

**USING SEMANTIC WEB TECHNOLOGIES TO RECOMMEND SUSTAINABLE
BUILDING TECHNOLOGY PRODUCTS**

Zainab Shamsiya Usman

A thesis submitted to the Faculty of Technology, Design and Environment in
fulfilment of the requirements for the degree of
Doctor of Philosophy

**OXFORD BROOKES UNIVERSITY,
OXFORD, UNITED KINGDOM**

**Director of Studies
Professor Joseph Tah, Oxford Brookes University**

**Co-director of Studies
Dr. Henry Abanda, Oxford Brookes University
Dr. Charles Nche, American University of Nigeria**

July, 2023

Abstract

The rise in global concerns over the continuous increase of greenhouse gas emissions and the increasing demand for energy, has led to an increase in the uptake of sustainable building technology products and photovoltaic (PV) systems technology. However, despite the wide adoption of PV globally as a substitute to the grid, it faces information-related challenges that hinder its adoption. The enormous volumes of information available on these technology products overwhelm intended users when acquiring valuable knowledge to make informed decisions. Similarly, information on PV is represented on the web using non-machine-processable languages that focus on describing the pattern of data rather than execution models. The semantic web and its associated technologies have become state-of-the-art solutions to such knowledge modelling, representation, and reasoning challenges.

This research aims to exploit the extent to which semantic web technologies can be used for knowledge modelling in the domain of sustainable building technology products with emphasis on photovoltaic systems. This research develops a unique methodological framework to develop rich and comprehensive theoretical and conceptual knowledge models of the domain. As part of operationalizing the methodology, the research investigates the integration of semantic web with the standardised information foundation classes (IFC) for efficient knowledge discovery. This led to the development of a rich and modular ontology for the PV domain called the *IfcOWL_pvOntology* that extends the standard *IfcOWL* ontology to represent knowledge on photovoltaic systems. Consequently, this research further investigates and proposes a selection model with an algorithmic approach for sizing and selection of photovoltaic technology components involving multicriterial design parameters. Backed by mathematical computations, this process model is used alongside the *IfcOWL_pvOntology* to develop a successfully evaluated semantic web-based navigable prototype system that provides recommendations on choices of PV technology products to intended users.

The prototype system is developed to serve as a proof-by-demonstration and a proof of the concepts established in investigating the extent to which semantic web technologies can be exploited to provide decision support in sizing and selection of PV systems to intended users. The decision to extend IFC is paramount because only continuous use can make standards advance. The proposed ontology could be adopted and approved by the buildingSMART working group in the world wide web consortium (W3C) as an official extension of IFC ontology for the photovoltaic technology domain. This prototype will be treated as part of a future large expert system where the knowledge and inference models will be used in practice in the industry to provide decision support for not only photovoltaic systems but other sustainable building technology products at large.

Table of Content

1. Introduction	7
1.1. Introduction	8
1.2. Rationale	9
1.3. Identification of Problem Domain	13
1.4. Aim and Objectives	15
1.5. Summary of the Research Methodology	16
1.6. Research Achievements	17
1.7. Organisation of the Thesis	19
2. Review on Sustainable Building Technology Products	24
2.1. Introduction	24
2.2. Sustainability and Sustainable Development	24
2.2.1. Need for Sustainable Construction	25
2.3. Sustainable Building Technology Products	27
2.4. Renewable Energy (RE) Technologies	29
2.5. Overview of Photovoltaic System Technology	35
2.5.1. Components of the Photovoltaic System	37
2.5.2. Types of Connections	40
2.5.3. Methods Involved in Photovoltaic Systems (PVS) Design	42
2.5.3.1. Geolocation and Solar Irradiance	42
2.5.3.2. Determining the Energy Load of the Building	44
2.5.3.3. Sizing and Selecting Components of Photovoltaic Systems	46
2.5.3.3.1. Modules	46
2.5.3.3.2. Inverter	48
2.5.3.3.3. Battery	48
2.5.3.3.4. Charge Controller	50
2.5.3.3.5. Wires and Cables	50
2.5.4. Mounting Types	51
2.6. Classification and Rating Standards for Construction Products	53
2.7. Advantages of Adopting Photovoltaic Systems	54
2.8. Challenges in the Uptake of Photovoltaic Systems	56
2.8.1. Poor Performance of Installed PV Systems	56
2.8.2. Challenges in Sizing and Selection of PV Components	60
2.8.3. Abundance of Unstructured Data on Products Over the Web	62
2.9. Conclusion	63
3. Review on Semantic Web Technologies	65
3.1. Introduction	65
3.2. Weakness of the Current Web Technology	65
3.2.1. Presentation of Information	66
3.2.2. Current web Search Techniques	66
3.3. Semantic Web Architecture and Standards	68
3.4. Ontological Engineering	73
3.4.1. Reason for Developing Ontology	74

3.4.2. Earlier Ontology Development Methodologies	76
3.4.2.1. Uschold and King and Toronto Virtual Enterprise Methodology	77
3.4.2.2. KACTUS and METHONTOLOGY	77
3.4.2.3. SENSUS Methodology	78
3.4.2.4. On-To-Knowledge Methodology	78
3.4.2.5. Ontology Development 101	78
3.4.2.6. NeOn Methodology	79
3.4.3. Other Ontology Re-Engineering Methodologies	80
3.4.4. Ontology Engineering Tools	82
3.5. Software Engineering	85
3.6. Ontology Evaluation	89
3.7. Existing Domain Ontologies for Possible Re-Use	91
3.8. Applications of Semantic Web in Construction	93
3.9. Conclusion	96
<i>4. Overview of Building Information Modelling, Industry Foundation Classes And Applications of Semantic Web</i>	97
4.1. Introduction	97
4.2. Overview of Building Information Models (BIM)	98
4.2.1. Applications of BIM in Construction Domain	100
4.3. Overview of Industry Foundation Classes (IFC)	102
4.3.1. IFC Formats and Schema Specifications	103
4.3.2. IFC Data Schema	106
4.4. Industry Foundation Classes (IFC) and Semantic Web	108
4.4.1. From EXPRESS to Web Ontology Language (OWL), the ifcOWL Ontology	109
4.4.2. Future of IFC: EXPRESS to Unified Modelling Language	110
4.5. Industry Foundation Classes (IFC) in the Context of this Research	110
4.6. Applications of Semantic Web with BIM and IFC in Construction	112
4.7. Conclusion	115
<i>5. Research Design and Methodology</i>	116
5.1. Introduction	117
5.2. Overview of Research Methodologies	117
5.2.1. Justification for Exploratory Case-Study Research Methods	120
5.3. Knowledge Engineering	121
5.3.1. Justification for CommonKADS Knowledge Engineering Methodology	124
5.4. Methodologies Adopted in the Research	125
5.4.1. Justification for Agile Software Development Methodology	126
5.4.2. Justification for Ontology Development 101	128
5.5. Research Methodological Framework	128
5.5.1. Knowledge Identification	133
5.5.1.1. Domain Familiarisation	133
5.5.1.2. Identify Potential Reusable Components	134
5.5.2. Knowledge Specification	135
5.5.2.1. Select Task Template	135
5.5.2.2. Initial Domain Conceptualization	136
5.5.2.3. Complete Knowledge Specification	137

5.5.3. Knowledge Refinement	137
5.5.3.1. Initial Domain Verification	137
5.5.3.2. Complete Knowledge Refinement	137
5.5.4. Prototype Development	138
5.5.4.1. Prototype Design	138
5.5.4.2. Prototype Implementation	138
5.5.4.3. Prototype Evaluation	140
5.5.4.4. Prototype Documentation	140
5.5.5. Recommendations for Use in Practice	141
5.6. Conclusion	141
6. Knowledge Modelling (Acquisition and Representation) of Sustainable Building Technology Products and Photovoltaic Technology	142
6.1. Introduction	142
6.2. Knowledge Identification	142
6.2.1. Domain Familiarisation	143
6.2.2. Knowledge Analysis	143
6.2.3. Identify Potential Knowledge Components for Reuse	144
6.3. Knowledge Specification	148
6.3.1. Selecting a Type of Task Template	148
6.3.2. Initial Domain Knowledge Modelling	149
6.3.2.1. Initial Establishment of Concepts for Sustainable Building Technology Products	149
6.3.2.2. Initial Establishment of Concepts for Photovoltaic Systems	155
6.3.2.3. Concepts Reused from Existing Domain Ontologies	162
6.4. Complete Conceptual Knowledge Specification for the Domain	165
6.4.1.1. Establish Annotation Properties	165
6.4.1.2. Determine Object Properties	166
6.4.1.3. Determine Data Properties	166
6.5. Alignment of Datatype Properties using IFC Schema Specification	167
6.6. Knowledge Refinement	175
6.7. Instance Acquisition for Ontology Development	175
6.8. Conclusion	175
7. DESIGN AND DEVELOPMENT OF PROTOTYPE SYSTEM	176
7.1. Introduction	177
7.2. System Requirements and Specifications	178
7.3. Software Components and Tools	180
7.4. System Architecture	182
7.5. Ontology Development	184
7.5.1. Proposed Approach to Integrating Semantic Web and IFC	185
7.5.1.1. Model View Definition (MVD) of IFC	185
7.5.1.2. Conversion of IFC Specifications into Semantic Web	186
7.5.2. Extending IfcOWL Ontology	188
7.5.3. Transformation of the Conceptual Model (in UML) into Corresponding OWL	190
7.5.3.1. Classes	192
7.5.3.2. Properties of Classes	195

7.5.3.3. Facets of Properties	202
7.6. Storage of Ontology	204
7.7. Addition of Instances by Product Manufacturers	206
7.8. Conclusion	210
8. DEVELOPMENT OF SEMANTIC WEB IFC-BASED PHOTOVOLTAIC SYSTEM (SIPS)	211
8.1. Introduction	211
8.2. System Requirements and Specifications	212
8.3. Software Components and Tools	214
8.4. System Architecture	217
8.5. Development of Semantic Web-Based Application	218
8.5.1. Back-End System Configuration	219
8.5.2. Extraction of Knowledge from Ontology into Eclipse Using Jena	220
8.5.3. Photovoltaic Component Sizing and Computations	228
8.5.4. Front-End - Client User Interface	236
8.5.5. Display of Recommendations	240
8.5.6. Manufacturer Webpages	244
8.6. Conclusion	247
9. PROTOTYPE EVALUATION	247
9.1. Introduction	248
9.2. Semantic and Syntactic Verification of Ontology	249
9.2.1. Semantic and Syntactic Anomalies	250
9.2.2. Checking and Clearing Inconsistency, Incompleteness and Redundancy Errors	251
9.3. Quality Check Evaluation	257
9.4. Instance Data Evaluation	259
9.5. Validation Using a Case Study	260
9.5.1. Description of Case Study	261
9.5.2. Abstraction of Knowledge into Ontology	264
9.5.3. Design and Selection of PV-system Components for Oxford Brookes University	267
9.5.3.1. Module Sizing and Components Selections	269
9.5.3.2. Display of Results	271
9.6. Conclusion	275
10. CONCLUSION	276
10.1. General	277
10.2. Attainment of Research Objectives	278
10.3. Key Findings from Literature	280
10.4. Contribution to Knowledge	282
10.5. Implications for Practice	284
10.6. Challenges and Limitations of the Research	285
10.7. Recommendation for Further Studies	287
REFERENCE	289
APPENDIX	309
Appendix 1: Energy load Guide	309

1. Introduction

1.1. Introduction

The environmental impacts of greenhouse gases being generated from construction activities have been comprehensively investigated and documented (Shafique, Luo and Zuo, 2020; Liao and Li, 2022). The studies have informed the design of new policies, measures, and targets to address impacts of greenhouse gas emissions caused by the construction industry by governments and environmental agencies globally. To ensure set emissions targets are met, these agencies and governments recommend changes in practice such as substituting high carbon emission products with Sustainable Building Technology Products (SBTPs) to minimise and subsequently reduce the environmental impacts from construction. Furthermore, their recommendations are made appealing through grants and financial support that are given to construction projects that use these SBTPs to mitigate environmental impacts. The benefits of the uptake of SBTPs have gone beyond a matter of personal choice. The use of sustainable building materials has become a new trend in the construction industry (Beroe Inc., 2021). They have been implemented in projects aimed at increasing energy efficiency and cost savings. Individuals are more inclined towards acquiring homes with high sustainability documentation where energy-efficient materials are used. This ensures reduced energy costs of the building in the long-run (Beroe Inc., 2021).

However, despite the uptake of SBTPs and available incentives from the government, several factors continue to hinder a rapid growth in their use. These include the high initial cost, exacting institutional and government permits, and unclear long-term effects (Ogunmakinde, et al., 2016, Beroe Inc., 2021). Relatedly, the greatest challenge that constrains the quick adoption of sustainable building technologies is the lack of adequate knowledge about sustainable products amongst construction professionals and end-users. (SolarFeed, 2021). Although basic information about SBTPs is available from books or the web, they cannot be exploited by either professionals or end-users to make informed decisions about the selection or optimization of these products. Therefore, there is a need for

better ways of representing knowledge on SBTPs so they can be used by construction professionals. The rationale of this study is based on solving the above mentioned challenges. This study explores an emerging semantic web technology for knowledge acquisition and representation of SBTPs domain. This is so that obtained knowledge is meaningful to humans or/and end-users as well as computers and exploited to provide solutions to the challenges of the problem domain. A more detailed discussion of the problem domain is provided later in the chapter after the rationale of the study is presented in the ensuing section.

1.2. Rationale

It is no longer news that there has been a rise in global concern over the continuous increase of greenhouse gas emissions and the negative impacts they have on the environment (Lee, 2015; Shafique, Luo and Zuo, 2020). While methane is often considered the most potent greenhouse gas, the worst in terms of greenhouse sustainability effects is carbon dioxide (CO₂). This is because it is more abundant than other greenhouse gases and it strongly infiltrates the earth's pure atmosphere (Ecobin, 2021). While CO₂ obtained from the natural exhaling of humans, animals, plants, and emissions from volcanic eruptions cannot be controlled, emission of this gas and other greenhouse gases from combustion in vehicles and industries can, and ought to be minimised. In 2020, due to Covid-19 pandemic, carbon emissions dropped in advanced economies on a per capita basis. For instance, CO₂ emissions dropped by 4% below its level in 2019 in the United States, by 3.7% in Japan, and by 2.4% in the European Union (IEA, 2022a). However, in 2021, a report provided by the International Energy Agency (IEA) showed the global carbon footprint rebounded to a record high of 36.3 billion tonnes with China's emissions accounting to 11.9 billion tonnes, making 33% of the global total (IEA, 2022a). In 2022, global energy-related emissions increased by 321 metric tons (IEA, 2023a).

The construction industry has a large impact on sustainability due to the nature of the heavy types of machinery used in construction activities and their

respective carbon emissions (Liao and Li, 2022). The entire life cycle of buildings contributes significantly to the emission of greenhouse gases, from construction to demolition. The construction industry accounts for 35% of the global carbon emissions (Liu, Pyplacz, Ermakova, & Konev, 2020). For instance, in the United Kingdom (UK), carbon emissions generated directly from buildings (from the use of fossil fuels for heating) make up 17% of the total greenhouse gas emissions, while indirect building emissions (from electrical appliances) account for 23% of the total (Climate Change Committee, 2020). Furthermore, construction and demolition waste also contribute to carbon emissions. These include emissions from machinery used in waste transportation, decomposition of waste, and the release of contaminants and pollutants into the landfill. In the UK, about 40% of landfill waste comes from building and construction waste (GOV.UK, 2021a), while in the United States, 15% of its total emissions comes from its landfill waste (Ecobin, 2021).

These concerns call for sustainability goals such as the effective use of natural resources and minimization of environmental degradation and waste caused by construction. To tackle these climate impacts, governments and industries have begun to promote the adoption of sustainable practices and strict emission mitigation measures. Various agencies around the world have put pressure on the construction industry to incorporate an important aspect of sustainable construction, that is, the adoption of sustainable building technology products into construction projects. These are renewable, reusable, or recyclable products manufactured from pure and clean sources which lessen the adverse consequences of environmental impacts of carbon emissions. Thus, construction projects that adopt SBTPs are widely supported through huge government subsidies and incentives as an encouragement to reduce carbon emissions (EnergySavingsTrust, 2021). The World Green Building Council (WorldGBC) has called for a 40% less carbon emission target by 2030 and 100% net zero emissions buildings by 2050 (WorldGBC, 2019). In Sweden, Finland and Norway, approaches have been introduced to promote voluntary reduction in embodied carbon to ensure the global carbon emission targets are met (WorldGBC, 2019). In the UK, the government has set its targets and minimum requirements that

construction companies must abide by. In their recent sixth carbon budget, an ambitious target was set to reduce carbon emissions by 78% by 2035 and attain a net-zero emission by 2050 (GOV.UK, 2021b). It is evident that controlling emissions of greenhouse gases may be the only absolute solution to global warming (Ecobin, 2021).

The driving factors explained in the paragraphs above have made sustainable building technologies gain prominence as a new global trend in the construction industry (Beroe Inc., 2021). However, a more rapid increase in the use of SBTPs is required to meet the set carbon emission targets. Despite the appealing advantages of SBTPs, availability of government incentives, and support for the adoption of these technologies, many construction companies are still reluctant to adopt them fully. Some of the reasons include high initial costs of using sustainable materials, strict regulations, exacting institutional and government permits, unclear long-term effects, and overwhelming amount of information about products constrains the quick adoption of sustainable building technologies (Beroe Inc., 2021, Gehry, 2021, Ogunmakinde, et al., 2016, SolarFeed 2021). Furthermore, they argue that using sustainable building technology is more complex than conventional technologies, thereby requiring artisans and expert service providers. They further claim that finding such experts are often challenging and time-consuming thereby making sustainable technologies less desirable to use. It has been argued that there is a lack of adequate education, training, knowledge, and integrated research on the use of sustainable technologies (Ogunmakinde, et al., 2016, Solar Feeds, 2021). This study will not provide a detailed analysis on the numerous barriers that hinder the uptake of sustainable technologies due to the wide scope involved. However, it will focus on two information-related challenges to adopting sustainable building technologies.

Firstly, the need for sustainable construction has led to the development of state-of-the-art sustainable technologies that are energy-efficient and optimise the performance of construction processes and buildings. Thousands of these sustainable building technologies are available in the market with distinctive features. The enormous volume of information has made it difficult for

construction professionals or intended users to acquire valuable knowledge to make informed decisions about technologies to use. Hence, better knowledge management about these technologies need to be implemented using knowledge modelling techniques to facilitate decision-making in the construction domain. This raises a number of research challenges. What are the best practices for knowledge modelling? How can knowledge about the SBTPs domain be made readily available to domain professionals?

Secondly there are limitations of the current web. The most comprehensive source of information on any domain is generated and presented on the most popular and widely used medium, the internet. The World Wide Web has recorded enormous success in electronic access to large repositories of information about numerous domains including sustainable building technologies. Hundreds of sustainable building technology manufacturers and suppliers use different websites to showcase their products. However, despite the ease of accessibility the web provides, it is increasingly challenging for users who wish to purchase and install sustainable technologies to traverse millions of irrelevant pages generated by the search engines. Large proportions of results may not be related to the user's needs. It is therefore difficult to acquire relevant information that would enable professionals and users to make informed decisions about these technologies. Furthermore, information is represented on the web using some non-machine-processable languages such as HTML, that the computer does not understand. Such languages focus on describing the pattern of data and they do not contain execution models. Computers cannot process data represented in these formats. Therefore, it is imperative to develop better ways of information representation on sustainable building technologies on the web, so construction professionals and users can easily obtain useful information with the aid of intelligent decision support tools. The research challenge here is what format should knowledge on SBTPs be represented on the web so it can not only be understood by humans but also by computers as well as to facilitate decision-making? What is the best approach to overcome the limitations of the current web? The solution to these information-related challenges may increase the adoption of sustainable technologies in practice.

To overcome these challenges, the use of a machine-processable language that provides semantics to electronic information is proposed (Gruber, 1993). The semantic web and its associated technologies improve information representation, inferencing, and interoperability. These have emerged to create a globally connected knowledge base of information presented in machine-understandable formats that enable software agents to perform complex tasks to provide solutions to users (Gruber 1993, Berners-Lee et al., 2001, Antoniou et al., 2012). One of the most fundamental techniques for representing domain knowledge in the context of the semantic web is through an ontology. When developed, an ontology provides well-defined formal syntax and semantics to domain knowledge and enables the implementation of decision-support tools and applications. However, to provide an ontology with formal semantic structures it is necessary to research and identify the best knowledge engineering frameworks that would facilitate the capture and specification of quality domain knowledge and represent it in modelling languages (Stevens et al, 2014). Furthermore, it facilitates the re-use or extension of existing domain ontologies based on standard taxonomies such as Industry Foundation Classes (IFC) (BuildingSmart, 2022a) to provide rich domain concepts of SBTPs. Since its emergence, the semantic web and its technologies have been used in several studies to provides solutions in a number of problem domains (Antoniou et al., 2012; Grzybek et al., 2014; Pauwels and Terkaj, 2016; Abanda, Foguem and Tah, 2017; Fernández-Izquierdo and García-Castro, 2021). A description of the problem domain for this study is provided in the next section.

1.3. Identification of Problem Domain

Due to the large number of existing SBTPs and the associated enormous volume of information, this research addresses sustainable building technologies with a particular focus on the Photovoltaic (PV) system problem domain. The PV system is believed to be one of the most popular sustainable building technologies where researchers and industry professionals have developed a tremendous interest in

(Al-Soud and Alsafasfeh, 2015; Energy Savings Trust, 2023). In the construction domain, the adoption of PV systems results in a significant reduction in carbon emissions thereby making it an appropriate choice of technology to meet the climate change targets set. PV's high energy-saving advantages and increase in adoption compared to other sustainable energy technologies affirms why it is a choice of focus in this study. IEA records high growth of PV use worldwide with a record 156 TWh in 2020, a 20% increase to reach 821 TWh (IEA, 2021). A growth in installations in the UK was noted from 2.8GW to 5GW in 2014, 8GW in 2015, 13.9GW in 2020, and a projection of 30TW by 2050 (Hartley, 2015, Whitlock, 2021). However, despite its popularity, its adoption still faces several challenges. Users face multiplicity in design and selection of PV technology due to variation in its components and their corresponding types, sizes, costs, and efficiencies. Furthermore, observations from research and practice have revealed that installed PV systems significantly underperform when poorly designed thereby generating inadequate electric output to the user as against what they have been designed to generate (Ma et al., 2018). The complexity of the permutations of PV components and unique features must be explored in order to recommend an appropriate approach and model to the design of PV systems. The intention is to develop a selection process that covers sustainable building technologies at a high level and then narrow the scope down to PV systems as a representative problem domain for detailed investigation. Knowledge gained can then be generally applicable to other domains.

