan, Chen, & Wilkins. (2010). *A comparison of a graph database and a relational database: a data provenance perspective.* Paper presented at the Proceedings of the 48th annual Southeast regional conference.
W3C. (2020). Semantic Web Standard. In.
Wang, & Wang. (2011). *An ontology-based traffic accident risk mapping framework.* Paper presented at the International Symposium on Spatial and Temporal Databases.
Wang, Wu, Chi, & Li. (2020). Adopting lean thinking in virtual reality-based personalized operation training using value stream mapping. *Automation in Construction, 119*, 103355.
Wang, Zhang, Gu, & Pung. (2004). *Ontology based context modeling and reasoning using OWL.* Paper presented at the IEEE annual conference on pervasive computing and communications workshops, 2004. Proceedings of the second.
Wathern. (2013). *Environmental impact assessment: theory and practice*: Routledge.
Watson, Watson, & Vallmuur. (2015). Estimating under-reporting of road crash injuries to police using multiple linked data collections. *Accident Analysis & Prevention, 83*, 18-25.
Werbrouck, Pauwels, Bonduel, Beetz, & Bekers. (2020). Scan-to-graph: Semantic enrichment of existing building geometry. *Automation in Construction, 119*, 103286.
Wood. (2000). Ten years on: an empirical analysis of UK environmental statement submissions since the implementation of Directive 85/337/EEC. *Journal of Environmental Planning and Management, 43*(5), 721-747.
Wu, Qin, & Wan. (2019). TOST: A Topological Semantic Model for GPS Trajectories Inside Road Networks. *ISPRS International Journal of Geo-Information, 8*(9), 410.
Wu, Wang, Wu, Wang, Jiang, Chen, & Swapan. (2021). Hybrid deep learning model for automating constraint modelling in advanced working packaging. *Automation in Construction, 127*, 103733.
Wu, Wu, Wang, Jiang, Chen, & Wang. (2021). Ontological knowledge base for concrete bridge rehabilitation project management. *Automation in Construction, 121*, 103428.
Xu, & Cai. (2020). Semantic approach to compliance checking of underground utilities. *Automation in Construction, 109*, 103006.
Yabuki, Kikushige, & Fukuda. (2011). A Management System of Roadside Trees Using RFID and Ontology. In *Computing in Civil Engineering (2011)* (pp. 307-314).
Yang, Cormican, & Yu. (2019). Ontology-based systems engineering: A state-of-the-art review. *Computers in Industry, 111*, 148-171.
Yang, Yang, & Cohen. (2017). Differentiable learning of logical rules for knowledge base reasoning. *Advances in neural information processing systems, 30*.
Ye, Yang, Jiang, & Tong. (2008). An ontology-based architecture for implementing semantic integration of supply chain management. *International Journal of Computer Integrated Manufacturing, 21*(1), 1-18.
Yin, Neubig, Yih, & Riedel. (2020). TaBERT: Pretraining for joint understanding of textual and tabular data. *arXiv preprint arXiv:2005.08314*.
Zapater, Escrivá, García, & Durá. (2015). Semantic web service discovery system for road traffic information services. *Expert Systems with Applications, 42*(8), 3833-3842.

Zaveri, Rula, Maurino, Pietrobon, Lehmann, Auer, & Hitzler. (2013). Quality assessment methodologies for linked open data. *Submitted to Semantic Web Journal, 15*, 16.

Zeb. (2017). An eco asset ontology towards effective eco asset management. *Built Environment Project and Asset Management*.

Zeb, Froese, & Vanier. (2015). An ontology-supported asset information integrator system in infrastructure management. *Built Environment Project and Asset Management*.

Zhai, Chen, Yu, Liang, & Jiang. (2009). Fuzzy Semantic Retrieval for Traffic Information Based on Fuzzy Ontology and RDF on the Semantic Web. *JSW, 4*(7), 758-765.

Zhang, Boukamp, & Teizer. (2015). Ontology-based semantic modeling of construction safety knowledge: Towards automated safety planning for job hazard analysis (JHA). *Automation in Construction, 52*, 29-41. doi:10.1016/j.autcon.2015.02.005

Zhang, & El-Gohary. (2013). Information transformation and automated reasoning for automated compliance checking in construction. In *Computing in civil engineering (2013)* (pp. 701-708).

Zhang, Li, Zhao, & Ren. (2018). An ontology-based approach supporting holistic structural design with the consideration of safety, environmental impact and cost. *Advances in Engineering Software, 115*, 26-39.

Zhang, & Yin. (2008). Exploring Semantic Web technologies for ontology-based modeling in collaborative engineering design. *The International Journal of Advanced Manufacturing Technology, 36*(9-10), 833-843.

Zhao, & Ichise. (2014). Ontology Integration for Linked Data. *Journal on Data Semantics, 3*(4), 237-254. doi:10.1007/s13740-014-0041-9

Zhao, Ichise, Yoshikawa, Naito, Kakinami, & Sasaki. (2015). *Ontology-based decision making on uncontrolled intersections and narrow roads.* Paper presented at the 2015 IEEE intelligent vehicles symposium (IV).