The application of semantic web technologies in this study involves the development of an ontology to provide well-defined syntax and semantics to PV domain knowledge. Ontology is one of the most fundamental aspects of the semantic web because it transforms information from its unstructured format to a structured representation that is machine-processable (Patel and Jain, 2019). Several research problems must be overcome to avoid developing an isolated conceptualization of the PV domain. Primarily, the best knowledge engineering frameworks must be researched and identified to ensure complete knowledge acquisition of the domain. Another challenge that must be overcome is how to ensure domain knowledge is represented in standard terms by exploring the

potentials of re-using domain-specific online product repositories, taxonomies, and standards such as the Industry Foundation Classes (IFC). The compatibility of semantic web and IFC must be investigated. Furthermore, it is imperative to investigate how semantic web technologies can enable the implementation of decision-support tools and applications. Since the development of an ontology is a back-end application, this study also investigates the possibilities of further implementations based on the developed ontology through a graphical user interface as part of a web application. This decision is justified because while the ontology provides a formal standardised representation of knowledge, the semantic web application interfaces provide solutions to several challenges of the problem domain. The outcome of this investigation and implementation would be recommended to domain experts, manufacturers, and end-users for use in practice. The domain problems introduced above led to the establishment of the aim and objectives of this study.

1.4. Aim and Objectives

This study aims to investigate the extent to which semantic web technologies can be used for knowledge modelling to support informed decision making and recommendations on the selection of sustainable building technology products. The objectives are to:

- a. identify challenges associated with managing information and selection of sustainable building technology products on the web.
- b. develop a conceptual knowledge model of a representative sustainable building technology product problem domain.
- c. design and implement a prototype system that demonstrates the potential of semantic web technologies to provide decision support in the selected problem domain based on (b) above.

- d. evaluate the prototype system and make recommendations on the implementation of such systems in practice.

1.5. Summary of the Research Methodology

The application of semantic web technologies to investigate the extent to which it can aid with decision support in SBTPs cuts across three engineering stages: Knowledge Engineering, Ontology Engineering, and Software Engineering. It is for this reason that an exploratory qualitative approach that includes a selection of valued methodologies from the above-mentioned fields is believed most appropriate and selected for this study. A comprehensive investigation of the research problem established key findings. To suit the context of this research, evidence is provided (to support obtained findings) using substantial arguments and demonstration through a prototype. The use of these approaches to achieve the aim of the study is described against each objective.

Objective a was achieved through a critical review of SBTPs and emerging semantic web technologies. The exploratory review of SBTPs undertaken focused on renewable energy technologies with a particular emphasis on the PV systems domain. This led to the identification of state-of-the-art SBTPs and development of a selection process model based on the products' distinctive characteristics. The review identified challenges faced during uptake of SBTPs and the need to develop efficient ways of knowledge modelling. Similarly, state-of-the-art semantic web technologies were critically investigated to determine their expressiveness, reasoning and inference capabilities to overcome the limitations of the current web, how they can efficiently model knowledge on SBTPs and provide decision support and recommendation on the design and selection of SBTPs. Successively, existing domain knowledge bases were identified for reuse to ensure the development of a rich domain knowledge base. The research domain's theoretical background is established.

Objective b was conducted through an initial domain conceptualisation and a subsequent complete knowledge specification of SBTPs and PV domain. The thorough knowledge acquisition, specification, and refinement led to the establishment of the complete conceptual knowledge model of the domain.

Objective c was achieved by implementing the conceptual knowledge model developed in **(b)** above. A prototype system was used to conduct a proof-by-demonstration to verify and provide proof of the concepts established in the research.

Objective d was achieved by thorough evaluation of the developed prototype through semantic and syntactic verification. Validation was done using real-life data through a case study. Challenges experienced during prototype development were analysed and documented.

The successfully evaluated prototype that demonstrates the extent to which semantic web technologies can aid with decision support in the selected SBTPs led to the achievement of the aim of this study.

1.6. Research Achievements

This study focused on two things; to explore the domain of sustainable building technologies and to investigate the capabilities of semantic web technology in providing solutions to the challenges identified in the former. It is for this reason that effective knowledge modelling techniques and tools were studied to support knowledge modelling of SBTPs and their representation in an ontological format to provide decision support. This led to the following achievements:

A comprehensive conceptual model - of the sustainable building technology domain with a focus on photovoltaic systems which can be extended for further implementations.

Unique methodological framework - The creation of a semantic web-based methodological framework for knowledge modelling and representation of sustainable building technology and photovoltaic technology domain represented in machine compatible format suitable for use in application development.

Photovoltaic technology selection model - A process model and an algorithmic approach for sizing and selection of photovoltaic technology components involving multicriterial design parameters.

Extension of IFC - An integration of semantic web with IFC for efficient knowledge discovery about building domain knowledge. This led to the development of a modular ontology that extends the standard IfcOWL ontology to represent knowledge on sustainable building technology and photovoltaic systems through the IFC standard structure. The proposed ontology called IfcOWL_pvOntology could be used as an official extension of IFC ontology for the photovoltaic technology domain. This ontology could be adopted and approved by the buildingSMART working group in the W3C as a standard ontology for photovoltaic systems that can be applied for use in practice in the industry.

Semantic web application (prototype) - A semantic web-based navigable prototype system for exploring the integrated IFC and photovoltaic technology ontology named SIPS, meaning “**S**emantic web **IFC**-inspired **P**hotovoltaic **S**ystem” application for design and selection of PV systems. The prototype demonstrates decision support for photovoltaic technology products.

Published Articles - Articles published include:

An integration of ontology with IFC for efficient knowledge discovery in the construction domain (Usman, et al., 2018).

A process model developed based on the design of PV system components (Usman et al., 2020a).

A Critical Appraisal of PV-Systems' Performance, discussed concerns and performance challenges faced in installed PV systems (Usman, *et al.* 2020b).

Contemporary Applications of Semantic Web Technologies in the Construction Domain. (Usman, *et al.* In Press.)

1.7. Organisation of the Thesis

The thesis contains ten chapters. Some chapters are requisite to the development of the subsequent chapters. For easy comprehension, the structure and arrangement of the chapters are provided in figure 1.1 below.

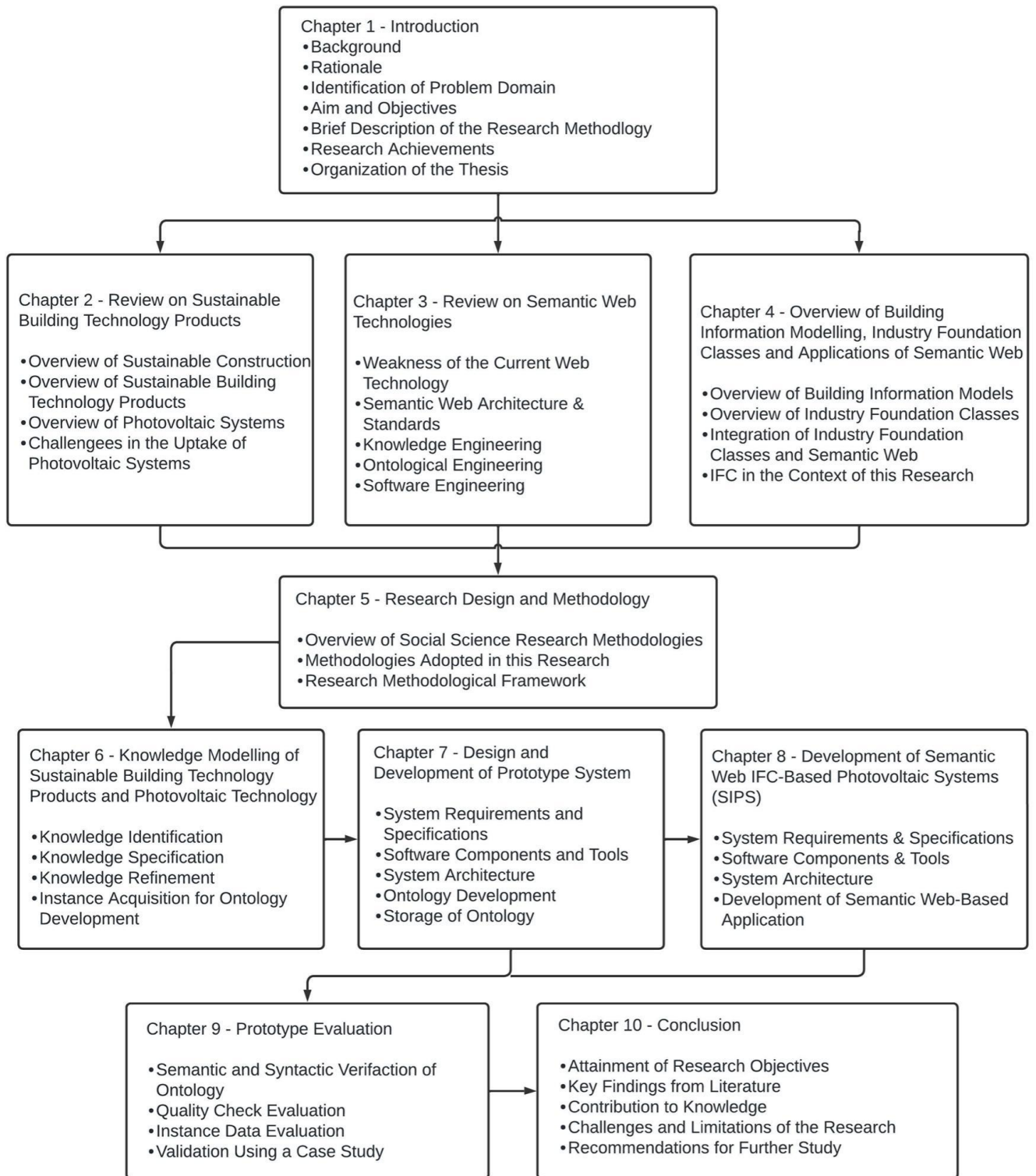


Figure 1.1: Structure of the Thesis

Chapter 1 presents a general overview of the research. It presents the background of the study and highlights the environmental impacts of construction. This led to

the establishment of the role of sustainable building technology products in sustainable construction and the challenges associated with their use. Furthermore, challenges of knowledge representation and use on the current web were presented. For that reason, the potential of emerging technologies such as the semantic web to represent knowledge on sustainable products are discussed. This is followed by the aim and objectives of the study. A brief discussion of the problem domain, the research methodology, and the main achievements of the study are presented.

Chapter 2 presents an overview of sustainable construction and sustainable building technology products with an emphasis on photovoltaic system technologies. This review resulted in the identification of three key knowledge gaps in the domain. These knowledge gaps include the impact of sustainable construction and sustainable building technology products on alleviating environment concerns, the selection and use of sustainable building technologies into existing and future construction projects, and the establishment of the significance of using state-of-the-art information technologies in modelling domain knowledge. This could potentially promote use of sustainable building technologies in the construction domain. This formed the rationale, aim, and objectives of this study. Furthermore, the need to apply state-of-the-art information technologies in modelling knowledge on sustainable building technologies led to the investigation of knowledge modelling techniques and semantic web technologies in chapter 3.

Chapter 3 provides a comprehensive exploratory review on knowledge modelling methodologies and the emerging semantic web technologies with their potential in providing solutions to the problem domain. This review resulted in the decision to have integrated/combined techniques of knowledge modelling knowledge engineering, ontology engineering, and software engineering approaches. Therefore, the chapter reviews these approaches, suitable methodologies, tools, and development languages that fit the context of this study. It was noted that the latest digital developments concerning sustainable building technologies involve Building Information Modelling (BIM) and domain standards

such as the Industry Foundation Classes (IFC) to support interoperability between the disparate applications used in the industry. Because the resulting prototype system in the study is recommended for use in practice, it was imperative to develop it using domain standard taxonomies to facilitate future use in practice. These standards are reviewed in chapter 4.

A review of BIM and IFC conducted in chapter 4 provides the rationale to use these standard domain taxonomies in the study. The chapter assessed the structures and forms of representation of building information represented by IFC. Furthermore, the different schemas and specifications that resulted in the different versions of the IFC standard are appraised. This led to the establishment of the need to integrate IFC and semantic web in developing domain applications. IFC's Model View Definition (MVD) suitable for the implementation in this study is identified and the choice was justified in this chapter. Based on the review on sustainable building technology in chapter 2, review on semantic web technologies in chapter 3, and review on BIM and IFC in chapter 4, it was imperative to design a unique methodology to extend IFC and develop an ontology to model knowledge on sustainable building technologies to address the knowledge gaps identified. This forms the basis for chapter 5. Because the study cuts across three engineering approaches: Knowledge Engineering, Ontology Engineering, and Software Engineering, the knowledge engineering methodology CommonKADS is adopted as the primary research framework and extended to include methodologies from ontology modelling, software engineering and exploratory case study approach. The research methodology has been presented in chapter 5. This methodology is implemented in chapters 6, 7, 8, and 9.

Chapter 6 presents the development of the conceptual knowledge model of the problem domain. This includes a group of concept representations under the Unified Modelling Language (UML). Concepts, properties, and their relationships have been identified from research sources, filling up knowledge gaps identified in the review chapters. This conceptual model is then transformed into equivalent Web Ontology Language (OWL) concepts to develop an ontology in chapter 7. In this chapter, the first half of the implementation for this study is conducted. This

chapter discusses the system architecture, and software components required to develop the ontology. One of the key challenges faced was how to effectively extend the standard IFC ontology called ifcOWL to contain knowledge on sustainable building and photovoltaic system technology products. The chapter presents how semantic web technologies can be used in knowledge modelling.

Chapter 8 presents the second half of the implementation which focuses on the development of a prototype system - a web-based application that provides recommendations on the choice of photovoltaic components to the user. This chapter used the knowledge represented in the ontology developed in chapter 7. Implementation done in this chapter involves several software components and languages. The justification for the choice of adopted components is provided. The development of this semantic web-based application is a proof of concept and confirmation of the rationale of the study. It is imperative to represent knowledge in machine-processable formats, so it can be read, processed, reasoned on, and used by computer applications to solve domain problems. The prototype system developed is evaluated in chapter 9. An evaluation chapter is vital to ensure that the developed prototype justifies the claims that emerged from the review chapters. It also ensures the methodologies presented in the methodology chapter are followed. Evaluation proves that the prototype is technically fit and correct, and performs the functions it has been developed to do. Syntactic and semantic verification was conducted. Similarly, because an exploratory research approach was adopted, the prototype is also validated using a real-life case study. Evaluation results are presented.

Chapter 10 concludes the thesis by providing a summary of the achievements of the research. It presents the contribution to knowledge, recommendations for use in practice, and the scope for further research. Challenges encountered throughout the research are also provided.

2. Review on Sustainable Building Technology Products

2.1. Introduction

This chapter provides an overview of sustainable building technology products with an emphasis on photovoltaic system technologies. The impacts of sustainable construction and the applications of sustainable building technology products in mitigating the continuous environment concerns on carbon emissions is discussed. The chapter analyses the design, selection and uptake of these products in construction projects. Based on the challenges that hinder the adoption of sustainable building technology products identified, the chapter signifies the importance and implication of using state-of-the-art information technologies in modelling domain knowledge. This led to the investigation of knowledge modelling techniques and semantic web technologies in chapter 3.

2.2. Sustainability and Sustainable Development

Sustainability, sustainable development and sustainable construction are interrelated terminologies and practices. "Sustainability" as a term has a wide range of multidisciplinary meanings, depending on the context and "what" resource or material is being sustained. In the dictionary, it is simply the ability for a material to endure and maintain itself. In academic literature, there are over 300 definitions of sustainability. Fedkin (2020), takes a stance that sustainability presumes that resources are finite and therefore should be improved, managed, and maintained, to avoid depletion and to remain available over a long period of time. In an attempt to study the various definitions of sustainability, researchers identified key components such as long-term, holistic approach, environmental justice, social claim (Freyman, 2012). Similarly, sustainable development has several definitions, but the most popular and frequently quoted definition is its definition by The World Commission on Environment and Development (WCED), popularly known as the Brundtland Commission. It defines sustainable development as development that "meets the needs of the present without

compromising the ability of future generations to meet their own needs' (WCED, 1987). However, one of the most useful definitions is the triple bottom-line approach which defines sustainable development in respect to the three P's - people (social justice), planet (environmental quality) and profits (economic prosperity). These three form the pillars of sustainable construction and establish the connection between sustainable development and sustainable construction.

2.2.1. Need for Sustainable Construction

Sustainable construction focuses on achieving sustainable development by applying environmental, economic, and human indicators of sustainability at all stages in a construction's lifecycle. Social sustainability focuses on "people first". It encourages systems and structures to support diverse, equitable, inclusive, and habitable communities (de Fine Licht and Folland, 2019). In terms of construction, social sustainability means the design and quality of buildings should promote healthy living and high quality of life (Missimer & Mesquita, 2022). Similarly, it could also mean safety and health measures in respect to injuries and accidents on construction sites. Likewise, the construction industry is labour-intensive and contributes to global employment rates. High-rates of employment lead to high standards of living while low employment rates lead to poor quality of life. This is also directly linked to economic sustainability. The role of the construction industry in national economic development and sustainability is vast. The construction industry contributes to 13% of global Gross Domestic Product (GDP) expected to rise to 14.7% by 2030. This is achieved through fixed assets and real estate and high employment rates (Khaertdinova, Maliashova, & Gadelshina, 2021). According to the construction industry employment statistics, in the UK alone, the industry is responsible for about 7% of all jobs (Crewit, 2022).

Environmental sustainability has the largest impact compared to the other two sustainability practices. From inception to demolition, all stages in a construction project's life cycle pose significant threats to the environment. The design, construction, maintenance, and demolition of buildings have significant negative impacts on the environment. It leads to depletion of natural resource bases,

causes chemical pollution, increases waste, and increases energy consumption (Thomas, Nair and Enserink, 2023). Construction industry is responsible for 35% of total greenhouse gas emissions (Liu, Pyplacz, Ermakova, & Konev, 2020), growing to over 36.8 Gt in 2022 (IEA, 2023a). Construction worldwide accounts for 17% of the world's freshwater withdrawals, 25% of its wood harvests, 40% of total CO₂ emissions, and 25% of ozone-layer depletion (Augenbroe, Pearce and Kibert, 2010; Sangmesh et al., 2023).

There are several other areas of key concern. Firstly, the manufacture of construction materials depletes natural resources and causes deforestation. Sangmesh *et al.*, (2023), affirm that a significant number of resources are wasted every year in construction, through excessive use of water and wood. This excessive use of wood has caused deforestation and hastening of global warming. Secondly, manufacturing and assembling processes of construction materials consume undue levels of energy. Both inherent chemical energy and embodied energy are released during production. Despite the increase in renewable energy utilisation in the world, buildings are still responsible for more than 40% of the total energy consumption (IEA, 2023b) to power buildings for heating, lighting, and ventilation purposes.

Ultimately, the foremost concerns are the excessive use of fossil fuels (to satisfy energy demands) and the increase of greenhouse gases (GHGs) in the atmosphere. For instance, in the United Kingdom (UK), carbon emissions generated directly from buildings (from the use of fossil fuels for heating) make up 17% of the total greenhouse gas emissions, while indirect building emissions (from electrical appliances) account for 23% of the total (Climate Change Committee, 2020). In 2022, global energy-related carbon emissions increased by 321 metric tons (IEA, 2023a), 40% of which originates from construction and buildings, while production of construction material accounts for 15% (Sangmesh et al.,, 2023). Furthermore, waste creation and mismanagement of construction products accounts for 33% of all waste globally (Thomas, Nair and Enserink, 2023).

Sustainability in construction therefore calls for the effective use of natural resources, minimisation of environmental degradation, waste, and greenhouse gas emissions, and optimisation of healthy built environments. Kibert (1994) reputedly defines sustainable construction as "creating a healthy built environment using efficient resources, minimising depletion of natural resources and enhancing an effective way of maintaining the environment". However, based on the discussions above, it can simply be construction practices based on the principles of sustainable development. Researchers emphasise the importance of adopting technology on sustainable development to reduce the negative impacts on the environment (Thomas, Nair and Enserink, 2023). The use of sustainable building technologies would fulfil the principles of sustainability by making construction affordable (economic), more suitable for human use (social) and to make large natural material and fossil fuel savings (environmental).

2.3. Sustainable Building Technology Products

The terminologies within Sustainable Building Technology Products (SBTP), will be broken down to facilitate a clear understanding of the definition. Sustainable building can be defined as a building that is efficient in terms of its use of energy, materials, its water conservation and its reduced impact on the environment. In other words, a building is sustainable if it is built and operated in a resource-efficient manner. The term "technology" is an application of knowledge for practical purposes, mainly to solve a specific problem in an industry. In this context, a technology does not only mean a device but goes beyond to include processes or practices aimed at providing solutions. Therefore, sustainable building technologies can be defined as heuristic fixes to the triple bottom line principles of sustainable development.

Different Sustainable Building Technology Products (SBTP) exhibit different degrees of sustainability. These technologies could be components for off-site manufacturing, water management technology and waste minimization. SBTPs

also include renewable energy technologies. As the name implies, water conservation technologies are vital in minimising the use and waste of water in buildings. These include recycling runoff water, rainwater harvesting and greywater recycling technologies (Stang *et al.*, 2021). It also includes regulating domestic water flow through efficient dishwashers, baths, and flush systems and several other water saving technologies (Lai *et al.*, 2023).

Waste minimization technologies are technologies used in eliminating waste related to construction and buildings. Waste creation and mismanagement of construction products accounts for 33% of all waste globally (Thomas, Nair and Enserink, 2023). Furthermore, construction and demolition waste also contribute to carbon emissions. These include emissions from machinery used in waste transportation, decomposition of waste, and the release of contaminants and pollutants into the landfill. In the UK, about 40% of landfill waste comes from building and construction waste (GOV.UK, 2021a), while in the United States, 15% of its total emissions comes from its landfill waste (Ecobin, 2021). It is for this reason that waste management techniques are promoted. These include prevention, minimisation, re-use, recycling and disposal. Tan, Cai and Li, (2022), studied the effects of construction demolition waste and developed an approach to recycling them based on geopolymers technology. Wang *et al.*, (2023) studied waste carbon fibre and their effects on construction materials and provided a review of state-of-the-art methods of recycling such waste in the construction industry. Further knowledge on waste reduction and cleaner technologies can be obtained from Hussain, Paulraj and Nuzhat, (2022).

Most smart home solutions are sensor based IoT applications that perform real time data prediction, monitoring and managing the building's performance (Lai *et al.*, 2023). Smart building systems fulfil social sustainability by improving the standard of living in homes. Similarly, they improve energy conservation for instance through light building sensors that operate only when the occupant is present thereby reducing energy costs (and attaining economic sustainability). From the environmental point of view, the use of smart building systems largely reduces the building-related CO₂ emissions. Patil, Boraste and Minde, (2022)

provide a comprehensive review of emerging smart building technologies and the use of internet of things (IoT) to collect construction data. In another study, researchers discussed the benefits of using emerging information and communication technologies in smart building technologies and provided an insight into their eventual applications (Zhao *et al.*, 2023).

The most beneficial SBTPs are renewable energy technologies because they generate energy from natural sources; that are clean, abundant, and free (Helal, 2023). A large number of rural communities especially in developing countries have no access to cheap grid connection. Households connected to the grid may want to contribute to their energy consumption to cut down their electricity bills. Furthermore excess electricity generated by renewable energy technologies can be sold into the grid (Okay, Eray and Eray, 2022); this is a payback on investment. An overview of renewable energy technologies is provided in the ensuing section.

2.4. Renewable Energy (RE) Technologies

Renewable energy technologies are increasingly used in sectors that are hard to decarbonize, especially the construction industry (Helal, 2023). The world's net-zero economy goal by 2050 has motivated the exponential increase in the adoption of renewable energy technologies globally (Airswift, 2022). It is broadly believed that the transition to a net carbon economy would continuously increase renewable energy storage requirements significantly imminently (Khan, Asfand, & Al-Ghamdi, 2023; IEA, 2022b; Kalay, Kılıç and Sağlam, 2022). In 2022, global energy-related carbon emissions increased by 321 metric tons (IEA, 2023a), 40% of which originates from construction and buildings (Sangmesh-B *et al.*, 2023). In the UK, the government has set its targets and minimum requirements that construction companies must abide by. In their recent sixth carbon budget, an ambitious target was set to reduce carbon emissions by 78% by 2035 and attain a net-zero emission by 2050 (GOV.UK, 2021b). Therefore it is essential to adopt renewable energy technologies in order to achieve this target. The most widely

adopted renewable energy technologies are wind, hydro, biomass, concentrated solar power, heat and photovoltaic technologies.

Wind turbines are renewable energy technologies in which turbines convert the kinetic energy of the wind and convert it into electric energy (Nejad *et al.*, 2022; Selin, 2023). Wind power is the renewable energy source. Zafar (2018), claims wind energy technology is the cheapest widely available renewable energy technology with minimal damaging effects on the environment. The vertical and horizontal axes of the turbines are forced to rotate by the wind's force, to drive an electrical generator. Naturally, the size of a wind turbine required depends on its projected energy purpose. The largest turbines with diameter of over 162 metres can produce about 4.8 - 9.5 megawatts of power while smaller turbines of about 40 metres blade length can produce up to 1.8 megawatts of power (Selin, 2023). These smaller turbines can be used to provide power to individual homes. A critical factor that determines the amount of energy wind turbines generate is the average wind speed, such that for this technology to be an option, wind speed must be above 5-6 metres per second (Selin, 2023).

Another important factor to consider when adopting wind turbines is the swept area, the more extensive the diameter of the swept area, the larger the energy output (Jenkins, 2015). Wind turbines' best locations are areas exposed to strong, sufficient, and frequent winds without significant turbulence and obstructions from tall buildings. These areas are not necessarily near the metropolises where the electricity is required. Thus, transmission lines have to be built to transfer the electricity (Selin, 2023). Potential wind sites can take several years of monitoring to determine whether it is suitable or not. Wind speed also increases with height; therefore, regions near hilltops or coasts are perfect. Nonetheless, wind energy does not provide all the electricity a building requires; it only provides extra energy causing an incessant dependence on the grid and fossil fuels. There is also the risk of loss of blade; people and animals could be subjected to harm. Further details on wind turbines, their designs, generators, power converters, operation and maintenance can be found in Nejad *et al.*, (2022).

Another renewable energy technology is Biomass Generators. Biomass is defined as “any plant matter used directly as fuel or converted into other forms before combustion” (OECD, 2023). Electricity and heat are generated from organic combustion of a wide range of plant-based materials, wood species, woody waste, construction or demolition wastes, and cellulose as alternatives to fossil fuels. They are also obtained from vegetable oil crops, crop wastes, and palm oils for liquid energy and animal residues to make gas (Nehal, 2023). According to the U.S. energy information administration energy reviews (EIA, 2023), the amount of CO₂ biomass energy sources release when burnt is precisely the amount of CO₂ absorbed during the biofuels' growth. Thus, it is carbon-neutral. Because biomass raw materials can be found everywhere, it reduces residues and wastes that naturally rot and generate CO₂ and methane; hence they are better used as fuel. They are available all year round and can complement intermittent technologies such as solar and solar thermal energy. However, this may compete with actual food production and may become a matter of ethics and values. Is it appropriate to use crops as biofuels when people are hungry worldwide?

There are few challenges faced that hinder the swift adoption of biomass generators. These are high costs of operation and maintenance, policy changes, supply chain and bioenergy conversions (Helal, 2023). Other issues include land space to grow and store biofuels, proximity to alternative biofuel locations and transportation, and water availability for biomass plant cooling purposes. Several researchers have reviewed important aspects of biomass energy generation including design and modelling of biomass supply chains for biofuel generation (Martinez-Valencia et al., 2021), feasibility evaluation of bio-based industries with emphasis on the effects of supply chain (Lo et al., 2021), and biogas production, costs, and efficiency (Mottaghi, Bairamzadeh, and Pishyae, 2022). An exhaustive review of literature published is done on sustainable biogas optimization approaches, biomass-to-bioenergy supply chain research (Helal, 2023).

One of the oldest sources of renewable energy is hydropower. Hydropower converts the potential energy of a water source (stored at a height) to kinetic

energy as it is released on a hydropower technology (made up of massive turbines and electricity generators) (IHA, 2022). The gravitational energy from the water dams, together with the moving turbines' mechanical energy, generate electrical energy. This technology entails turbines, bearings, waterways, dams, ocean power, sealing (Quaranta and Davies, 2022). Hydroelectricity can be produced 24/7 in high demands, while dams/water can be closed and conserved when not in use. Hydroelectricity has been a reliable source of renewable energy worldwide with an increase of installed capacity to 1360 GW in 2021 and an estimated 3,600 GW by the end of 2023 (IHA, 2022). The international hydropower association (IHA, 2020) presents emerging hydropower technology trends that entail digitalization, flexibility, storage and novel scale of the technology. In Quaranta and Davies (2022), researchers discuss emerging and innovative materials used in hydropower technologies and their applications and provide emphasis on their performance, advantages and challenges. However, despite the state-of-the-art trends, the international hydropower association reports that the technology is falling short of the progress much needed (IHA, 2022).

Hydroelectric power plants are costly to build with a considerable capital cost and only possible in areas with huge volumes of water. The process of building a dam may create floods causing threats to aquatic life and the natural environment. When a dam is built in one area, the people down the river lose control over the water flow. This may cause controversies. An example is the Lagdo dam in Cameroon (Ma et al., 2015). The river Benue used to be a means of transportation and economic activities in northern Nigeria, but it is dry today. Likewise, when the dam has excess water, the water is released and flows into Nigeria, causing massive floods displacing tens of thousands of people every year. Furthermore, this renewable energy technology is adopted at a community or national level and not feasible for single household needs.

The renewable energy technologies most distinguished for household adoption and also practical for community usage are technologies that use solar energy from the sun for heating and electrical purposes for buildings. The potential of solar energy technologies exceeds the other renewable energy technologies due

to its inexhaustible source, decentralised reach (in terms of location) and its ability to transform solar energy into heat and electricity without the emission of CO₂ (Răboacă *et al.*, 2019). The key solar energy technologies are photovoltaics (PV), which converts the sun's light (solar energy) into electricity; concentrated solar power (CSP), which utilises the sun's heat (thermal energy) using concentrating collectors to run electric turbines and power cycles (Airswift, 2022; Khan, Asfand, & Al-Ghamdi, 2023); and solar heating and cooling (SHC) technologies, which uses thermal energy to provide heating and cooling solutions to residential, commercial and industrial buildings (Sezen and Gungor, 2023; SEIA, 2023).

In a review of literature, comparisons are done on the advantages of these technologies. Researchers give credence to CSP claiming that unlike PV technologies, CSP has a large capacity to store heat energy for long intervals for conversion into electricity at a later time (Răboacă *et al.*, 2019). Furthermore, CSP can produce electricity after sunset or when clouds block the sun's rays (Khan, Asfand, & Al-Ghamdi, 2023). An installation of 1 square collector generates about 400kWh of electricity per year with about 12 tons of CO₂ reduction (Răboacă *et al.*, 2019; Khan, Asfand, & Al-Ghamdi, 2023). However solar technology companies are convinced that although CSP collectors are run by a clean and renewable source, its high cost and complexity of installation hinders its uptake (Airswift, 2022).

The solar energy industries association has confidence that solar heating and cooling (SHC) technologies can completely displace the use of fossil fuels for electricity (SEIA, 2023). As the demand for air-conditioning increases due to global warming, an important trend to look out for in 2023 is emergence of more solar cooling technologies. Similarly, with rising energy prices, gas water heaters will soon be phased out and replaced with solar water heating technologies. For this reason, SHCs have become a very popular research topic where researchers study and deliberate on the effects of solar radiation and ambient temperature on the performance of the technologies (Zhou *et al.*, 2020; Li and Huang, 2022; Sezen and Gungor, 2023). Users can overlook the high costs of installation and operation

since SHC technologies provide a return on investment within 3 to 6 years (SEIA, 2023).

A renewable energy technology that undoubtedly stands out is the Photovoltaic (PV) technology. With PV, homeowners can have complete independence from the grid and fossil fuel generated electricity because it can be used on a small scale without a considerable increase in cost. Hirth (2015a) states, "...In contrast, coal, hydro, and wind power plants feature significant economies of scale, such that they cannot efficiently be deployed in household size" making PV the most widely adopted renewable energy technology. Studies related to renewable energy stipulate that solar PV contribution to net-zero electricity may be more than 50% by 2050 (Sharma and Sengar, 2022). PV is also the most pleasing and attractive renewable energy technology to NGOs and the most widely supported by the government who gives massive subsidies and incentives to help the needy people while reducing carbon emissions (EnergySavingsTrust, 2015). Because of the enormous scope of sustainable technologies and volume of renewable energy technologies, this study narrows down and focuses on PV technologies with prospects to expand on in further studies in future research. An overview of PV technology is provided in the ensuing section with robust reasons for its selection as the problem domain in this study.

2.5. Overview of Photovoltaic System Technology

Photovoltaic (PV) technology converts solar energy directly from the sun to electricity through an arrangement of its components that include the semiconductor PV panels, charge controllers, solar inverter, batteries and cables (Silva et al., 2019). Researchers and industrialists have generated tremendous interest in PV-systems and their design. They are widely recognized for numerous reasons. Primarily, most habitual places worldwide have enough solar radiation; therefore, solar energy is an easy choice for individuals who wish to use sustainable energy technologies (Kalay, Kılıç and Sağlam, 2022). Similarly, it is widely applicable globally. Residents can install a few PV panels to reduce their electricity bill or a large installation to completely offset their daily demands or benefit from massive PV farms that provide the community with carbon-free electricity (SEIA, 2023). PV is suitable for either centralised or distributed power generation, designed from a few watts to power small applications such as calculators, bus shelter lighting, road signs, fence electrification, to thousands of kilowatts for homes, businesses and industries.

According to the renewable energy market forecast by the International Energy Agency Affairs (IEA), the world's renewable electric power capacity will rise by 1200 gigawatts by 2024 due to reductions in cost and PV-systems account for 60% of this increase (IEA, 2019). Another survey shows that PV-systems can provide sufficient energy capacity worldwide and exceed all other sources of energy by the next decade (Donev et al., 2018; (Sharma and Sengar, 2022). Likewise, they are modular and can be easily expanded if power demand increases in the future. Nonetheless, they have caused a high reduction in greenhouse gases. A study by the World Energy Council shows that 1kW of electricity installed in a home reduces its carbon footprint by over 3000 pounds annually (Aresolar, 2023). The International Energy Agency (IEA) reported that in 2022, 231 gigawatts of PV were installed globally making a total of 1.2 terawatts of electricity (NREL, 2023). PV systems have a prediction to quintuple the generation of electric energy until 2040.

A scholar outlined a notable conclusion that renewable energy technologies must be cheaper than non-renewable energy technologies, "...we can make as many projects as

we want, but decisions will be made against them." (Sustainable Sources, 2015). It is common to choose the option that is cheapest at first glance. This is a reason why PV could be selected over hydro technology that is capital intensive. In support of that, researchers (Bazilian et al., 2013; Hernandez and Martinez, 2013) discuss more causes of solar cost reduction. Hirth (2015a) proposes a 3-utilisation strategy for PV; firstly, merge solar-generated electricity with the grid electricity, secondly, reduce the cost and finally expand the production of PV technologies. Moore's law of PV implies that PV can achieve substantial growth rates (Nelson, 2010). Thus, the cost of installing PV increases to drop (Hartley, 2015). Although the initial cost of PV is high, its low cost of operation and maintenance outweighs the fuel generator's low initial cost and its high cost of operation. This makes PV the most cost-effective in the long run (Stand-Alone PV, 2015). Similarly, Pra et al. (2015) prove the energy required in producing a PV module compared to the energy it generates throughout its lifetime is insignificant.

An unfavourable element of PV technology is its intermittent nature (Gasparin *et al.*, 2022). It requires a sufficient amount of solar radiation to function, and so it may not generate electricity during winter or on cloudy days. A simple solution is to combine different renewable energy sources to form a hybrid system that provides electricity all year round. Although all renewable energy sources have complementary sources, PV has the largest complements. However, these complement energy sources must be selected depending on the building's geographical location and load capacity (Tudu et al., 2014). Scholars (Bansal et al., 2014; and Rahman et al., 2014) argue the hybrid system is most reliable and cost-effective compared to when a PV technology is used alone. Several researchers agree and have presented different techniques to optimise the PV hybrid energy system. Sanchez et al. (2010) analysed a combination of PV-wind systems. Liu et al. (2011) conducted a feasibility study for PV-wind-biomass hybrid system by while Bilal et al. (2010), Koutroulis and Kolokotsa (2010), and Fadaee and Radzi (2012) proposed a PV-wind-battery hybrid system. PV-wind-hydro combination is appropriate for electricity in remote areas (Bhandari et al., 2014), while the use of photovoltaic/solar thermal (PVT) systems is beneficial for large-scale heating purposes (Gulhane and Chavan, 2014). Rahman et al. (2014) show there can be no limit to multiple combinations. He proposed a Hydro-PV-Wind-Battery-Diesel based hybrid power system at a geographical location where there is a good potential for all three energy

sources. Khan et al., (2014) also created an optimal combination of PV, wind, micro-hydro, and diesel systems to satisfy the energy load. When renewable energy technologies cannot meet the load demand, one can use a diesel generator as a back-up (Barbose et al., 2013). The hybrid system can meet high electricity demands enough for a large community (Al-Soud and Alsafasfeh, 2015). However, it is complicated as interactions between the PV arrays, inverters, batteries, generators, and the complementary energy technologies must be controlled. The system becomes complex.

There are colossal PV technologies in the market, each with unique characteristics. Many manufacturers own and use thousands of PV technology products, and a lot more emerge. A market research firm (TPT, 2010) made available reports describing over 250 PV technologies that use different semiconductor materials with different designs and components. The components also have distinctive sizes, efficiencies, and prices. They also range from a few kilowatts to hundreds and thousands of megawatts. Therefore, this different characteristic of photovoltaic systems makes it typically challenging for individuals who wish to purchase and install PV to know precisely the type of PV technology they require. It is challenging to select from large logs the photovoltaic system type with the exact requirements needed by the user. Intended users need to know this unique and diverse state-of-the-art technology product to make the best choices. Careful planning and selection are essential.

2.5.1. Components of the Photovoltaic System

Several studies (Salim and Abu Dabous, 2023) have shown that the PV system domain is knowledge intensive domain. Therefore, for the context of this study, the scope is narrowed to focus on the major components of the PV-systems, the characteristics that differentiate them, the guidelines for their application, design methods and installations.

PV-systems generate electrical energy through an arrangement of a set of components. These begin with photovoltaic cells that are assembled to form solar modules (Marion, 2015; Schachinger, 2019; Silva et al., 2019). Most PV-systems installed in residential buildings need an inverter to convert the DC electricity generated by solar modules into

alternating (AC) current for use by the building (Ali, 2018). Additionally, most residential PV systems require batteries to store backup or excess electric energy. Typically, batteries should hold an adequate amount of electric energy for use during non-sunny hours and at night. Batteries require charge controllers to regulate the electrical current and voltage that goes through to the system. Additionally, they prevent batteries from excessive charging and discharging. Generators are used to connect the PV system to other energy sources when required. Finally, cables and wires ensure that all elements of the system are connected, thus providing paths for electricity flow throughout the PV-system. The following section describes the major components of PV components.

Modules - Modules are made up of photovoltaic solar cells containing two layers: the top silicon and lower boron layers (Barbose et al., 2013; Lee, 2015). These layers capture solar energy and convert it to electric energy (Aresolar, 2023). Each PV cell gives a small amount of electricity; therefore, they are assembled and connected using frames to form solar modules in standard sizes such as 36-cell, 60-cell, and 72-cell modules (Marion, 2015; Schachinger, 2019; Silva et al., 2019). PV-system modules can be found in different forms of materials, sizes, efficiencies, and costs. Each comes with its unique watt rating (given by the manufacturer of the module). Despite their difference, they all perform the same task of converting solar energy to direct current (DC) electrical energy. A grouping of these modules builds a solar array of a desired electric output. Figure 2.1 presents a typical PV-array field showing the standard terms of some of the components.

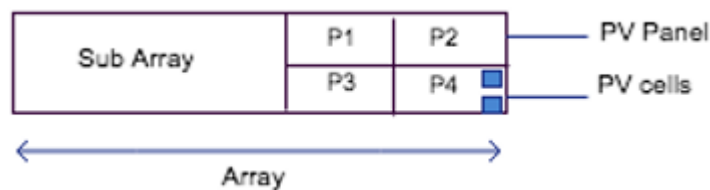


Figure 2.1. A Photovoltaic Array Field.

There are different types of module materials. The most common are monocrystalline, polycrystalline, and thin film. Monocrystalline modules are made from a single thin slice of pure, smooth and even silicon. They are the most efficient and most effective modules. As the name implies, polycrystalline modules are made from several layers/crystals of

silicon. They are less efficient than monocrystalline modules and therefore less expensive. Thin films are made by depositing thin layers (films) of photovoltaic material on a substrate such as a plastic, metal or glass. The next accepted module type is amorphous module. Amorphous cells are a type of thin-film cells (EnergyInformative, 2019a). Most small appliances that use PV-cells such as calculators are powered by thin film made of amorphous cells. Other types of thin-film module types are composed of cadmium telluride (CdTe) and copper indium diselenide (CIS) (EnergyInformative, 2019b). However, these technologies have high toxin substances that require cautious handling during manufacturing and disposal (Seme et al., 2019). Other types of PV-system modules include gallium arsenide and multi-junction cells which are less common due to high cost. Additional module materials include Perovskite cells, organic solar cells, dye-sensitised solar cells, and quantum dots; however, they are new and not fully explored (Donev et al., 2018). Three (3) important factors that make each module type suitable for different electric energy outputs are efficiency of cells, surface area, and costs (Shafique, Luo and Zuo, 2020). Table 2.1 below provides the difference between the dominant module types.

Table 2.1: Different Types of Photovoltaic Systems Modules

Module Type	Description	Efficiency	Advantages	Disadvantages
Monocrystalline	<ul style="list-style-type: none"> · They are made from a single, uniform, pure thin silicon crystal (Notton, Lazaro and Stoyanov, 2009) · Module has a distinctive hexagonal shape and a dark crystal look (Donev et al., 2018) 	20-25%	<ul style="list-style-type: none"> · High efficiency rate · High power output · Highly suitable for commercial uses · Less affected by high temperatures · Space efficient · High lifespan 	<ul style="list-style-type: none"> · Expensive · Manufacturing process is slow · It is labour intensive
Polycrystalline	<ul style="list-style-type: none"> · They are made from multiple grains of silicon crystals (Notton, Lazaro and Stoyanov, 2009) · They have a bright blue look, with distinct squares. · Polycrystalline modules are the most dominant PV modules in the market. They make up 70% of the world's PV modules. 	15-20%	<ul style="list-style-type: none"> · Less expensive 	<ul style="list-style-type: none"> · Less efficient · Sensitive to high temperatures · Shorter lifespan · Not space efficient

Thin Film	<ul style="list-style-type: none"> · These modules are produced by layering thin layers of PV material on a substrate such as glass, plastic, or metal (Donev et al., 2018). · The final product is very flexible and contains less than 1% of the silicon used in making crystalline modules (Al-Ghussain et al., 2023) 	7-10%	<ul style="list-style-type: none"> · Flexible · Easy to produce · Much lower price due to economies of scale · Less affected by high temperatures 	<ul style="list-style-type: none"> · Much less efficient · They take up a lot of space · These cells experience about 20% drop in efficiency in the first few months of installation · Much shorter lifespan
Concentrated PV cells	<ul style="list-style-type: none"> · It uses curved lenses to focus solar energy on its cells · It uses solar trackers to increase efficiency. · They use direct normal irradiance (DNI) 	Up to 40%	<ul style="list-style-type: none"> · High Efficiency under DNI. · Small array size · Low cost of manufacturing · Less affected by high temperatures 	<ul style="list-style-type: none"> · Far less common than the conventional cell types; it is not standardised. · Power output could be affected by shifts in radiation

Inverter - Most residential PV systems require an inverter to convert electricity from DC to AC (Malek, 2014). The details of the required inverter most often depends on the module types and array size selected for the system. The inverter consists of several electronic components, connectors and transformers. However, a comprehensive review of these sub-components goes beyond the scope of this study.