Zhong, He, Huang, Love, Tang, & Luo. (2020). A building regulation question answering system: A deep learning methodology. *Advanced Engineering Informatics, 46*, 101195.

Zhong, Xing, Luo, Zhou, Li, Rose, & Fang. (2020). Deep learning-based extraction of construction procedural constraints from construction regulations. *Advanced Engineering Informatics, 43*, 101003.

Zhou, & Tao. (2011). *A framework for ontology-based knowledge management.* Paper presented at the 2011 International Conference on Business Management and Electronic Information.

Zhu, Chong, Zhao, Wu, Tan, & Xu. (2022). The Application of Graph in BIM/GIS Integration. *Buildings, 12*(12), 2162.

Zhu, Wang, Wang, Wu, & Kim. (2019). Integration of BIM and GIS: Geometry from IFC to shapefile using open-source technology. *Automation in Construction, 102*, 105-119.

Appendix (all other materials related to the study)

Appendix 1 List of publications

Lei, X., Wu, P., Zhu, J., & Wang, J. (2021). Ontology-based information integration: A state-of-the-art review in road asset management. *Archives of computational methods in engineering*, 1-19.

Lei, X., Wu, P. (2022, June). An novel environmental impact assessment ontology using a graph-based database In *CIB World Building Congress* (pp. 147-156).

Zhu, J., Wu, P., & **Lei, X.** (2023). IFC-graph for facilitating building information access and query. *Automation in Construction*, *148*, 104778.

Appendix 2 Focus group questions

Part 1: General information

1. What is your background?
2. How many years of experience do you have?
3. What is your position/level?
4. What are your main duties in the role?

Part 2： Evaluation method

Combined with other well-accepted data visualization evaluation methods, a basic functional and availability comparison has been conducted to highlight the features of different data models. A fourteen-criteria evaluation system is listed below.

- Visualization method: the method (structure) of the visualization function.
- Large ontology capacity: the ability to present large datasets (e.g., over 10 k triples/vertexes).
- List review: review the information as lists.
- Table review: review the information as tables.
- Zooming: zoom in or out of the figure.
- History tracking: show the history of the action.
- Query: query the information directly in the visualized information.
- Filter: choose the presenting information by filter.
- Click selecting: select the information by clicking.
- Drag and drop: move the information smoothly.
- Textual editing: edit textual information directly.
- Visual editing: change the visual style.
- Class checking: review the class information directly.
- Annotation: perform the annotation function
- Property characteristics: review the embedded properties on the inference.

After introducing and demonstrating these functions with running cases, participants are invited to rate each function (if available) on a scale from 1 (worst) to 5 (best) based on the questionnaire.

How does the model perform in this aspect? Or: How do you think the model performed in this visualization aspect?	Not useful at all (1)	2	3	4	Excellent (5)
Visualization method					
Large ontology capacity					
List review					
Table review					
Zooming					
History tracking:					
Query					
Filter					
Click selecting					
Drag and drop					
Textual editing					
Visual editing					
Class checking					
Annotation					
Property characteristics					

Part 3: Open-ended questions

After marking all aspects, participants will be presented with some open-ended questions, given their understanding of ontology and the differences between models. The listed potential questions are as follows, though additional valuable questions may arise during the discussion:

1. What is the typical information flow approach in an engineering project or management process?

2. How do you perceive the utility of ontology and information visualization in project management or other processes?
3. In your opinion, which model is most suitable for engineering projects?
4. Are ontology and information visualization currently applied in any step of project management?
5. Regarding the integration of information from various organizations, groups, and documents with different formats, do you believe it leads to a waste of time and budget? What challenges do you foresee?

Borgo, Ferrario, Gangemi, Guarino, Masolo, Porello, Sanfilippo, & Vieu. (2022). DOLCE: A descriptive ontology for linguistic and cognitive engineering. *Applied ontology, 17*(1), 45-69.

Du, Wei, Dimitrova, Magee, Clarke, Collins, Entwisle, Torbaghan, Curioni, & Stirling. (2023). City infrastructure ontologies. *Computers, Environment and Urban Systems, 104*, 101991.

Espinoza-Arias, Garijo, & Corcho. (2022). *Extending ontology engineering practices to facilitate application development.* Paper presented at the International Conference on Knowledge Engineering and Knowledge Management.

He, Chen, Dong, Horrocks, Allocca, Kim, & Sapkota. (2023). DeepOnto: A Python package for ontology engineering with deep learning. *arXiv preprint arXiv:2307.03067*.

Isailović, & Hajdin. (2022). *Ontologies as the Key for Common Understanding of Infrastructure Assets.* Paper presented at the Proceedings of the 1st Conference of the European Association on Quality Control of Bridges and Structures: EUROSTRUCT 2021 1.

Liu, Ott, Goyal, Du, Joshi, Chen, Levy, Lewis, Zettlemoyer, & Stoyanov. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Poveda-Villalón, Fernández-Izquierdo, Fernández-López, & García-Castro. (2022). LOT: An industrial oriented ontology engineering framework. *Engineering Applications of Artificial Intelligence, 111*, 104755.

Spoladore, Pessot, & Trombetta. (2023). A novel agile ontology engineering methodology for supporting organizations in collaborative ontology development. *Computers in Industry, 151*, 103979.