Battery - Most stand-alone PV systems require the use of a battery to store back-up energy. Detailed descriptions of the different types of batteries suitable for PV-systems are provided in the design subsection below.

Charge Controller - As the name implies, charge controllers keep batteries within a suitable limit of charge and prevent it from excessive charging and discharging. They help to prolong the lifespan of batteries in a PV-system. Controllers can control switching offloads, and alert users to do specific actions such as turning on a standby power supply (Stand-Alone PV, 2015). For long-lasting connections, single-conductor wires are used to connect PV-system panels.

2.5.2. Types of Connections

A PV-system can be a stand-alone, grid-connected, or hybrid system. Stand-alone systems are often used to power buildings in isolated areas far from grid electricity (Ali,

2018). They operate independent of the grid and require a battery to store their excess electricity. On the contrary, grid-connected PV-systems may not require the use of batteries. They are connected to the grid through an inverter. If the PV system generates more electricity than the building requires, the excess electricity can be sold to the grid (Okay, Eray and Eray, 2022). This is a huge return on investment. A single challenge of grid connected PV systems is its compatibility with the grid utility because each generates electricity with different frequencies (Khosrojerdi, Gagnon, & Valverde, 2021). Governments have conditions that must be met in order for one to be eligible to sell their extra electricity to the grid. For instance, the United Kingdom government contains detailed technical guidelines, guarantees, and legal demands that regulate transmission of solar electricity to the grid. Figures 2.2 and 2.3 show the typical connections for stand-alone and grid-connected PV.

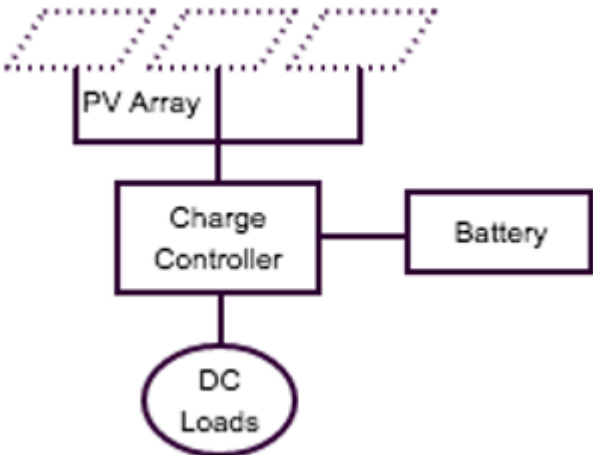


Figure 2.2: Stand-alone PV System Connection

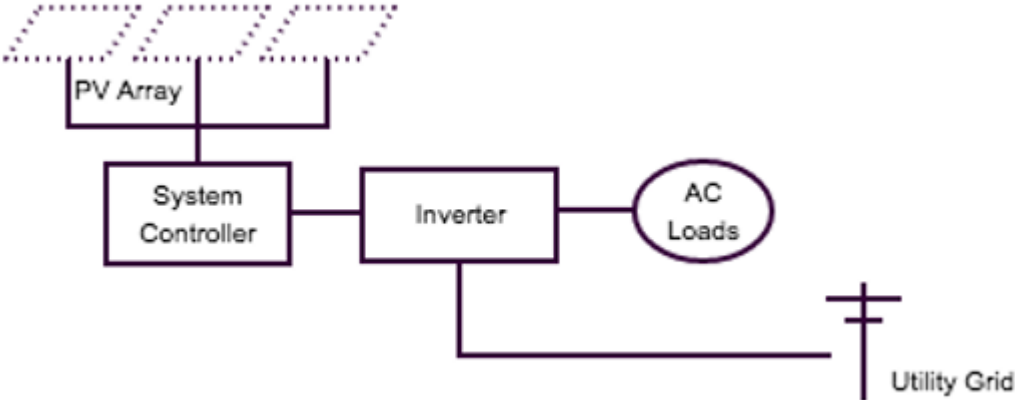


Figure 2.3: Grid-Connected PV System

PV hybrid systems combine a photovoltaic generator with other power sources—typically a diesel generator, but occasionally with another renewable technology such as a wind turbine (QuetteDiaf et al., 2007). The PV-system generator would usually be sized to meet the base load demand. This type of connection is the most prevalent in countries (e.g., Nigeria) where the grid electricity is insufficient.

2.5.3. Methods Involved in Photovoltaic Systems (PVS) Design

Designing a PV system is quite complex. Several conditions and compositions must be analysed. The first step in PV system design is to investigate the meteorological conditions of the intended location, followed by the determination of the energy load of the building. The size of the PV components are then determined based on the energy output needed. Once all the components are selected and sized, the design is examined to ensure that the negative conditions that affect performance are eliminated.

2.5.3.1. Geolocation and Solar Irradiance

For an effective design of a PV-system, the first factor to consider is the location of the proposed building where the PV-system would be installed. Because of the PV technology's intermittent nature (Gasparin *et al.*, 2022), locations close to the equator and other areas with high solar radiation are the best for PV adoption. The higher the solar insolation, the better the PV system works (Lee, 2015). However, PV does not operate well at very high temperatures. During very high-temperature days, PV components get overheated, which affects performance drastically (Yolcan and Kose, 2023). For that, adequate airflow around the panels is necessary. Al-Ghussain *et al.*, (2023) confirm that PV systems in locations with high solar radiations with enough airflow generate the highest electricity output. This is because the performance of PV-system components depends on the solar irradiance and temperature (Yolcan and Kose, 2023).

Every geographical location has unique values for longitude and attitude. Furthermore, the irradiation for each location comes in three (3) unique parameters (Dutta, 2019). These are global horizontal irradiance (GHI), diffuse horizontal irradiance (DHI), and direct normal irradiance (DNI). Table 2.2 below presents the descriptions of each of these parameters. These components, together with the plane of array (POA) and the ambient temperature of any given location, are vital to solar energy conversion technologies. POA is a combination of direct solar radiation, diffuse radiation, and radiation reflected by the ground (Kurtz et al., 2013).

In mathematical terms,
 $GHI = DNI \times \cos(\theta) + DHI$,
 $POA = I \cos\Theta$,
 where I = direct normal irradiance;
 Θ = angle of inclination (David, 2013).

Table 2.2 Difference between DNI, DHI, and GHI.

Names	Description	Instrument
Direct Normal Irradiance (DNI)	This gives the value of solar radiation coming directly from the sun.	Pyrheliometer. (David, 2013)
Diffuse Horizontal Irradiance (DHI)	This is the amount of solar radiation reflected by the ground (scattered in the sky) very close to the horizontal surface of absorption.	Shaded pyranometer. (Hay, 1979)
Global Horizontal Irradiance (GHI)	This is the geometric total amount of the solar radiation received in each dimension on the ground.	Shaded pyranometer (Hay, 1979)

Most PV-system performance models use either GHI or POA irradiance and ambient temperatures as input factors. However, these parameters are rarely used in the design of small-scale PV-systems because their measurement instruments are expensive (Dutta, 2019). Therefore, the accuracy of irradiance values for small-scale PV-systems rely on general calculations made for POA and ambient temperature in any given location (Gasparin *et al.*, 2022). Mathematical models are available in the literature for POA

(David, 2013). While the ambient temperature is the normal temperature around the PV-system modules at the time of solar radiation, for a standard operating condition (SOC):

$$T_A = T_C - (NOCT - 20^\circ\text{C}),$$

where T_A is ambient temperature, T_C is the module temperature, and NOCT is the nominal operating cell temperature given by the PV module manufacturer (Ayompe et al., 2010). Comprehensive mathematical models for ambient temperature can be found in (Rupnik and Westbrook, 2014) and are adopted by satellite-derived values. In literature, different design models were analysed to estimate the solar irradiance for a PV-system resulting in dissimilar values for the energy yield (Da-Silva, 2019; Gasparin *et al.*, 2022). Researchers have argued that irradiance estimation errors are caused by the parameters available for solar irradiance measurements (Bharadwaj and John, 2019; Al-Ghussain *et al.*, 2023). The choice of the measurement parameter used can make a difference of a few percent (Kurtz et al., 2013). Further studies on the difference between the two methods of irradiance calculations, including the benefits and drawbacks of each, can be found in (Ali et al., 2018; Dutta, 2019; Hay, 1979; Gasparin *et al.*, 2022). The predetermined solar irradiance values of any location in the world can be found with NASA meteorological (NASA, 2019), Meteonorm (Meteotest, 2019) solcast (Engener, 2019), Total Solar Irradiance (TSI) Composite Database (NOAA, 2019), Sorce (Sorce, 2019) Photovoltaic Geographical Information System (PVGIS) (Commission, 2019), etc.

2.5.3.2. Determining the Energy Load of the Building

The size of a PV-system is primarily dependent on the energy requirement of the user. This energy load can be acquired in different forms. The primary form of obtaining the energy load of a building is by reading the electricity bill provided by the electricity supplier in kilowatt-hours (kWh). Occasionally, a building may be charged a standing (fixed) amount per day regardless of how much energy is consumed. However, the most common method is to be charged per unit rate depending on the amount of energy consumed. Individuals who may not comprehend the information provided in electricity

bills may only note the amount due to be paid. In such situations, one could use the unit rate that is charged by the electricity provider to calculate the energy consumption of the building. For example, in the United Kingdom, although the price per unit rate may vary depending on the region and the type of energy plan subscribed to, the average unit rate of electricity is 14.37 p/kWh (pence/kilowatt-hour) (UKPower, 2020). Where this average unit rate applies and a monthly electricity bill of £70 is charged, the energy load of the building could be obtained through a simple equation where $\text{electricityBillAmount} = \text{unitRate} \times \text{energyLoad}$. The result is 487.3 kWh of electricity consumed.

Another method is to obtain a cumulative sum of the energy consumed by all the electrical appliances used in the building (Alsadi and Khatib, 2018). Electrical ratings, such as wattage, current, voltage, frequency, etc., are normally clearly written on labels on all appliances. However, if the wattage is not provided, it can be easily calculated using its current and voltage such that

Active or Real Power (W) = Current (I) × Voltage (V), ... (i)

Electrical Energy (consumed by the appliance) = Active or Real Power (W) × Total time the appliance is in use (hours).(ii)

Once the energy consumed by each appliance is obtained, a summation of all the values gives the electrical energy load of the building. An energy load guide is provided as an appendix at the end of the thesis.

Other approaches for determining the energy load of the building are available in the literature. Menezes et al., (2014) created two (2) models for estimating energy consumption through sampling of monitored data and a “bottom-up” approach that predicts energy demand. Results from validation of both models gave similar values with metered data. Researchers proved that it is an improved approach when compared to common practices discussed above; however, their models are only suitable for appliances with small wattage or power ratings. Similarly, average system loss of components is estimated to be 30%; therefore, 130% of the calculated building load is used in that study for the system design. In another study (Zheng et al., 2017), the researchers developed a model that generates heating and cooling profiles for a building

using real-life values obtained from monitoring mean temperatures outside versus the corresponding electricity consumption inside the building. The model was applied on case studies and proved accurate in determining energy loads of the buildings. However, it is imperative to note there can be variable energy demand (peak, average, and minimum energy load) depending on the time of day or the seasons. During the design process, when the peak load is adopted, this means that the PV-system will be large and expensive. Likewise, the energy generated will be wasted during the average and minimum periods. On the contrary, if the minimum load is adopted, then users may require alternative sources of energy to power appliances during the average and maximum periods.

2.5.3.3. Sizing and Selecting Components of Photovoltaic Systems

The overall design of the PV system depends on the sizes selected for each component. They are designed to operate together and evaluated in respect to one another so that the overall PV system obtains the desired energy output. When the components' sizing is well-optimised, optimum performance of the PV system can be guaranteed. The subsequent sections review how these components are sized as part of a PV system design.

2.5.3.3.1. Modules

Four vital factors must be explored to help select the most suitable PV module that would generate the peak output of electricity required by the customer. These are types of modules, the surface area of modules, the cells' efficiency, exposure to solar radiation (location), and building load (Stand-Alone PV, 2015; Zulkifli, 2014). Researchers (Al-Ghussain *et al.*, 2023) argue that when the building roof is at an optimum angle, the PV system generates a peak output of energy regardless of the efficiency of cells or surface area. However, Jenkins (2015) asserts that the type of module and geographical location are the most important factors to consider. Others researchers agree since the ambient temperature, solar radiation and wind speed largely affect the PV module cell temperature; and the temperature is directly related to the instantaneous efficiency of the modules (Yolcan and Kose, 2023). An important point to make is that a module of

higher efficiency generates more electrical energy output than a similarly sized module of a lower efficiency. Comparably, modules of higher efficiency require lower amounts of mounting surface area than modules of lower efficiency. However, the higher the efficiency, the more expensive the technology. An important question to ask in selecting modules is what matters most to the user. A PV-system owner with a large mounting space may select a less efficient module type to cut down price, while another person may prefer the most efficient module type if they have a small mounting space. Price, space, and efficiency trade-offs are unavoidable when selecting modules. Conversely, surface area constraint is mandatory, such that $\text{mountingSurfaceArea} \geq \text{arraySurfaceArea}$.

Once a module type has been selected, the total number of the modules needed to collectively generate the required capacity of electric output gives the array size. Several researchers and experts have different ways of calculating the module's peak output and array size (Leonics, 2013; Zeman, 2015; Stand-Alone PV, 2015). Some studies calculate the array size as a ratio of total watt or power rating required, to the rated power of the module type selected. In another study, the total power required is increased by 30% to cover for energy losses (from inverters, batteries and wiring) in the system. Furthermore, a varying factor known as the Panel Generation Factor (PGF) is considered. PGF helps to determine the energy output after losses due to the temperature, dirt, poor irradiance, and ageing of panels (Abdeljalil and Omer, 2023). A similar calculation is available in (Ali, 2018). In (Notton, Lazaro and Stoyanov, 2009), models were developed to calculate module efficiency and maximum power, where:

$$\eta_{pv} = \eta_{ref} \{1 - a[(GB)/18] + TA - 20\},$$

where η_{pv} is the module efficiency, η_{ref} is the reference efficiency, a - is the power correction factor determined by the study, GB is the solar irradiance, and TA is the ambient temperature. Similarly, the maximum PV power is determined by:

$$W_p = \eta_{pv}GBA,$$

where η_{pv} is the module efficiency, GB is the solar irradiance, and A is the PV module surface area (Notton, Lazaro and Stoyanov, 2009). A comprehensive guide for array sizing can be found in (IEEE, 2020). Solar energy engineers (Sustainable Sources, 2015;

Stand-Alone PV, 2015) say when more PV-modules (than required) are installed, the PV system performance will increase, and battery life will be enhanced. However, when fewer PV-modules (than required) are installed, the PV system would not perform when the weather is cloudy. As soon as the array size is determined, the design and size of the inverter can be determined.

2.5.3.4. Inverter

Inverters are selected and sized based on the total energy output expected from the PV-system. They are often selected based on their operational power and size. A suitable PV-system inverter should be sizable enough to hold the total energy load of the building at any point in time. Researchers analysed how and if variable energy demands can affect inverter choice (Schweighofer *et al.*, 2023). They proposed an algorithm to balance nonlinear energy load. For a safe and practical purpose, the inverter's rating should be equal to the rating of the PV array (Lee, 2015). However, others argue that the size of the inverter should only be 80–90% of the power rating required to save costs, since the PV-systems hardly generate its maximum required energy (Gougui *et al.*, 2019). However, other scholars disagree and argue that the size of the inverter should be 25–30% bigger than the required energy output (Jansson, Whitten and Schmalzel, 2011). Others agree and state that at high temperature, an undersized inverter becomes strained and regularly reduces the performance of the PV-system (Al-Ghussain *et al.*, 2023). However, it is important to note that the bigger the inverter, the more expensive it is. The configuration and reliability of inverters must be considered at the design stage of the system. In (Jordan *et al.*, 2020; Schweighofer *et al.*, 2023), problems related to poor inverter selection were reportedly the most common reason for poor performance of the installed PV-systems. System interruptions caused by inverters led to poor energy production for weeks. A detailed study on the performance of PV-system inverters can be found in (Schweighofer *et al.*, 2023). Likewise, a design margin in terms of inverter reliability is recommended (Sangwongwanich *et al.*, 2018).

2.5.3.5. Battery

Batteries should be sizeable enough to store ample energy for use during cloudy days and at night. Naturally, the nominal voltage of inverters and batteries should be the same. However, because batteries are used to store enough backup energy, it is recommended to design the PV-system with a battery capacity as high as the user can afford (Schweighofer *et al.*, 2023). This is because most batteries' usable capacity is usually only 20% of the rated capacity given by manufacturers. Factors considered in sizing of PV-system batteries are therefore its performance, durability, and depth of discharge. The best battery types are those that can withstand deep discharge. Battery types with short lifespan are normally the types that charge slowly and discharge rapidly. Alternative energy (AlternativeEnergy, 2019) calculates battery as a ratio of the total watt-hours needed from PV modules to the system voltage. A comprehensive guide for battery sizing according to accepted methods can be found in (IEEE, 2020; Chibuko *et al.*, 2023; Schweighofer *et al.*, 2023).

Batteries could be deep-cycle or shallow cycle and could also be either liquid electrolytes that are sealed or flooded or captive electrolyte (Lee, 2015). Shallow-cycle batteries are light, less expensive, but less durable than deep-cycle batteries that are highly efficient, require low maintenance, and have a high life expectancy. For that, a deep cycle is considered the best type of battery for PVS use (Sustainable Sources, 2015). However, (Stand-Alone PV, 2015) believes batteries with Nickel Cadmium (NiCd) plates are equally proficient such that some designers do not use a controller with it. Other researchers recommend the lead-antimony battery because it can endure deep discharges (Kumaresh *et al.*, 2014) but Lee, (2015) disprove, stating they require constant maintenance and are costly. They further stated that although the lead-calcium battery has a cheaper initial cost, its lifetime is shorter. A study in battery management claims the lithium-ion is one of the best energy storage batteries due to its low discharge rate and high power density (Okay, Eray and Eray, 2022). However, the authors assert that other automobile batteries and all other types of batteries that recharge slowly and discharge rapidly should be avoided where an alternative is available. Such batteries have a short lifespan. A review of battery selections for use in PV-systems can be found in (Khalid *et al.*, 2019) while detailed review of causes of poor battery performance can be found in (Chibuko *et al.*, 2023).

2.5.3.6. Charge Controller

Researchers argue that the charge controller should match the PV-array and batteries' voltage with a size directly proportional to the total input current of the PV, where the maximum value of current and voltage is the minimum value of the controller needed (Kumaresh et al., 2014). Others advice to size the charge controller based on the energy output and an added percentage of 1.3 as a safety ratio (Ali, 2018). However, it is important to note that the bigger the charge controller, the more expensive it is. Sustainable Sources (2015) thus recommends a way to reduce the costs of controllers. Instead of using a 100 amperes controller, 5 of 20 amperes controllers can be connected in parallel to produce excellent outputs. For the context of this study, it is recommended to seek the advice of trusted battery manufacturers on what controller works best with a client's battery of choice.

2.5.3.7. Wires and Cables

Numerous types of wires can be used for PV-system connections, but they must meet the requirements of the "UL 4703 Standard for Photovoltaic Wire" (ULStandards, 2019). Standard wires and cables are flexible. They also have moisture resistant coverings and insulation jackets to protect them during wet days or during harsh radiation periods. Experts in PV-system wiring often recommend the best wires to be used; however, it is important to note that human errors may occur, and this leads to the poor performance of installed systems. Likewise, it is also imperative to note that electricians who are familiar with indoor wiring do not necessarily have comprehensive knowledge of the exposed cables used in PV-system connections. The UL 4703 standard should be used to select the most suitable wires.

In a PV-system array of modules, there is usually a non-linear relationship between the current and voltage. However, there exists an operating point where maximum power is generated by the array. This is called maximum power point (MPP) (Chibuko *et al.*, 2023). However, it is often very challenging to obtain the MPP when a PV-system array is partially shaded (Dhimish et al., 2018; Gao *et al.*, 2023). Numerous studies have been conducted to derive algorithms for detecting MPP under partial shading conditions (PSC). As a result, several algorithms have been developed to improve the energy

efficiency in PSC to find the MPP. Ma et al. (2018) has complained that these algorithms use a sudden change in energy output as an input indicator for PSC and argued that this method is not efficient enough to distinguish between PSC and an abrupt change in the weather condition of the environment.

To avoid PSC from the design stage of the PV-system, it is imperative to apply machine learning techniques to detect the possibility of shading (Sarkar, 2014). Ma et al. (2018), Spataru et al. (2014), and Salem et al. (2015) have applied machine-learning based methodologies to detect shading of PV modules. In Ma et al. (2018), a sorting algorithm is used not only to predict shading but also to estimate the shading strength and voltage at MPP. It takes in the current, voltage, and cell temperatures of the PV-system array as input parameters. This method has been tested and validated on existing PV-systems. Although these algorithms successfully predict shading and can ascertain the MPP of the PV-system, its effect on the other components of PV (such as inverters) have not been fully explored (Dhimish et al., 2018). A comprehensive review (Chibuko *et al.*, 2023) offers a list of MPPT techniques that have been developed to maximise the output of PV-systems with a thorough appraisal of the advantages and limitations of each technique. In another study, a divide and conquer algorithm is proposed to reduce the effects of shading on the PV arrays (Gao *et al.*, 2023). These techniques increase the maximum power of the PV-system regardless of shading.

2.5.4. Mounting Types

In practice, installation and mounting could be done in three (3) ways. Modules could be mounted on the roof with PV panels mounted either flat on the roof facing the sky or PV panel sloping at an optimal angle using frames (Stand-Alone PV, 2015). The PV modules could also be mounted on open ground using a concrete foundation to support the structure. The PV-system could also be building-integrated attached to the structural support of the building. Building Integrated PV (BIPV) could be solar collectors, solar shingles, or solar slates, which replace roof tiles, so the additional cost is required. Some researchers have examined the performance of PV depending on its tilt angle (Al-Ghussain *et al.*, 2023). They considered monthly and (or) seasonal calculations to derive what optimum angle is best for different locations (Rhodes et al., 2014; Lee,

2015). Others (Roaf, Fuentes, and Thomas-Rees, 2012) believe PV modules should be slanted at an angle pointed at the sun. Nevertheless, Lee (2015) argues that the optimum angle of inclination should be perpendicular to sun rays and as horizontal and possible for maximum energy output. This increases energy production up to 35 to 50% in some months. Region specific techniques of determining optimal angle are provided in Al-Ghussain *et al.*, (2023).

All forms of mounting are effective. Nonetheless, researchers recommend ground mounting for stand-alone systems (Stand-Alone PV, 2015). They propose that roof mounting be avoided because they must be installed at the most compatible angle to the sun's radiation and are the most difficult to install. Others (Gulhane and Chavan, 2014) argue it could also be challenging to have fixed and secure attachments to the roof. There must also be enough airflow to cool the PVS. However, several module manufacturers and suppliers sell mounting hardware. This hardware is made to hold the PV panels' full weights, strong and durable to withstand the worst wind and last for decades. They are straightforward and cost-effective. A new trend in PV mounting type is the integration of green roof and photovoltaic technology. These are simply PV technology installed on a garden green roof. These reduce energy consumption by lowering the temperature of the building's roof while enhancing aesthetics. The benefits, limitations and trends of PV-green roofs can be found in a thorough appraisal of about 145 published articles by Shafique, Luo and Zuo, (2020).

Because buying and installing PV technology is expensive, experts are required to analyse the expenditure, life expectancy, safety concerns, and all other matters relating to PV system purchase and installations from start to finish. Only certified and experienced suppliers and installers should be used (Sinitskaya *et al.*, 2019). There is a substantial number of PV manufacturing companies and installers available worldwide, most who showcase their products on the web. There are also intermediary professionals who have extensive knowledge and are specialised in implementing PV technology who can help the homeowner communicate with installers (Rigo *et al.*, 2022). Similarly, electrical contractors may be proficient in electrical works but may not have professional PV design knowledge. If a PV system is government sponsored or subsidised, there may be a legislative requirement for choosing installers. EST (2011)

recommends the use of an installer licensed with the Microgeneration Certification Scheme. Countries that sponsor PV systems may undergo accreditations for installers. This will make them easy to find as such recommendations from satisfied customers can be used to find the best installers. When expert suppliers and installers are found it is essential to know if they have worked on grid-connected PV before as interconnections with the grid have unique connections. However, (Sivestre et al., 2015) recognize that off-grid systems could be more technically complicated, so experience in them is equally important. Installers should give customers the best advice regarding permits and licences from grid-connection and net-metering agreements. It is recommended to use organisations and companies certified by renowned domain accreditation standards such as energy savings trust (EnergySavingTrust, 2022) and carbon trust (CarbonTrust, 2022).

2.6. Classification and Rating Standards for Construction Products

Several renowned government and non-government agencies and research establishments publish important knowledge on the domain of sustainable building technology products. Some of these include Building Research Establishment, Energy Saving Trust, and the Carbon Trust.

Building Research Establishment (BRE) is a leading centre for building science, infrastructure, and research best known for its validation and certification system for the built environment (BRE, 2022a). BRE has provided standards by which building materials and components can be evaluated and certified. Since the development of the BREEAM (BRE Environmental Assessment Method) group of schemes, it has become the most widely used sustainability standard for buildings (Liu, Pyplacz, Ermakova, & Konev, 2020; RTF, 2021; BRE, 2022a; Allu-Kangkum, 2023). The Green Book Live (Bre, 2022b; WIPO, 2022) contains over 1500 accredited specifications through an A-Z catalogue of 'green' products approved under the BRE certification schemes. End users can select from these products to validate their commitment to zero carbon emissions. One limitation of the catalogue of products is it lacks a comprehensive list of the sustainable construction products itself; it only provides responsible sourcing certificates issued by BRE (company name, certificate and issue date).

Energy Savings Trust (EnergySavingTrust, 2022) and Carbon Trust (CarbonTrust, 2022) both aim to promote the use of sustainable and low carbon technologies in order to promote environmental sustainability and eliminate the climate crisis. They have reliable experts on energy efficiency, climate change strategies, low carbon solutions and recommendations for advances. They publish knowledge on sustainable building technology products and present key classifications that can assist companies, institutions and construction domain professionals to meet standard specifications for their products and their projects (energysavingstrust, 2022). Carbon trust has certified over 27,000 products across the globe, well placed as the leading label for carbon foot printing (CarbonTrust, 2022). The carbon trust footprint label appears on the individual products. A carbon footprint tells you the total greenhouse gas emissions generated by a product's entire life cycle (NIssinen, Seppala, & Heinonen, 2022). It also tells you if the product is certified renewable, certified carbon neutral, certified reducing each year or certified lower than other products (within the category) in the market. However, it only publishes the details of its certification and a list of certified organisations. Nonetheless, a user can traverse and select organisations that have earned brand reputation in sustainability. Concepts available within these standards could potentially be used in this study.

In a recent study, researchers review rating systems that are more recent and have extended advantages than the likes of BRE (Tang, Foo and Tan, 2020; RTF, 2021; Allu-Kangkum, 2023). These rating systems have been developed to verify environmental performance, quantify improvements in sustainability and to promote marketing (Allu-Kangkum, 2023). Leadership in Energy and Environmental Design (LEED) is believed to be the one of the most popular rating systems in the world (LEED, 2023). Other rating systems include BCA Green Mark Scheme (BCA, 2023), Pearl (RTF, 2021), Green Globes (GreenGlobes, 2021), and many more. The details of these rating systems are beyond the scope of this study. However, it is imperative to note that they mostly provide criteria and benchmarks for their sustainability ratings.

2.7. Advantages of Adopting Photovoltaic Systems

An advantage of adopting PV technology is that the more it is utilised, the more its cost of installation reduces. Researchers have reviewed the vital trends that have caused the prices of PV-systems technology to fall in recent years. The primary reason is the increase in adoption. Firstly, the installation of PV-systems has been largely supported by some governments Non-Governmental Organisations (NGOs) such as the Honnold Foundation, who give huge subsidies and incentives. Secondly, there is a recorded improvement in the production of more efficient solar cells (Kalay, Kılıç and Sağlam, 2022). These two reasons have led to a massive surge in the production and installation of PV-systems, which has forced the cost to gradually reduce. Therefore, PV-systems are benefitting from an economy of scale. The higher the demand of the technology, the cheaper it gets. The price per watt of crystalline silicon PV modules in 1977, 2006 and 2023 are \$76.67, \$3.25, and \$1 respectively (Malek, 2014; SEC, 2015; Hawkins, 2023). Since modules are the most expensive of all PV components, their price decrease has caused an increase in patronage (Sharma and Sengar, 2022). Evidently, the price per watt (\$/Wp) of PV-systems modules will continue to drop even without subsidies (Apostoleris et al., 2019). The more it is adopted, the cheaper it gets. In many places of the world, PV-systems have reached grid parity (Ali, 2018).

Vast familiarity of PV systems requires little or no maintenance and has long durability compared to other renewable technologies. It is suitable to be used by most remote communities where grid connections will likely never be established while diesel generators are too costly (Lawal, 2012). Furthermore, individuals connected to the grid want to contribute to energy production (EnergySavingsTrust, 2022) to reduce their long-run electricity bills so as not to be susceptible to the future price increase. In contrast, others enjoy the electricity independence that PV systems provide. Companies and industries have a green vision and an environmental commitment to reduce fossil fuel use and reduce greenhouse gas emissions (Hossan et al., 2011; Lee, 2015). In countries where feed-in-tariffs and net metering are available, household owners sell excess electricity their PV systems generate into the grid. The UK's Department of Energy and Climate Change (DECC) has specific criteria for a feed-in-tariff (FIT) scheme from which eligible individuals benefit. They get paid for a set rate for each electricity

unit they generate; the additional rate of electricity fed into the grid. It has become a considerable investment.

In some cases (places too far from the utility grid) now, it is cheaper to generate electricity using PV than install a new connection to the grid utility line and buy electricity from retail markets; this is grid parity. Several researchers believe once a technology reaches grid parity, it becomes economically efficient (Hirth et al., 2015b). However, it is essential to note that grid parity ignores grid fees, tariffs, and taxes added to the retail price.

2.8. Challenges in the Uptake of Photovoltaic Systems

Regardless of the numerous advantages of adopting PV-systems provided above, several challenges exist that hinders its adoption. Despite the positive level of adoption of PV-systems, much still needs to be done to ensure they become a mainstream technology. This section will examine the most important challenges that fit the scope of this study. There are poor performance of installed PV-systems, lack of clear guidelines in PV-systems designs and component selections and abundance of unstructured data on the technology over the web.

2.8.1. Poor Performance of Installed PV Systems

An imminent concern faced by individuals who install PV-systems is poor performance (Ma et al., 2018). Once the PV has already been installed, its performance can be monitored through Internet of Things (IoT) technologies (Kalay, Kılıç and Sağlam, 2022). Several studies have identified alterable and unalterable factors that affect the performance of PV-systems (Al-Ghussain *et al.*, 2023). The rationales for the disparity in the performance of PV-systems are many. Firstly, a PV-system consists of so many components (e.g., cables, inverters, batteries, PV-modules, etc.), with each of these components having different characteristics, including different performance indicators. Thus, evaluating the performance of PV technology as a system requires a detailed

understanding of the performance of the constituent components. Secondly, the different permutations in the combination of components can influence the performance of PV-systems. In practice, stand-alone and grid-connected PV-systems are examples that can perform differently because of the combination of different components and how they are integrated. Thirdly, the effect of alterable environmental factors such as location on earth, poor solar irradiance, the presence of dust, and shading on components compel thorough evaluation (Al-Ghussain *et al.*, 2023).

Frequently, the PV-system components that work best for one location may not necessarily be well optimised in another. Every location on earth has unique coordinates (longitude and latitude) and weather features such as temperature, humidity, wind velocity, and so on. Al-Ghussain *et al.*, (2023) studied the relationship between PV-system components and the temperature of the environment and concluded that modules ought to suit the ambient temperature of the location to be able to perform well. Modules that work best in Europe may not work properly in Africa. For example, monocrystalline modules are less affected by high temperatures; thus, they are suitable for locations around the equator, while polycrystalline modules are sensitive to high temperatures and may not generate the desired electrical energy output (Al-Ghussain *et al.*, 2023). In a case study conducted, modules designed to be used in the USA were adopted for use in Fortaleza Brazil and a performance error (a criteria used by the researchers to measure array loss (LA) higher than 20% was observed (Bharadwaj and John, 2019). Unexpected climatological effects such as lightning strikes, hurricanes, tornadoes, visible hail, and large snowstorms can all have terrible effects on the installed PV system (Jordan *et al.*, 2020). There is a need for academics and specialists in this domain to use location-specific values to avoid location-related performance problems.

In a study of performance evaluation of PV-systems, it is argued that poor solar irradiance has led to the poor performance of PV-systems (Sambo *et al.*, 2014; Ma *et al.*, 2018). The poor radiation received by solar panels is often due to wrong angles of inclination. If the tilt and orientation have been calculated wrongly during the design stage, then the optimal position of the PV-system in relation to the direction of solar radiation cannot be obtained (Al-Ghussain *et al.*, 2023). Performance parameters such as the reference yield, Y_r , will give incorrect values if there are errors in the irradiance

values. In (Suetsugu, Takahashi and Fujiwara, 2018), an installed PV-system generated a small amount of electric energy due to the poor design methods used for computing the correct angle of inclination. In another study, scholars concluded that, for an optimal performance, the angle of inclination should be very close to the angle of latitude of that location (Al-Ghussain *et al.*, 2023). Other researchers have stated that poor PV-system performance often results from the incorrect selection of the solar irradiance measurement parameters to use (Hassan *et al.*, 2015). Once there is a poor absorption of solar irradiance, the poor performance of the PV-system will be inevitable (Al-Ghussain *et al.*, 2023).

One of the most important weather-dependent factors that affect PV-systems' performance is dust (Younis and Alhorr, 2021; Lu, He and Zhao, 2023). Numerous performance analyses conducted have attributed PV-system power loss to dust deposition on the panels (Liu *et al.*, 2019; Al-Ghussain *et al.*, 2023), because the accumulation of dust prevents the passage of solar radiation to it. Researchers have studied the long-term effect of dust accumulation on PV modules and evaluated the effect of cleaning on system performance (Kazem *et al.*, 2022). The amount or level of the performance loss is, however, determined by the nature of the dust, climate condition, wind velocity, PV-system angle of inclination, and array surface conditions (Al-Ghussain *et al.*, 2023). Studies have deliberated on the effects of different types of pollutants within the accumulated dusts (such as silica, sand, red soil, ash) to determine how PV output is affected based on the proportions of the dust components (Kazem *et al.*, 2020).

Dust particles are scattered in the atmosphere and are easily carried by the wind. For that, researchers claim that the nature of dust is site-specific and is influenced by the environment (Lu, He and Zhao, 2023). In an earlier study on the effect of the performance of a PV-system, it is observed that the dust elements generated from industrial environments have caused an 80% drop in the electric output of PV-systems (Hassan *et al.*, 2015). Another experiment was conducted on the effect of airborne dust on PV-systems' performance and observed a decrease in efficiency of up to 65.8% (Vieira *et al.*, 2018). Others have revealed that areas of high humidity also have a high tendency to have accumulated dust on panels (Zaihidee *et al.*, 2016). Lu, He and Zhao,

(2023) investigated dust accumulation in relation to wind velocity and observed that slow wind patterns support dust collection while, in contrast, fast wind would clean up the dust amassed on panels. Similarly, a horizontal PV-system installation is more likely to collect dust than an installation made inclined at a tilt angle (Lu, He and Zhao, 2023). Additionally, a sticky panel will be more likely to accumulate dust than a less sticky surface. Simple maintenance procedures such as the cleaning of the panels are routinely required to reduce the effect of dust on performance (Ma et al., 2018). Further details on the effects of dust on the performance of PV is discussed in Younis and Alhorr, (2021).

Shading is one of the most damaging factors in the poor performance of PV-systems (Khosrojerdi, Gagnon, & Valverde, 2021). The shading of PV panels causes non-uniform solar radiation due to the temporary obstruction of solar rays by trees, neighbouring buildings, utility poles, clouds, or other obstacles (Ma et al., 2018; Ahmad et al., 2020). Studies undertaken have proved that once a part of the PV-array receives less irradiance due to shading, its temperature becomes lower than the unshaded part of the array. The thermal distribution across the array becomes non-uniform, and thus the PV-system performance decreases (Mudlapur et al., 2019). Shading conditions also cause hot spots on modules, and these damage the cells in the short run (Ma et al., 2018). In a study by (Dhimish et al., 2018), it was established that the shading of arrays causes inconsistent panel voltage and shifts in power transmissions. These severely affect the performance of the converter and subsequently the total output of the PV-system.

A previous study (Jordan et al., 2020) published a performance analysis of 100,000 installed PV-systems and found that problems relating to PV-system performance are caused by performance quantification methods, ineffective grid data collection, poor weather conditions, hardware failure, installation, and quality assurance. Building/roof renovations during which the installed PV-system may be switched off or temporarily removed (for renovation purposes) are also common causes of poor PV-system performance. In this review (Jordan et al., 2020), an average energy production loss of 20% was recorded, with a continuous increase in the subsequent years. From the occurrence and maintenance reports gathered by the authors, performance loss rates were apparent and clearly visible, although they could be recoverable with time. Another study (Dag and Buker, 2020) performed a performance assessment of

PV-systems by observing different module types and provided a report of their energy outputs after 2.5 years of use. Al-Ghussain *et al.*, (2023) discussed the effects of ambient temperature, dust accumulation, and wind speed on the PV power outputs. A similar study (Venkateswari and Sreejith, 2019) appraised the factors that influence the efficiency of PV-systems; however, this review only considered material types, Maximum Power Point Tracking (MPPT) techniques, and converters. Most available studies in the literature address single factors. For instance, the effect of dust on the performance of PV-systems (Younis and Alhorr, 2021; Lu, He and Zhao, 2023); energy loss due to the soiling of PV-systems (Conceicao *et al.*, 2019), the performance assessment of PV modules (Campo-Avila *et al.*, 2019), efficiency of PV battery under realistic irradiance and load (Chibuko *et al.*, 2023). These studies focused on single factors affecting the PV-system performance.

2.8.2. Challenges in Sizing and Selection of PV Components

There is not a single standard method of selecting and sizing of PV components as observed in chapter 2.7.1 above. There is a multiplicity in the design methodologies used in selecting and designing PV-systems. They largely differ. Each method for the design of PV-systems requires different assumptions, leading to different accuracies in the design of the different components and systems. Poor design methods create a great difference between the projected energy output and the actual output obtained after installation. Since PV-system output varies with different component characteristics (sizes, efficiencies, and performance indicators), design methods must include a detailed understanding of the selection and performance of each component. Furthermore, different permutations for the combination of components must be analysed because different permutations could result in dissimilar performance estimates. For example, since the connections for stand-alone and grid-connected PV-systems are different, methodologies to obtain compatible configurations are different and challenging to decide. The synchronisation and optimization of the different components would ensure that the PV-system provides the required and intended energy output.

Recent studies have revealed that one of the reasons installed PV-systems fail is due to poor design methods. Studies were conducted to compare the use of ambient temperature against module temperature to determine solar irradiance, and a large difference in the performance of the PV-systems was indicated because of the different design parameters/methods used (Gasparin *et al.*, 2022). In Suetsugu, Takahashi and Fujiwara, (2018), an installed PV-system generated a small amount of electric energy due to the poor design methods used to determine the angle of inclination. This poor performance was due to the fact that the modules were installed at different angles. This design method is unusual from the conventional approach of installing modules at a uniform angle towards the sun's radiation. Jansson, Whitten and Schmalzel, (2011) describe the impact that module inter-row spacing design methods have on the electric energy output and performance of a PV-system when different depths of shading are applied. Similarly, errors in inverter selection and sizing strains the entire PV-system and drastically reduces its performance (Al-Ghussain *et al.*, 2023). In (Jordan *et al.*, 2020; Schweighofer *et al.*, 2023), problems related to poor inverter selection were reportedly the most common reason for poor performance of the installed PV-systems. System interruptions caused by inverters led to poor energy production for weeks. In a paper by (Da-Silva, 2019), different design models were analysed to estimate the solar irradiance for a PV-system resulting in dissimilar values for energy yield. This difference in energy yields presented proves that different design methods result in the different performances of PV-systems where the required electric energy output is not generated due to errors in the design stage. These most often confuse the intended user as to which methods to adopt.

Another factor that hinders adoption under design methods is lack of in-depth knowledge of the design methods. Knowledge on PV technology is vast and if the homeowner personally researches, designs his/her PV system and selects the components by themselves, there is no guarantee that the knowledge obtained is enough to make an adequate design. However, since self installation is rare, installation companies/installers are a link between the manufacturers and the users. Most often, they are responsible for educating an interested homeowner on the technology through sales and installation (Rigo *et al.*, 2022). Conventionally, Once the client contacts the installer and provides their energy requirements, the installer develops a

location-specific design of components and cost quotation that the client can review. Once it is approved, the installer obtains the components, handles the regulatory process and installs the system (Sinitskaya *et al.*, 2019). The installer makes decisions on the technology manufacturer/supplier, price and quality of installation. Following this routine, it means the client's PV installation experience is significantly determined and affected by the decisions made by the installer. Furthermore, there is much more research on enhancing the PV as a technology and the PV energy sector in general than on decisions made by users or installers (Rigo *et al.*, 2022). The knowledge gap is a challenge that holds back adoption of PV technology (Sharma and Sengar, 2022). Various solar training programs exist to bridge the knowledge gap (Solar-Energy, 2023; SRH, 2023). However, it is still vital to conduct more research on decision making tools that would enable clients and installers to make informed decisions on design and selection of PV.

2.8.3. Abundance of Unstructured Data on Products Over the Web

One of the challenges in the uptake of PV-systems is the difficulty users face when trying to obtain relevant data on products over the web. As already mentioned above, there is multiplicity in design methods. However, even when a user identifies a design method suitable for use, it is challenging to know where to find the products best suited for the project. Thousands of PV-system products are available in the market with different and unique characteristics such as their types, sizes, outputs, efficiencies, costs, makes and models. These components also have different performance indicators and many more detailed knowledge one requires for their uptake. It is often overwhelming to the user to find the right (and all the) knowledge they require on these components. Similarly, these products are manufactured by hundreds of manufacturers who showcase their products through their websites. Therefore, it is typically challenging for intended users to install PV, to traverse hundreds of available manufacturer websites to find the most appropriate components or the best manufacturer to purchase their products from. It is also challenging to obtain the right information (from the manufacturer websites) on which products would give the user the required electric output they seek. Even when

users browse for specifications on the web, results are most often thousands of irrelevant pages that are not implicit to the user's needs.

To summarise, clients face multiplicity when sizing and selecting components due to the multiplicities in the unique attributes each component has and the availability of hundreds of product manufacturer websites. However, it is imperative to note that one of the main causes for this challenge is the unstructured nature of the available abundant information over the web. Manufacturers represent their components in non-machine processable formats. The features of the components cannot be processed by computer agents or softwares to aid the user to make informed decisions. There is no shared formal taxonomy of PV technology and its components to represent the products efficiently. There is thus a lack of a unified system to find all the required information. There is also a lack of a single decision support system that would provide essential information to help a customer to select the most suitable PV system based on their electric needs.

To tackle the challenges associated with representing and searching for PV components on the web, a scalable solution is imminent. Therefore, it is imperative to develop better ways of information representation on sustainable building technologies and photovoltaic systems on the web, so construction professionals and users can easily obtain useful information with the aid of intelligent decision support tools. Firstly, knowledge modelling of information on PV systems and products is required so they are represented in machine processable formats. Secondly, a decision model to help users issue queries and retrieve specific attributes of the product they explicitly require is necessary. To overcome these challenges, the use of a machine-processable language that provides semantics to electronic information and issue queries is proposed (Gruber, 1993). The semantic web and its associated technologies improve information representation, inferencing, and interoperability. These have emerged to create a globally connected knowledge base of information presented in machine-understandable formats that enable software agents to perform complex tasks to provide solutions to users (Gruber 1993, Berners-Lee et al., 2001, Antoniou et al., 2012). The next chapter provides an overview of the semantic web and its technologies.

2.9. Conclusion

This chapter provided an overview of sustainable building technology products with an emphasis on photovoltaic system technologies. Sustainable construction was discussed in respect to sustainability and sustainable development principles. The impacts of the applications of sustainable building technology products and renewable energy technologies in mitigating the continuous environment concerns on carbon emissions is discussed. An overview of photovoltaic systems outlined the multiplicity in its components and its design methods. It was argued that despite the large volumes of research on photovoltaic systems technology and its continuous adoption, several challenges are faced in the uptake of the technology. To overcome these challenges, it is further argued that there is an eminent need to use state-of-the-art semantic web technologies to model knowledge on photovoltaic systems domain to aid users in making informed decisions when it comes to the design and selection of the technology. This led to the investigation of knowledge modelling techniques and semantic web technologies in chapter 3.

3. Review on Semantic Web Technologies

3.1. Introduction

The review of sustainable building technology products outlined the need to explore advanced knowledge management techniques for managing domain knowledge. Therefore, this chapter investigates the semantic web technologies and how they can be adopted to facilitate valuable recommendation for sustainable building technology products. This chapter reviews the emerging semantic web technologies, their strengths, weaknesses, and their potential in providing solutions to the problem domain. Since ontology is one of the most fundamental aspects of the semantic web, the chapter further studies ontological engineering, its approaches, methodologies, tools, and several development languages. The key semantic web technologies that fit the context of this research are presented with justification of why they are adopted.

3.2. Weakness of the Current Web Technology

The volume of information on the current World Wide Web has grown overwhelmingly since the early 2000s where the idea that knowledge should be open, shared and reusable became the trend (Sánchez de Castro, 2022). This continuous expansion is also due to growth in academic research, online publications, business and product representation, increase in global information sharing by governments, a surge in circulation and analysis of information through social media, and so much more (Gandhi and Madia, 2016). This exponential growth in published information on the web has led to numerous research challenges. This includes information overload, poor representation and retrieval challenges. It is challenging to find relevant information amongst the overburdened data content of the web (Patel and Jain, 2019). Similarly, the current web is designed for only human consumption and does not allow machine automated processing (Pascal, 2021; Belozarov and Klimov, 2022). Furthermore, informal retrieval through search engines has very low precision since they rely on content-independent algorithms. The limitations and inefficiencies of the current web are well elaborated in the ensuing sections.

3.2.1. Presentation of Information

Most documents on the web are represented using semi-structured and non-machine-processable languages that the computer does not understand. The Hypertext Markup Language (HTML), the syntactic language for the current web, displays information and delivers it only for human-consumption (Belozerov and Klimov, 2022). An important challenge in representing information in HTML is that HTML tags do not describe the meaning of data included in its documents. It is difficult to extract meaning from text represented in HTML. This information representation format cannot support automatic processing and reasoning by the machine, making the web a 'web of documents' rather than a 'web of meaningful data' (Patel and Jain, 2019). It is important for users to derive common sense basis automatically from published data on the web; hence, expert knowledge should be represented in a format that is computable (Sánchez de Castro, 2022). In summary, since HTML is simply about the visualisation of information that is not machine understandable, it does not support what is required of the web and can no longer serve its purpose.

3.2.2. Current web Search Techniques

To obtain information from the web, users traverse the web and retrieve information through catalogues. However, since catalogues are created by human experts and it is difficult to maintain as the web exponentially grows, other search techniques were devised to eliminate the human involvement in cataloguing web pages. Several web search techniques have been reviewed in literature (Akhoon, 2019; Rana, 2020) with different content, speed, size and search options. However, for the purpose of establishing a rationale in the context of the semantic web, the most common search techniques such as keyword search and natural language processing will be examined below.

In the keyword-based search, keywords supplied in queries are matched to the keywords in meta-tags of the HTML design of web pages. Keywords are matched using an indexing mechanism that stores information subject to frequency of words, the system computes a response by matching the words against stored indexing. The user often experiences either of the following.

- Firstly, thousands of unrelated pages are generated where in most cases the given keyword may appear only once and results mostly turn out irrelevant to the user's requirements. Similarly, a word can have several meanings. For instance, a search query with homonyms such as *a tree bark* or *a dog's bark* will pull up irrelevant documents. The huge volume of documents obtained due to these reasons is often problematic to the end-user to traverse.
- Secondly, there could be no result at all when terms used are not found in expected pages. Slight difference or an error in spelling of the search keywords affects the statistical score of documents.

Users often expect an aggregated result from their search, which often involves obtaining data from several resources and integrating the content to give the user the desired result. Keyword search engines do not provide the results that are implicit to the end-user's needs. Thus, the precision of search results is low. Queries from users are often ambiguous and the search engines lack the capability to understand the context and relationships of search terms provided (Belozarov and Klimov, 2022). It retrieves information but does not provide any semantics about the content.

Overcoming the inefficiencies and limitations of the current web (information overload, representation in HTML and low precision of search), entails the use of machine-processable languages to add a layer of meaning to data on the web (Berners-Lee et al., 2001; Sánchez de Castro, 2022). The semantic web aims to provide a standard for effective communication between human agents, data, and computers (Patel and Jain, 2019). Hence, it proposes to represent data in machine processable formats. Technologies and languages have emerged to become the 21st-century solutions to effective elicitation, structuring, representation and reasoning of information on the web. As previously defined in chapter 1, semantic web is defined by Berners-Lee et al. (2001) as a globally connected database of information presented in machine-understandable formats that enable software agents to perform complex tasks to provide solutions to users. The semantic web provides inference advantages over stored data (Patel and Jain, 2019). Typically, SW can be viewed in two ways; as a stack of technologies layered on one another for data representation and processing and SW web

as an enormous repository of machine-processable knowledge that have been compiled based on those stack of technologies. Both aspects are further discussed later in the chapter. Since its emergence, significant progress has been made in standardising these technologies that make up the semantic web architecture discussed in the next section.

3.3. Semantic Web Architecture and Standards

As an extension of the current web, the semantic web has proven itself in numerous domain applications for academia and industries (Abanda, Tah, and Duce, 2023; Pauwels and Terkaj, 2016; Pauwels, 2018; Huang et al. 2020; Costa et al., 2020; Gilani, Quinn and McArthur, 2020; Hu *et al.*, 2021; Sánchez de Castro, 2022) through utilisation of its well layered, defined and standardised technologies. These technologies include Resource Description Framework (RDF), Ontology Vocabulary (Web Ontology Language OWL), Rules Interchange Language (RIF) and SPARQL Protocol and RDF Query Language (SPARQL) (Sánchez de Castro, 2022). The hierarchical architecture of the SW stack allows the technologies to build on the proficiencies of each of the layers below it. Each of these technologies has unique capabilities that they add to make the SW feasible. They are standardised by the World Wide Web Consortium (W3C) (W3C, 2020) and used for development of SW applications. The ensuing section discusses the concept of ontology; the most applied SW technology to solve challenges of representation of domain knowledge.

The lower layer technologies include Unicode, Uniform Resource Identifier (URI), and Extensible Markup Language (XML). These are part of the earliest technologies used on the web, but they provide a strong foundation for the semantic web. The **Unicode** provides consistent encoding and representation of data in the semantic web (Belozerov and Klimov, 2022) through unique numerical values for each character (or string data) it encodes, so computers can explicitly interpret that. Unicode is best used to improve software localization (W3C, 2020a). The **Uniform Resource Identifier (URI)** uses global identifiers to represent addresses of resources on the web (Nrip, Ghorpade and Musale, 2021). This is done in three ways; uniform resource names (URNs, for off the web referencing such as hard copies of books in a library), uniform resource citations (URCs, e.g., author of a resource on the web), and uniform resource locators (URLs

represent web resources such as <http://literatureReview.com>). The URI simply identifies resources.

The Extensible Markup Language (XML) provides serialisation syntax for data on the semantic web (Pascal, 2021). XML's strength is its ability to create semantic markup of web pages and allow numerous serialisations for markup languages (Horrocks et al., 2001). XML describes hierarchies and makes it easy to control contents via the XML schema. However, combining distinctive XML data in the same application is very complicated. Thus, XML is not an ideal tool for data integration. Likewise, processing XML data mainly involves handling syntax while ignoring the data's semantics (Oliveira *et al.*, 2020). XML is currently not a relevant SW technology as it is often substituted by other syntaxes. XML lacks the basic semantics to support semantic web necessities; hence, a more robust model for describing resources (RDF) emerged.

Resource Description Framework (RDF) represents data in the form of machine-readable syntaxes (RDF/XML (.RDF), Turtle (.TTL)) (Antunes, Cardoso and Barateiro, 2022; Belozarov and Klimov, 2022). It is the W3C open standard for data representation and data exchange on the web (Horrocks, 2008). RDF specifies unambiguous semantic relationships of resources within a domain using three RDF syntax; subject, predicate, and object to make clear statements about resources (Izquierdo *et al.*, 2021). An RDF statement would look like the following: <subject> has a property <predicate> valued by <object>. Its graph-like nature also allows resources to be easily linked and published together in the form of triple stores (Van Assche *et al.*, 2023). This easy connection makes the semantic web qualify as a semantic network of information represented in the form of RDF graphs (Similea et al., 2017). However, researchers (Hall & O'Hara, 2009; Bechhofer & Horridge, 2011; Chawla et al., 2016; Van Assche *et al.*, 2023) claim the RDF has limited resource descriptions such as domain constraints, range constraints, property, and descriptions. Likewise, RDF has limited expressive power. The Resource Description Framework Schema (RDFS) is an extension of the RDF model developed to overcome the limitations (Antunes, Cardoso and Barateiro, 2022). RDFS represent properties of RDF-based resources with added semantics to generate vocabularies (El Idrissi *et al.*, 2022). Although an RDFS interpreter deduces additional RDF statements available via constructs, Horrocks, Krotzsch and

Simancik, (2013), RDF (2013), Jagvaral et al. (2017), El Idrissi *et al.*, (2022) argue that RDFs does not provide sufficient referencing required to support the semantic web. Formal semantics for instances defined in RDFs are not provided. Thus, its expressivity is restricted. Therefore, an advanced language OWL was developed.

Web Ontology Language (OWL) is a W3C recommended language designed to provide explicit and formal conceptualization of domain knowledge (W3C, 2012a). It provides well-defined syntax and semantics, adequate reasoning support, and sufficient expressive power (Buoncompagni, Kareem and Mastrogiovanni, 2022). It overcomes limitations and has more semantics than the RDF, and RDFs vocabularies. Syntactically, RDF and OWL are compatible (Pascal, 2021). OWL generates complex RDF statements such as cardinality and type restrictions, and RDF graphs constructed with OWL to make up OWL ontologies. It is important to note that OWL ontologies are generally represented as RDF documents since OWL documents can be serialised as RDF documents as well (El Idrissi *et al.*, 2022). OWL is designed for software applications that perform reasoning on the content instead of just representing information (Rector et al., 2014; Jagvaral et al., 2017). It specifies a collection of operators to develop concept descriptions. OWL has three expressive subsets known as OWL Lite, OWL DL and OWL FULL used in implementations to specify classes, properties and restrictions (W3C, 2012a).

The difference between OWL Lite and OWL DL is its level of expressiveness, while the difference between OWL DL and OWL Full is the extent to which the user requires the RDFs facilities and level of completeness. Developers adapt the sublanguage that best suits the needs of their ontology (Antunes, Cardoso and Barateiro, 2022). OWL Full is complex (Belozerov and Klimov, 2022). It is highly implausible to have a reasoning software that can support all its reasoning features. Similarly, OWL Lite is limited in terms of complete representation of class hierarchies in ontology editing software. However, OWL DL supports users who require computational completeness and high expressive power (Lenzerini, Lepore and Poggi, 2021) through description logic (Patel and Jain, 2019). For these reasons, OWL DL will be adopted for use in this study. This research requires advanced reasoning and therefore it would benefit from the many reasoning plug-ins that exist which support OWL DL.

OWL can easily express which objects belong to what classes and what property values are for each object. Conversely, OWL's significant extension of RDFs can provide restrictions on how properties behave (Horrocks et al., 2014a). However, OWL has syntactic challenges to extend the DL, RDF/XML, RDFs and still maintain an easy syntax to understand. There is also a demand for its expressive power to go way beyond its current level. To overcome these difficulties, OWL2 was created by the W3C OWL working group (OWL, 2012b). OWL2 extends OWL and provides three necessary improvements (Horrocks et al., 2014b). Firstly, OWL 2 extends OWL with the ability to moderate cardinality restrictions. Secondly, it provides extended support for data types and annotations, and thirdly, it increases the expressive power of the complete language (Lenzerini, Lepore and Poggi, 2021). The level of expressiveness of the OWL language determines what is represented in the OWL ontology (Pauwels and Terkaj, 2016). The W3C recommendations and specifications for OWL are available at OWL (2012a) and OWL 2 language at W3C (2012b).

RDF, RDFs, and OWL are practical semantic web languages for ontology and representation of data on the semantic web; however, as soon as the data has been published, a query language must create meaning out of data that is published. The SPARQL Protocol and RDF Query Language is the standard query language for the semantic web; it provides both a query language and a protocol for RDF data on the semantic web (Izquierdo *et al.*, 2021). The main purpose of this query language is to execute queries against data stored in RDF triples and graphs to develop high-level applications across various domains (Izquierdo *et al.*, 2021). It queries unknown relationships in the RDF data, vividly describes the queried RDF graph subgraphs, and provides results that match the user's description (El Idrissi *et al.*, 2022).

Although SPARQL is very significant in the semantic web, it has shortcomings. SPARQL does not provide collective functions at the time of writing; external languages are used to run aggregations, thereby making RDF technologies complex (W3C, 2015). Researchers have also argued that SPARQL is a read-only language and doesn't allow users to modify RDF statements (Shadbolt et al., 2014). Likewise, data vagueness is also not fully supported by SPARQL because there are restrictions when expressing the

length of variables (W3C, 2015). Moreover, it is also challenging to denote negations, e.g., find all schools that take 2-year-olds but do not teach arithmetic. Likewise, because the SPARQL can be used to query a single RDF and a collection of RDF files, it is essential to know which RDF files to be queried before running a query. However, this limitation is yet to be overcome in some cases.

Similarly, the Semantic Web Rule Language (SWRL) uses logic to express IF-THEN rules in SW applications (Hu *et al.*, 2021). It identifies rules in ontologies, and maps between the ontologies, agents and services (W3C, 2015). SWRL is compatible with protégé-OWL and supports graphical editing of rules, rule-engine bridge, and incorporated libraries (Horrocks *et al.*, 2014a; W3C, 2015). However, SWRL is only a semantic web rule language and not a query language (El Idrissi *et al.*, 2022). Numerous researchers have adopted Resource Description Framework (RDF) (Bonduel *et al.*, 2018), SPARQL (Werbrouck, Senthilvel, Beetz, & Pauwels, 2019; Lenzerini, Lepore and Poggi, 2021) and SWRL (Hu *et al.*, 2019; Hu *et al.*, 2021) in their applications. These standards combined, create a semantic environment for effective knowledge capture, integration, representation and reasoning. In Izquierdo *et al.*, (2021), an algorithm is proposed to translate the keyword-based RDF query into SPARQL and generates a graph relevance ratio based on the number of relevant and irrelevant triples within an RDF graph. In another study, researchers attempted to repurpose SPARQL into mapping languages such as XSPARQL and SPARQL-Generate to transform heterogeneous semi-structured data sources into RDF (Van Assche *et al.*, 2023). Further descriptions of RDF, OWL, and SPARQL are available on the official W3C websites (W3C, 2015).

As a recap, each semantic resource has a unique URI and are linked together in the form of RDF graphs. Once generated, RDF graphs can be converted into triple stores. Stored data can then be queried using SPARQL queries. Published semantic data in the form of RDF graphs that are linked make up the linked data. Linked Building Data (LBD) is a large network of building datasets that are semantically linked together and exchanged amongst stakeholders in the domain (Antonios, Konstantinos and Christos, 2023). In Sánchez de Castro, (2022), the authors presented how knowledge graphs as part of linked data can be used to develop functional applications in several domains. Knowledge graphs are considered to be more industry-led, shifting from academic

research which largely contributed to the success of linked data (Pascal, 2021). Recent implementations of LBD can be found in (Terkaj, Schneider and Pauwels, 2017; Bonduel, et al., 2018, Djuedja, et al., 2019, Wagner *et al.*, 2022).

The logic layer uses first-order logic to write rules, the proof layer executes the rules and the trust layer confirms whether the given proof should be trusted (W3C). The trust layer provides the web of trust through encryptions and signatures. These digital signatures and proof languages are vital to the semantic web because users should trust the context of an application. Since this study deals with knowledge modelling to develop computational ontologies in the context of the semantic web (Shadbolt and Smart, 2015), the next section discusses ontological engineering and the reasons for developing ontologies. The methodologies and tools used in ontology development are further analysed.

3.4. Ontological Engineering

Ontology is one of the most fundamental aspects of the semantic web because it transforms information from its unstructured format to a structured representation that is machine-processable (Patel and Jain, 2019). Ontology has been known philosophically as the theory of knowledge until Gruber (1993) gave a simple definition of ontology in respect to computer science where he explained ontology as “a formal explicit specification of a shared conceptualization”. Although this is the most popular definition, researchers often propose different definitions to suit the domain of their applications (Gilani, Quinn and McArthur, 2020). Smith (2003) defines ontology as a “structure of things/objects, their properties and relations. In another definition it is known as a “computational artefact” that represents domain knowledge (Stephan, Pascal and Andreas, 2007). In Belozarov and Klimov, (2022), ontology is defined as a mathematical model that obtains logical inference from data. However, the definition of ontology by Pascal, (2021) and Spoladore and Pessot, (2022) is deemed most suitable for the purpose of this study. The authors define ontology as ‘a knowledge base’ that provides formal definition of concepts and their relationships, ‘specified in a knowledge

representation language'. This knowledge base is developed within a selected problem domain for efficient decision making.

Ontology development revolves around three main applications: communication (between people and agents), interoperability (between systems) (Spoladore and Pessot, 2022) and general system engineering (re-use, reliability and trust). Naturally, every ontology has a common pattern; concepts (the conceptual part of a domain), instances (objects with real value), relations (the relationship between concepts and instances), and axioms or role restriction (constraints on values of concepts and instances using logic-based languages) (Kotis, Vouros and Spiliotopoulos, 2020). Other elements include attributes (properties), restrictions (formal description of what must be valid for some input to be accepted), and rules (Buoncompagni, Kareem and Mastrogiovanni, 2022). In the earlier days of ontology research, any published taxonomy with a structured metadata on domain knowledge is considered a form of an ontology.

A functional ontology describes logical relationships between the defined concepts, and instantiates and validates the knowledge model through rules (Antunes, Cardoso and Barateiro, 2022). Semantic reasoners can then be applied to derive enhanced and implicit analysis of the captured construction knowledge. An increased interest in ontologies has led to new methodologies of knowledge expression, greater reasoning capacity and improved knowledge interoperability between different systems. Ontological engineering incorporates all activities involved in ontology development, life cycle, methodologies, languages and tools. Different problem domains require different solutions and unique approaches to address challenges. For that reason, there is no particular blueprint for which set of procedures to adopt. Therefore, different tools, languages and methodologies have been created and well researched in literature for different application needs (Geng *et al.*, 2020). The next sections discuss reasons for developing ontology followed by ontology development methodologies, languages and tools.

3.4.1. Reason for Developing Ontology

One of the primary reasons why ontologies are developed is how they differ from databases. It is imperative to understand that databases are designed to hold data for

specific applications while an ontology provides a comprehensive amount of knowledge on a domain and is used outside of an application with other software components (Buoncompagni, Kareem and Mastrogiovanni, 2022).

Furthermore, researchers in this domain have outlined several reasons why they have adopted ontologies in their studies. Early ontology researchers (Harmelen et al., 2000; Horrocks et al., 2001) outline the critical purpose of ontology. Firstly, to enable people, agents, and machines to share a common understanding of the structure of information and knowledge (Kotis, Vouros and Spiliotopoulos, 2020). Secondly, to analyse and facilitate re-use of domain knowledge, and thirdly to make domain assumptions clear. In a comprehensive study conducted to represent sustainable construction products, ontology emerged as the most potent machine-processable language for representing domain knowledge to solve their domain problem (Abanda, Tah and Keivani, 2013). In Geng *et al.*, (2020), an ontology is developed because of its capacity to perform cross-domain product comparisons. Likewise, ontologies have been widely adopted because they offer effective knowledge management (Spoladore and Pessot, 2022).

Another advantage of ontology is the standard vocabulary it provides. Currently, knowledge is represented on the web in an ambiguous format, where search precision is often affected by synonyms (different terms that mean the same thing) and homonyms (a single term with different meanings) correlated to one another. Ontology defines a structured vocabulary for representing domain knowledge in simpler terms. Entities (whether abstract, existent or non-existent, dependent or independent) are described in a domain alongside their relationships to each other. The difference between a natural language taxonomy and one specified through an ontology is the hierarchy definitions (Geng *et al.*, 2020). An ontology provides detailed and formal specification of subclasses and allows researchers to write formal and shared conceptualization of domain knowledge (Spoladore and Pessot, 2022; Antunes, Cardoso and Barateiro, 2022). This guarantees consistency.

Furthermore, because knowledge on the web is frequently expressed in natural languages, they lack semantics. One of the advantages of ontology development is that it adds well-defined syntax and semantics, adequate reasoning support, and sufficient

expressive power to domain knowledge, thereby making it machine-processable (Antunes, Cardoso and Barateiro, 2022). Similarly, one of the most popular reasons researchers develop domain ontology is for interoperability (Hu *et al.*, 2021). Since an ontology provides a common interpretation of the domain knowledge, all agents and computer applications within that domain can efficiently share and re-use domain knowledge (Antonios, Konstantinos and Christos, 2023). It is therefore agreed that ontologies are designed to support interoperability as part of the semantic web. For an increased interoperability, experts should use controlled or already existing vocabulary (re-use) to promote a shared understanding of that knowledge.

Several ontologies have been developed as solutions to different construction and building domains challenges. Some of which are for building products (Pauwels and Terkaj, 2016); for online customer product reviews (Geng *et al.*, 2020); smart and ongoing commissioning of buildings (Gilani, Quinn and McArthur, 2020), in business intelligent systems (Antunes, Cardoso and Barateiro, 2022); and so many more. Further discussion on applications of ontologies is done later in the chapter. The development of ontology involves the well researched ontology-based knowledge modelling techniques in literature (Yun *et al.*, 2021). These are otherwise known as the ontology development methodologies. The ensuing section discusses earlier and recent ontology development methodologies.

3.4.2. Earlier Ontology Development Methodologies

Ontology development is a vital aspect of this research, therefore it is essential to review and distinguish between many ontology development methods in order to gain an insight on which methodology is most suitable to adopt in this study. These methodologies help the ontology engineer to transform an informal knowledge successfully into a logic-based representation (Spoladore and Pessot, 2022). No domain has one exact modelling design because specifications vary accordingly. However, irrespective of the difference in the methodologies, Berners-Lee *et al.* (2001) asserts that an ontology development must be an iterative procedure. It is crucial to bear in mind its clarity, coherence, extendibility, and minimal ontological commitment because successful development and effective use of an ontology depends on the methodology used (Berners-Lee *et al.* 2001; Spoladore and Pessot, 2022). Kotis, Vouros and

Spiliotopoulos, (2020), infer that it is necessary for the selected methodology to involve all stakeholders during the development process. The earlier development methods are Uschold and King's methodology (Uschold and King, 1995), the Toronto Virtual Enterprise methods (Grüninger and Fox, 1995), METHONTOLOGY (Fernandez, 2000), KACTUS and SENSUS (Corcho et al., 2003), On-To-Knowledge methodology (Staab, Studer and Sure, 2003), Ontology Development 101 (Noy & McGuinness, 2001) and NeOn methodology (Baonza, 2010). These methodologies and recent ones published in peer-reviewed articles are reviewed in the ensuing sub-sections.

3.4.2.1. Uschold and King and Toronto Virtual Enterprise Methodology

The oldest ontology construction methods were proposed based on the experience gained during the development of the Enterprise Ontology by Uschold and King (1995) and the TOVE (Toronto Virtual Enterprise) project by Grüninger and Fox (1995). Both methodologies are mainly adopted for business processes and modelling domains. Uschold and King postulated four guidelines for developing ontology; identification of ontology's purpose and scope, building, evaluation, and documentation. However, the TOVE methodology uses scenarios and competency questions that the ontology has to answer to determine its size and concepts. It subsequently builds a logical model of the ontology knowledge base using formal representations (Fernandez, 2000; Grüninger and Fox, 1995).

3.4.2.2. KACTUS and METHONTOLOGY

KACTUS and METHONTOLOGY emerged at the same time in 1996 (Uschold and Grüninger, 1996). KACTUS methodology emphasises on developing ontologies that are specific to certain domain applications. It uses process abstraction (bottom-up) and focuses on ontology re-use and support without vivid descriptions of the development life cycle. However, METHONTOLOGY focuses on detailed development processes. One could either build ontology from scratch or re-use an existing methodology. The structure of METHONTOLOGY allows development of ontology at the knowledge level. It has six vivid steps: specifications and scope, knowledge acquisition, conceptualization, integration of existing ontologies, implementation, and evaluation. It also has unique techniques for carrying out activities in each stage of the development, with its life cycle constructed on evolving prototypes (Fernandez, 2000; Corcho et al., 2003).

3.4.2.3. SENSUS Methodology

SENSUS methodology was proposed in 1997 aimed to use natural language processing to provide a structure for developing machine translators. SENSUS is a product of extraction and information merging from two ontologies (PENMAN upper model and ONTOS then later WordNet) and uses the top-down approach to derive specific concepts from huge ontologies (Corcho et al., 2003). Because SENSUS has more than 50,000 concepts arranged in a hierarchy according to their level of abstraction, researchers (Fernandez, 2000; Corcho et al., 2003) propose identifying "seed" concepts relevant to the domain. The process is repeated until all relevant concepts are obtained and represented in a tree structure with sub-trees and nodes to build the exact domain ontology.

3.4.2.4. On-To-Knowledge Methodology

Staab, Studer, and Sure (2003) developed an On-To-Knowledge methodology to operate on the information of large organisations that are available electronically so that developed ontology can improve the quality of such information. It focuses on ontology knowledge management systems using two orthogonal processes: knowledge processes and knowledge meta-processes. Once the ontology requirements are obtained, capability questions are asked, and an analysis of scenarios is made to identify goals that the knowledge management tools should accomplish (Corcho et al., 2003). The ontology is then implemented. DILIGENT methodology (Pinto, Staab, and Tempich, 2004), is very similar to the On-To-Knowledge procedure, however, it supports collaborative ontology development using distributed settings to enable domain experts to develop ontology. DILIGENT uses distributed environments, which is suitable for users at dispersed locations to share that ontology.

3.4.2.5. Ontology Development 101

Ontology development 101 (Noy & McGuinness, 2001) provides detailed but simple integrated steps for developing ontologies. The development processes integrate efforts whereby the developer determines and defines the domain's scope and chooses what concepts to be re-used from existing ontologies. The developer further determines the

class hierarchy, outlines properties, defines relationships, and eventually creates the ontology class instances. The developer can then develop the ontology in one of the following three steps: The Top-down method (where general classes are identified and developed, then specific subclasses of each subclass are made), the Bottom-up approach (where precise concepts are defined and further grouped into more general and abstract concepts), or a Combination (where concepts are divided and the top-level and specific level concepts are related through the middle level) (Noy and McGuinness, 2001). There is no superior approach among the three. They are chosen and used per the ontology developer's requirements. Researchers, however, argue that this methodology is a guide and not an actual methodology. Nonetheless, it is the most widely trailed, referenced, and applied in research projects in a wide range of domains (Khosrojerdi, Gagnon, & Valverde, 2021).

3.4.2.6. NeOn Methodology

A more recent ontology development process than the ones described above is the NeOn Methodology (Baonza, 2010). The neOn methodology is a scenario-based methodology for developing, re-use, and evaluating ontology in distributed networks. End-users are actively involved (Suárez, 2012). Neon Methodology is most famous for the ontology reuse guidelines it provides. Firstly, it searches contender ontologies that would satisfy the solution. Secondly, it analyses the different re-usable ontologies for usefulness in the ontology network. Thirdly, the most suitable ontology is selected and lastly integrated to create the new ontology. Jimenez et al. (2012) propose using a decision-making analysis system to support the ontology selection process and consider "re-use cost, understandability, integration workload, and reliability" simultaneously.

The neOn methodology improves on previous methods. Since the proposed ontology is intended to serve as a knowledge base for intelligent queries and decision-making, it requires the smart functionalities NeOn methodology can offer. Although highly complex (when the ontology is developed from scratch), the ISO standards of NeOn allow it to be integrated easily into other ontologies for more excellent functionalities (Mattos et al., 2015). All inference tools that are compatible with OWL standards can be used in the NeOn methodology. Likewise, because SPARQL is a NeOn compatible tool, the ontology can execute complex queries (Mattos et al., 2015).

The general workflow of these methodologies is specification, conceptualization, implementation, and evaluation. However, each procedure has its characteristic limitations. An essential downside of the Uschold and King's method is its lack of a meticulous conceptualization process before ontology implementation (Uschold and King, 1995). For the TOVE method, it proposes a broad and vague approach that lacks specific details about how the actions are performed. The TOVE, KACTUS, SENSUS methodologies do not follow a detailed life cycle development process (Corcho et al., 2003). In KACTUS methodology, ontology is developed using an abstraction process from the knowledge base, while in SENSUS, it is built by thinning a huge ontology (Orcho et al., 2003). Similarly, KACTUS and On-To-Knowledge methodologies are highly application dependent while TOVE and SENSUS are semi-application dependent (Uschold and King, 1995). These reasons make the methods unsuitable for this proposed project.

METHONTOLOGY follows a very meticulous development process and is entirely application-independent since the development process is independent of the ontology's final use (Corcho et al., 2003; Corcho et al., 2005). However, it is essential to note that end users are also not involved. Only Ontology Development 101 focuses on the development process' collaborative aspects, which is a fundamental quality for this study (Zeginis et al., 2012). However, end users are only involved during knowledge acquisition but not engaged during the development process. Likewise, researchers (Noy and McGuinness, 2001) argue that ontology development 101 is a guide and not actual methodology.

3.4.3. Other Ontology Re-Engineering Methodologies

International scholars on semantic web and ontology have researched several ontology construction methods in different fields, including the construction industry. These are considered custom engineering methodologies. They may not explicitly define development phases in a systematic way but develop ontologies in an agile, collaborative approach (Kotis, Vouros and Spiliotopoulos, 2020). This section aims to present these published ontologies that provide unique methodologies to their developments. Xu et al. (2010) proposed an ontology construction method that uses relational database

schemas. It extracts data from this database and follows a set of rules to develop ontology. Other researchers (Sameen and Quratulain, 2012) have explored several algorithms to automatically construct ontologies from relational databases observing the accuracy, strengths, and challenges of each algorithm. In another study, researchers proposed another methodology with guidelines on how to publish data as Linked Data. They focus on the re-use of existing linked data (Villazón-Terrazas et al., 2011). Researchers (Wang., 2011) have proposed a text retrieving methodology, which extracts domain knowledge from web pages and builds domain ontology. It extends and analyses and creates a semantic vector representation of this analysis to match the user query and the domain ontology.

In another study, new algorithms to create ontology-based text statistics were proposed (Song et al., 2014). This algorithm obtains ontology concepts from the text, applies classification, and uses an extraction method to obtain domain concepts to develop the ontology. Song et al. (2014) established a skeleton method to create enterprise ontology. Once the business domain is selected, concepts are obtained, coded, evaluated, and documented. They proposed a seven-step method that has been primarily used in other methodologies; determine the scope, consider ontology re-use, list of terms, classes, and relations, define properties, and, lastly, instance classes. Song et al. (2014) believes the Seven Steps method to be the most widely accepted and applied methodology. Isiaq and Osman, (2014) proposed an ontology development methodology that models interdependent applications such that required results are obtained when sophisticated applications with temporal attributes are modelled. A method that uses dependency parsing is used to assist in product selection to help clients identify preferred products, but this uses the user's choice history (Gantayat et al., 2014).

In another study, the uniform process for ontology building (UPON) Lite methodology is designed to suit ontology prototyping and reduce the role of the ontology engineer (De Nicola & Missikoff, 2016). While Upon Lite focuses on knowledge reuse, SAMOD methodology proposed by Peroni, (2017) relies on fresh domain knowledge from domain descriptions. It uses agile, iterative and flexible ontology development approach. The latest version of the SIOC ontology (a popular ontology for representing social communities and content online) is developed based on a custom methodology that is a

collaboration of semantic web developers on the SIOC, data science institute, members of the unit for semantics and Wiki support (Berrueta *et al.*, 2018).

In Kotis, Vouros and Spiliotopoulos, (2020), collaborative and custom ontology engineering methodologies were thoroughly reviewed and evaluated. They conclude that non-collaborative methodologies result in ontologies that are not suitable for reuse and therefore propose that custom methodologies need to be supported by collaboration tool sets for successful development, maintenance and extension of developed ontologies. A similar appraisal done by Spoladore and Pessot, (2022) asserts that good ontologies can be developed by following the context and instructions of the ontology development methodologies and domain expert's contribution even if the developer is not an experienced ontology developer. Although ontology development processes are chosen based on the application's needs, researchers believe that through different use case scenarios, it has been proven that it is technically and scientifically not feasible to prove that one methodology is significantly better than another (Keet, 2018; (Kotis, Vouros and Spiliotopoulos, 2020). The next section reviews ontology engineering tools.

3.4.4. Ontology Engineering Tools

Ontology Alignment Tools - As previously explained, ontology could be developed either from scratch or built by re-using or extending already existing ontology. Numerous ontologies available in the same domain may carry the same concepts to express the same knowledge or with a different intended form of usage and distinct perceptions, so they contradict each other. Many ontologies overlap each other. Therefore, ontology mapping techniques are used to determine correspondence between concepts in different ontologies (Geng *et al.*, 2020; Antonios, Konstantinos and Christos, 2023). For each entity in the first ontology, concepts with the same or closest meaning are found in the second ontology. Entity x from ontology O1 and entity y from ontology O2 could be equal, one more general than the other or one more specific than the other, or disjoint. This correspondence is called an alignment (Geng *et al.*, 2020).

A series of ontology alignment tools have been developed and presented in literature. Chimaera (McGuinness et al., 2000) is an interactive tool designed to analyse ontologies to be matched; it checks for linguistic similarities and merges the ontologies if matches are found. Else, the user urges for a different action. Chimera, however, provides little automatic merging support. Noy and Musen have also established a series of ontology mapping tools including SMART (Noy and Musen 1999), PROMPT (Noy and Musen 2000), and PROMPTDIFF (Noy and Musen 2002). PROMPT is an interactive tool that allows the user to supply information about how their concepts are related and determine the next action. An extension PROMPTDIFF uses similarity matches between concepts to align and merge processes. Bing et al. (2014) profoundly argue that SMART is more specialised because it can automatically merge concepts of different ontologies without much user involvement. Nonetheless, this ability is minimal. Noy and Musen (2014) and Aberer et al. (2015) declare PROMPT is a more reliable tool because it fully involves user assessments. It creates a list of semantically similar class-matches and outlines the relationship between concepts. It further maps the ontologies together and provides a standard layer for information interchange. PROMPT is the most popular ontology alignment tool. It is also compatible with Protégé-OWL and therefore preferred for use for this research.

Reasoners - the semantic web has a strong connection to artificial intelligence through its ability to provide knowledge reasoning (Pascal, 2021). Algorithms developed to overcome the limitations of entailment and validation purposes are available in software tools known as reasoners. A reasoner uses first-order predicate logic to check if the ontology created is consistent and contains explicit logical knowledge (Antonios, Konstantinos and Christos, 2023). Reasoners infer knowledge-based models (Khosrojerdi, Gagnon, & Valverde, 2021). It responds to data retrieval and conceptual queries about the domain's structure, thereby making ontology languages efficient. Additionally, reasoners check anomalies and provide means to reason through ontology.

The most popular and widely used reasoners are FaCT++ (Tsarkov & Horrocks 2006), Pellet (Sirin et al. 2007), and HermiT (Horrocks, 2011). FaCT++ is a reasoner that supports OWL DL-based ontology languages. HermiT reasoner supports specialised reasoning services such as property and class classifications, SPARQL queries, and

description graphs. Pellet, an OWL 2 reasoner (Clark and Parsia, 2015; Dwivedi, Tirkey and Rath, 2016), is a leading choice for ontology-based data management applications for systems that require optimum OWL DL reasoning because it offers conjunction query answering and standard incremental reasoning (Khosrojerdi, Gagnon, & Valverde, 2021). HermiT (2015) has stated that HermiT is the first reasoner to identify ontologies that have demonstrated complexity for systems to handle, as it offers much more effective reasoning than previous reasoning algorithms have. To support that, researchers (Horrocks et al., 2014a; Alaya, Yahia, and Lamoille, 2016) conducted a series of tests to compare HermiT with the state-of-the-art reasoners, Pellet, and FaCT++. According to the authors, although HermiT does not super-exceed the other reasoners on all ontologies, it was much more robust, more efficient, and handled more complex ontologies. HermiT recorded the highest success and the lowest unexpected failures. However, others argue that because pellet fulfils the state-of-the-art optimization techniques in description logic, it provides greater support for reasoning with individuals (Khosrojerdi, Gagnon, & Valverde, 2021). Nonetheless, these reasoners are compatible with protégé-OWL editing software. However, the proposed research does not require support for complexity. Therefore, checks for anomalies would be done using both HermiT and FaCT++ plug-in. Hermit is highly effective, and FaCT++'s compatibility and incorporation in protégé-OWL is a major appealing factor.

Ontology Development Environments - Ontology editors have built-in inference engines and information extraction facilities designed to manage ontologies. Editors have essential features such as the ability to import and export foreign knowledge representation languages. For complete reasoning and consistency checking of an ontology, a reasoning engine must be run within an editor (Panzarella, Veltri and Alcaro, 2023). There are several widely used ontology development softwares. The Neon toolkit is an open-source OWL-supported tool with import, re-use, and management mechanisms for ontology and is especially suited for heavyweight projects (Erdmann and Waterfeld, 2012). OBO-Edit and SWOOP are both Java-based tools that provide management services for ontology (Day-Richter, Harris and Haendal, 2007). However, SWOOP still lacks a project website (Kalyanpur et al., 2006). Other projects have showcased a unique ontology editor called the Aspect-Oriented Visual Ontology Editor (AVOnEd), arguing that while most of the editors explained above represent ontology,

they cannot describe its distinctive aspects (Gesterkamp, Rebstadt, and Mertens, 2017). The AVOnEd builds complex ontology and focuses on one part of it for user visualisation, thereby simplifying it. Another functionality that makes AVOnEd stand out from the other state-of-the-art editors is its edit-time consistency checking feature (Hallay et al., 2017). However, although it supports both RDF import and export, it is compatible with only OWL-Lite (Gesterkamp, Rebstadt, and Mertens, 2017). In Preventis and Petrakis, (2021), a collaborative ontology editor named CLONE is proposed for applications that run in the cloud. Clone provides real-time collaborative features including team conversations, editing and authorizations.

However, since OWL is the preferred language for this research, a compatible editor is required. Protégé-OWL is the most popular and most widely used editor for OWL knowledge-based solutions in wide-ranging domains (protégé, 2015). It gives users a comprehensive understanding of OWL ontology development structure (Panzarella, Veltri and Alcaro, 2023). It is a free software. It comes with several plug-ins to serve specific purposes. It is exceedingly flexible for swift prototyping and uses other smart tools for exceptional, intelligent application development. Protégé-OWL's compatibility with OWL and OWL ontology validators makes it most suitable for this research.

Ontology development most often entails writing computer programs within an application software for systematic design, development, and maintenance of the ontology. Furthermore, ontology could be used alongside software products or applications to provide comprehensive solutions to domain problems. Therefore ontology development is interlined with software engineering. The next section reviews some of the software development methodologies used in practice and discussed in literature.

3.5. Software Engineering

A piece of software is developed as a result of executable programs that are well documented and are run by machines. Development of any software involves a group of stages a software engineer follows consecutively from obtaining the user requirements

to delivery of a software that satisfies those requirements. These development phases are what makes up software engineering. Several software engineering methodologies exist in literature with varying approaches of the development process and the involvement of the team (Javed, Khan and Tubbassum, 2019). Each has unique features, advantages, and limitations that are only realised by the software engineer as they develop different softwares for different solutions (Govil and Sharma, 2022). The methodology that suits one project may not necessarily work for another since each software has different requirements. Other factors often considered when selecting a software engineering methodology to adopt include the scope and functionalities of the project, team size and budget (Govil and Sharma, 2022). Therefore, the methodology to be adopted varies from one software or project to another. The following section provides a brief synopsis of the most widely applied software development methodologies.

Waterfall is one of the most traditional methodologies, often considered as “old school”. Although it is considered outdated, a clear understanding of its structure helps developers appreciate the flexibility of the modern methodologies. Waterfall approach has a sequence of rigid steps that do not alter once it has begun (Saeedi & Visvizi, 2021). From the onset, it requires meticulous documentation of user requirements and scope. The design and implementation stages are carried out religiously to the end. Each phase is uninterruptedly dependent on the output of the previous phase and one phase can hold back the entire development process (Javed, Khan and Tubbassum, 2019). Waterfall processes are neither iterative nor reversible (Saeedi & Visvizi, 2021). Should there be any changes in requirements or errors during development, the waterfall approach requires a complete restart. It also does not provide options for developers to know what the outcome of the project would possibly be; therefore, it can only be applied by experienced plan-driven software development teams who have clear understanding of the project at hand and clear definitions of functional and non-functional requirements. Otherwise, the more modern methodologies should be considered.

Agile methodology was developed as a solution to rigid and highly structured methodologies such as waterfall. As the name implies, agile ensures flexibility in

software development (Javed, Khan and Tubbassum, 2019). It accommodates changes and allows recurrent alterations (Govil and Sharma, 2022). In an agile approach, the development team works towards delivering certain parts of a working software in sprints within an agreed duration of time (Zima, 2015). It is known for its dynamic prioritisation in its continuous iterative cycle (Javed, Khan and Tubbassum, 2019). Researchers complain that software developed using agile approaches are often poorly documented. However, it is still preferred for its ability to detect and resolve errors without having to start all over (Pocsova et al., 2020). Through a series of feedbacks, software is improved in iterations and through continuous dispensing of tested and working deliverables (Saeedi & Visvizi, 2021). Each release is an improved version of the preceding version and is updated based on changes in user requirements or upgrade in technology used during the development process.

The flexible and iterative nature of agile methodology led to the development of other software development approaches. Scrum methodology is one of the most popular agile frameworks used in practice (Saeedi & Visvizi, 2021). It is known for its fixed-length iterative delivery of software called sprints (Pocsova et al., 2020). It is best suited for large projects that require deliberate progression. While scrum is best suited for disciplined developers, extreme programming (XP) methodology is applied for high-risk projects with very limited duration (Saeedi & Visvizi, 2021). It is known for its emphasis on customer satisfaction which is achieved through as much feedback as possible. Another approach known as feature driven development (FDD) incorporates an agile iterative goal of delivering working software one feature at a time until the complete model is developed. FDD's major limitation is that it requires a large team of developers to be able to handle the features effectively and on schedule.

Differences between the development methodologies can be found in literature. In Ali and Lai (2021), authors systematically reviewed over 200 studies to compare approaches used in multi-site software development. In Saeedi & Visvizi, (2021), the authors adopt scrum over waterfall methodology and believe that the scrum approach provides students with close-to-real world scenarios of software development projects. In Parizi at al., (2022) authors studied the goal of design thinking within the approaches of software development. Javed, Khan and Tubbassum, (2019) critically analysed the

suitability of agile methodology to waterfall approach and presented advantages and limitations of each as regards to project risk management. Govil and Sharma, (2022) conducted a critical analysis and ranking of the different software methodologies through mathematical fuzzy TOPSI method and ranked agile as the 1st and waterfall as the last in terms of effectiveness.

In research, a piece of software is often developed to test the findings of the research. Most often, this software is developed in an iterative manner in the form of a prototype. A prototype is broad in its definition. In software development, it can be a working model of a product or IT solution to demonstrate concepts and results (Kent *et al.*, 2021). In research studies, prototypes are developed as a proof of concept of findings in the research process. Most prototypes are not intended for deployment by the developer but to describe how the actual product should work to obtain feedback on design methods through testing (Okay, Eray and Eray, 2022). It is an experimental system and therefore can have different levels of sophistication from a simple sketch of an idea to a working prototype that imitates the actual software. However, some prototypes evolve into the real system. In research studies that involve development of knowledge models, prototype development can be an iterative process applied during any stage of the development. This makes it suitable for application within the context of this study. Agile ontology development involves a continuous process of knowledge specification and refinement as they are elicited. Therefore, since prototyping is done in an iterative manner, the ontology engineer can introduce new facts and requirements into the ontology as they emerge until a well structured prototype is derived.

An ontology is developed as a prototype in Clemente *et al.*, (2022) to serve as a base for an adaptive recommender system in the area of E-learning. This prototype is developed based on competence and modular nature of ontologies. In the photovoltaic systems domain, Okay, Eray and Eray, (2022) developed a prototype system that designs and manages batteries used in a grid-connected PV system. This prototype imitates energy generation from a PV system designed by the developers and demonstrates a consumption profile for retired and working household residents. The energy management function in the prototype system investigates and presents how energy is taken from the grid when the battery charge is insufficient. It further studied how excess

energy stored by a fully charged battery is sold to the grid. Further details on applications of prototyping, its benefits and review of literature can be found in Kent *et al.*, (2021).

3.6. Ontology Evaluation

With a growing adoption of ontologies, it has become very imperative to assess the quality and correctness of developed ontologies. Since the ontology is an approximate specification of a domain, its evaluation should investigate how close to the real-world the knowledge model is to ensure that the correct model of the domain is created (Fernández-Izquierdo and García-Castro, 2021). Evaluation involves two main processes: verification and validation of ontology. Verification means performing syntactic checking of the knowledge base to guarantee correctness, while validation consists of ontology valuation to ensure it represents/models precisely what it is created for (Staab and Studer, 2010). An effective evaluation technique ensures that developed ontology is semantically and syntactically correct and the ontology accomplishes its intended purpose. The challenge is, which verification and validation techniques should be adopted to achieve this aim?

Ontology verification can be performed in two (2) ways. Firstly, it is often recommended that if an ontology is built from scratch, a domain expert could perform semantic and syntactic checks to see if requirements and competency questions are implemented correctly (Raad and Cruz, 2015). However, this is often time consuming. Secondly, if the ontology is built from re-using or extending existing ontologies, it could be evaluated using ontology mapping tools (Noy and Musen, 2002). The selected ontology-mapping tool for this research, PROMPT, provides support and techniques for semantic verification of ontology (Noy and Musen, 2014). PROMPT analysis concepts from all re-used ontologies. Where a concept has been adopted entirely, there would be no requirement for semantic verification, however, when the concept is only partially re-used, the new knowledge must be verified.

Once the ontology is semantically verified, it is important to also check if the ontology created is consistent and contains explicit knowledge. Syntactic verification checks the

hierarchy, instances, and consistency of the ontology. Semantic web reasoners are often used to check for anomalies in the ontology. These tools such as FaCT++ and pellet have plug-ins inbuilt within protégé (Tsarkov, & Horrocks; 2016). Furthermore, it is also recommended to verify the compliance of the ontology with the required language, because for instance, if an ontology is not OWL compliant, it will not support OWL reasoning techniques. Ontology validation is mainly done through case studies to determine if the ontology achieves the objectives it has been developed for. Responses and criticisms should be obtained from domain experts to help researchers identify misunderstandings in requirements specifications, and improve the correctness and quality of the knowledge captured (Fernández-Izquierdo and García-Castro, 2021).

Several evaluation approaches have been proposed (that have been tested and trusted) by the earliest specialists in ontology modelling as well as the newest experts. Gruber (1995) proposed a criterion that checks for ontology clarity, coherence, and extendibility. From his classifications, an ontology should effectively communicate the context of concept definitions, it should sanction inconsistent and incomplete inferences and it should offer a robust conceptual foundation capable of solving current domain problems and extended to develop future domain solutions (Raad and Cruz, 2015). Gomez-Perez (2001) approach checks for what an ontology is defined correctly, what it defines incorrectly and what it does not define at all. His research has defined a set of criteria that checks ontology consistency and evaluates whether it is feasible to obtain contradictory results from valid input definitions. Similarly, a high-quality ontology should ensure sensitivity whereby changes in or additions to definition of concepts and properties should not affect the correctness of already existing concepts (Gomez-Perez, 2001). Some key discussions on semantic verification and validation can be found in Noy and McGuinness (2001), Gomez-Perez (2001), Noy and Musen (2002), Antoniou et al., 2012, Baumeister and Seipel, 2005), Hlomani & Stacey, (2014), Raad and Cruz, (2015), Bandeira, et al., (2017), Sun et al., (2020) Fernández-Izquierdo and García-Castro, (2021).

3.7. Existing Domain Ontologies for Possible Re-Use

Ontology development can be based on standards and technologies that rely on collaborative development and that have been thoroughly validated. Existing knowledge models were examined through a thorough review of literature and knowledge databases. Ultimately, several existing knowledge models were identified, and a few of these domain ontologies were ascertained for possible re-use.

This research identifies several existing models in the domain of sustainable building technology products and presents them in four categories: solar irradiance, energy consumption and analysis, building structural elements, and photovoltaic system technology. Firstly, the deployment of photovoltaic (PV) systems technology requires meteorological data to obtain data on solar irradiance. Most simulations done to predict the behaviour of installed PV involve assumptions of weather and climate conditions. This importance in the representation of meteorological data led to the development of an ontology for solar forecasting (Kantamneni and Brown, 2018) and another OntoWind ontology (Kucuk & Kucuk, 2018). They provide a formal representation of solar irradiance as used in designing photovoltaic systems.

Secondly, it is imperative to determine the building's energy consumption onto which PV systems will be installed. Therefore, this research studied existing building energy management models. Researchers bring forward coherent data on energy targets, new findings, and the latest approaches to the deployment of renewable energy systems. Different studies propose semantic structuring of these data. Several ontologies have been developed to model power consumption in homes. These include Energy Resource Ontology (Kofler, Reinisch and Kastner, 2012); SEMANCO Ontology (Madrazo, Sicilia and Gamboa, 2012); and the Electricity Markets Ontology (Santos, et al., 2016). Another Fiesta-IoT ontology model (Agarwal, et al., 2018) uniquely defines energy consumption taxonomy that aligns with and increases interoperability amongst several internet of things (IoT) ontologies. It enhances privacy and describes smart-home appliances. It is most appreciated for its description of demand-side energy management services. The Smart Applications REference (SAREF) ontology (SAREF, 2020) provides standardised

extensions for buildings (SAREF4bldg) and smart energy (SAREF4ener) (Daniele, Hartog, and Roes, 2015; Gilani, Quinn and McArthur, 2020; SAREF, 2020). It enables interoperability between smart buildings technologies. Another comprehensive model identified is the Open Energy Ontology (OEO) (OpenEnergyFamily, 2020). This ontology analyses energy systems. It handles both energy supply and consumption. The main target of the researchers who developed OEO was to obtain a shared conceptualization of energy systems data and analysis and share it across the domain for different applications. This model is thorough and inclusive. However, for the context of this study, a comprehensive representation of the electrical aspects of PV systems beyond household electrical appliances is not required.

The third category considered is ontologies that represent construction models that define building structural elements. One of the ontologies considered within this category is the Energy-Aware Systems (SEAS) ontology (Lefrancois, 2017). It describes the relations and properties of all physical aspects of a building. Conversely, one of the most popular ontologies within the construction domain is the ifcOWL ontology developed by Pauwels and Terkaj (Pauwels and Terkaj, 2016) and a group of many contributors as part of the BuildingSMART Linked Data Working Group and W3C Community Group on Linked Building Data (W3C, 2020b). Because it provides a semantic representation of the IFC Schema, researchers and practitioners can represent building data using state-of-art technologies. The graph-like model (RDF) and other technologies within the semantic web stack allow building data to be easily linked to product manufacturer data, classification structures, sensor data, meteorological data, and so many more. Linked building data, effective data management, and exchange in the construction domain have never been more straightforward. Another existing knowledge model is the Building Element Ontology (Pauwels, 2018). It is an extension of IfcBuildingElement, which contains a taxonomy of the building element classes available in that IFC specification. The IfcBuildingElement is an abstract specification, and it cannot be instantiated. Therefore, the ontology provides a formal definition of these structural elements (such as walls, doors, columns, roofs, and others). This ontology has not been widely adopted.

The Building Product Ontology (BPO) (Wagner and Rüppel, 2019) provides methods that thoroughly describe building products and their properties without type descriptions. The ontology is aligned to SEAS' (Lefrancois, 2017) and W3C's schema.org ontology (Schema, 2021). However, it provides a semantic description of products without geometric and material descriptions. Furthermore, one of the most recent ontologies that represent construction models is the BOT ontology. The Building Topology Ontology (BOT) (Rasmussen, et al., 2021) describes a building's most fundamental topological concepts. The developers understand situations whereby products or technologies (such as PV technology) need to be described in the context of a building onto which it is installed. The BOT gives a minimal yet extendable description of concepts as they are related to a building. Although it has recently been published under the W3C community contributor conditions, it is not yet a W3C standard.

The fourth category includes existing knowledge models that have been identified within the domain of photovoltaic system technologies. The Photovoltaic Technology Ontology Systems (PV-Tons) (Abanda, Tah, and Duce, 2013) manages knowledge on PV systems and applies several semantic web technologies to enable users to select PV components. A second model recognized is an ontology for solar soft costs (SSCO) (Beck & Rai, 2019). This ontology accounts for non-hardware expenses of developing and installing photovoltaic systems such as supply chains, permit fees, inspections, design, as many more. Soft costs fall under diverse categories and vastly vary in terms of purposes. Therefore, the ontology is aimed towards improved definitions and standardizations.

3.8. Applications of Semantic Web in Construction

Applications of the semantic web and its technologies have been thoroughly researched over decades in several domains. Several domain applications that solve complex problems are being developed thanks to the semantic web's capabilities of effective data integration, interoperability, and semantic reasoning. In the literature review, semantic web applications are found in medicine, business and e-commerce, engineering, education, sustainable development, and so many more. This study will not discuss the thousands of semantic web applications available in those fields; however, we would focus on applications in the construction domain. This section discusses applications of

the semantic web in project management (Huang et al. 2020), Building Energy Performance (BEP) (Hu *et al.*, 2021), Smart Homes (Antonios, Konstantinos and Christos, 2023), cost estimation (Abanda, Foguem and Tah, 2017), product manufacturing (Huang, et al., 2020), Building Information Modelling (BIM) (Petrova et al., 2019), and Industry Foundation Classes (IFC) (Pauwels & Terkaj, 2016).

An appraisal shows that the bulk of existing semantic web applications in construction are apparent during the project management phase of the construction project (Pauwels et al., 2017). Large volumes of data are identified, specified, and exchanged during project management. Since semantic web has information management capabilities, ontologies are therefore developed to enable effective exchange of unambiguous data amongst collaborators (surveyors, engineers) that handle different phrases (design, operations) of construction through a structured format without loss of data (Huang et al. 2020). These applications allow real-time knowledge exchange making it possible for alterations to be easily made, thereby resolving interoperability concerns in project management.

In Building Energy Performance (BEP), models are created by collecting heterogeneous data from different sources, including energy tariffs, meteorological databases, and other building product repositories (Costa et al., 2020). Semantic web provides a framework for BEP simulation models to be built. The development of formal annotations of BEP models results in a reusable ontology. In the SEMANCO project, energy performance analysis is done using linked data (comprising meteorological data, census data, building data) (Madrado, Sicilia and Gamboa, 2012). Hippolyte et al. (2018) applied semantic web to create a communication link between energy performance systems and the local energy monitoring systems. In another application of semantic web in BEP, data is obtained from Energy Conversion Measures (ECMs) and converted into RDF, upon which a web application was developed (Costa et al., 2020). In this application, semantic web technologies were used to enable large-scale simulation of the energy performance of buildings.

Gilani, Quinn and McArthur, (2020) presented a thorough review of ontologies developed within the smart and ongoing commissioning domain. In a more recent

application, energy performance assessment is conducted using OWL ontologies, RDF instances and inference rules to perform semantic reasoning on building domain knowledge (Hu *et al.*, 2021). This application of semantic web technologies aids building managers with insightful assessments to improve their energy efficiency. Furthermore, the state-of-the-art approach to energy management in buildings is the use of smart homes applications. Most smart home applications are sensor-based Internet of Things (IoT) applications. Researchers (Hontl & Abonyi, 2019) confess that interoperability is one of the biggest challenges faced in developing IoT solutions, therefore relying on semantic web and its technologies to guarantee interoperability. A recent appraisal of the use of semantic web technologies to enhance interoperability in the domain of smart cities and internet of things can be found in Antonios, Konstantinos and Christos, (2023).

In the domain of Sustainable Building Technology Products (SBTPs), semantic web and its technologies have been applied to resolve different domain problems. A review of semantic web applications from the year 2000 to 2013 presented by Abanda, Tah, and Keivani (2013) highlighted a developing shift of semantic web adoption from traditional construction to sustainable construction applications. Researchers have presented approaches to link BIM to Linked Data in order to promote building sustainability (Djuedja *et al.*, 2019). In another study, authors attempted to use semantic web to monitor solar modules' performance in a solar plant (Orlandi *et al.*, 2019). They provided a review of methodologies used to obtain PV data and analysed the challenges observed. Subsequently, an RDF-like framework was developed to capture weather records and PV data from the solar plants linked to other semantically obtained data. In Khosrojerdi, Gagnon, & Valverde (2021), an ontology is developed to represent concepts of PV associated with MPPT-based controllers. This ontological model facilitates decision support to non-technical users through SWRL. In a recent study, authors propose a flexible linked data approach called linked building product data (Wagner *et al.*, 2022). This is implemented alongside an ontology for building products (such as building integrated PV modules) that support semantic reasoning and queries.

Applications of semantic web in building information modelling (BIM) and industry foundation classes are provided in the next chapter.

3.9. Conclusion

This chapter presented the inefficiencies and limitations of the current web in representing and processing information. To overcome the weaknesses discovered, the semantic web and its technologies were explored to determine their potential to model knowledge on sustainable building technology products (SBTPs). Due to the capability of ontology to develop a knowledge base that provides formal definition and representation of concepts and their relationships, the chapter discussed ontological engineering methodologies and tools to explore how the knowledge base can provide decision making support in the selected problem domain. Existing domain ontologies in the domain of sustainable building technology products (SBTPs) that can be potentially re-used in this study were outlined. The chapter concludes with applications of semantic web technologies in different fields in construction including SBTPs. Applications of semantic web in building information modelling (BIM) and industry foundation classes are provided in the next chapter after a review of literature.

4. Overview of Building Information Modelling, Industry Foundation Classes And Applications of Semantic Web

4.1. Introduction

The choice to review BIM in this research is due to its relevance and connection to IFC and semantic web technologies. The importance of BIM in the construction industry is comprehensive and it is seen amongst BIM experts and industry practitioners (Hamil 2021; Lee, 2022; Ferdosi et al., 2022; Parsamehr *et al.*, 2022; Autodesk, 2022). BIM offers solutions to common challenges in the construction industry, such as cost overrun, productivity loss, and labour defects. However, with regards to this study, BIM is relevant for two main reasons. Firstly, one of the objectives of this research is to provide decision support to sustainable building technologies. BIM and its associated data structures and tools have the ability to enhance decision making at every stage of a construction project, including the choice of sustainable technology to acquire. Hence, it is essential to apply BIM-based research and solutions to provide decision support in the domain of this study. An oracle construction and engineering report confirms BIM's relevance in the area of decision-making and collaboration in the construction domain (Bell, 2022). A survey was conducted as part of buildingSMART virtual summit where organisations were asked how BIM has helped their organisation the most. About 71% of respondents attest to improvement in decision-support (Bell, 2022).

Secondly, one of the most critical and consistent challenges in the construction industry has been interoperability. Initial interoperability concerns led to the formation of the International Alliance for interoperability (IAI), now known as the Building Smart International. The efforts of the buildingSMART group evolved in the development of BIM. To develop a practical software solution in the construction domain, interoperability conditions must be met. It entails use of a standard domain data model that is implementable and can be deployed. IFC, being a BIM multi-platform standard for representing building products, is hence deemed fit to be used as the standard data schema specification to represent built assets within the photovoltaic system domain. This study utilises IFC's functionality as a product classification library and taxonomy, to

describe and represent detailed products' data and effectively exchange and re-use required building data models.

The goals of IFC overlap with the purpose of semantic web technologies to support data interoperability, distributed data representation and application of reusable tools (Pauwels and Terkaj, 2016). This chapter provides an overview of BIM, its associated standards, and software tools in section 4.2. BIM's tremendous impact in the construction industry is visible through the increase in its adoption as provided in section 4.2.1. An overview of IFC is given in 4.3 with descriptions of its different formats, specifications, and data schemas in sections 4.3.1 and 4.3.2, respectively. Overlapping goals of BIM, IFC and the semantic web are discussed in 4.4 and this forms the rationale for the integration of IFC and Semantic Web as discussed in 4.4.1 and the future of IFC in 4.4.2. Furthermore, the logic behind the adoption of IFC in the context of this research is provided in 4.5 while the applications of semantic web in the construction domain are presented in 4.5.

4.2. Overview of Building Information Models (BIM)

BIM is often incorrectly understood as the software for construction purposes. However, it is neither a software nor a file format. It goes beyond that. A definition of BIM by the National BIM Standards (NBIMS) (Construction, 2021) states, "a BIM is a digital representation of physical and functional characteristics of a facility..." (Construction, 2021). Abanda et al., (2015) defines BIM as a process facilitated by BIM software systems and driven by people. This coincides with definitions provided by the Construction Industry Council BIM protocol (CIC, 2018). However, the best definition of BIM and its correlated data structures is from the ISO 19650 international standards. According to the ISO 19650, BIM is a process for creating and managing an entire construction project (Hamil, 2021; BSI, 2022). A digital description of all aspects of the built asset is created using relevant BIM tools. All details related to the project, such as structural elements, equipment, materials, etc., are connected and represented in a digital archive in the form of information-rich 3D models and shared amongst all the project's participants. It is extended to cover all phases of a project life cycle (Hamil, 2021; Fargnoli and Lombardi, 2020). The complete building can be well envisioned.

The process of modelling construction information entails complex processes such as 3D modelling, designing, project scheduling, and many more. This process of managing information over the entire life cycle of a built asset has improved construction processes that were complicated, uncontrollable, time and cost-ineffective over the years. A few examples of other benefits of BIM adoption include enhanced visualisation, automated design checks, risk detection, faster project delivery, and decision support (Cha and Jiang, 2020). The organisation and digitization of information on construction projects is guided by available BIM standards such as the ISO 19650, the UK BIM Framework and the UK 1192 standards (BSI, 2022). The principles of these standards are aligned. Details on the ISO 19650 series of standards that involve digitization of building information including modelling concepts and principles, operational phase of assets, approach to information management, modelling information exchange, and modelling delivery phase of assets can all be obtained from the British Standards Institution (BSI) (2022). To facilitate this process, appropriate software technologies for digitising building information are used.

A global leader in the development of highly proficient BIM software solutions for architecture, engineering and construction is Autodesk (Autodesk, 2022). Some of the BIM tools developed by Autodesk are Revit and Naviswork. Revit is one of the most popular BIM softwares used in practice. It offers a smart approach to all stages of a project's life cycle by using models to solve architectural and design problems (Revizto, 2022). It is considered a 4D BIM. Naviswork is different from Revit through its ability to offer review and coordination solutions to professionals (Autodesk, 2022). It identifies and resolves clash problems from the on-set and can predict the outcome of the construction project before it even begins. However both Revit and Naviswork can combine their models with models from other Autodesk BIM solutions. Another tool called Revizto, is known for its platform that integrates 3D and 2D workflows (Revizto, 2022). It facilitates a collaborative environment with synergy for 2D and 3D models where accurate data is available to all stakeholders of the project. Other BIM solutions include ArchiCAD (best used for design and modelling processes), Midas Gen (which is proficient in building management), Buildertrend (most suitable for remodelers) and so many others. Abanda et al., (2015) provides a holistic appraisal of BIM software systems being used in managing construction project information with details of their unique

specifications, imports, outputs and available plugins. Further reviews of the top BIM tools are available at Revizto (2022) and SoftwareAdvice (2022).

One of the main objectives of BIM and its application in construction is motivated by the target to solve ongoing interoperability concerns in the domain (Wagner *et al.*, 2022). Construction project participants (architects, surveyors, engineers, etc.) use computer applications individually and separately to insert, update and modify building information that they may be working on or information that may have been created and sent by one or more other collaborators. Information is endlessly exchanged amongst collaborators (using software applications) through a project's entire lifecycle. It is crucial to use compatible software technologies so that information shared flows efficiently without re-creation. This makes interoperability a crucial part of BIM because BIM data needs to be exchanged to be beneficial. This ability to share important information on the entire construction project with all the stakeholders involved makes BIM an eminent collaborative tool. For this reason, the adoption of BIM is highly relevant to decision support.

Emerging BIM applications in construction have enhanced the definition, integration, and semantic analysis of building data models. Instead of sharing a large volume of unstructured data for simulation, design and construction, facility management, operations, and so many more, significant progress has been made towards developing clear semantic structures for domains. Experts have become intent on sharing the "semantics" of a building. Similarly, imagine having a single model that manages the heterogeneity of complete building data provided by different stakeholders across the domain. BIM makes this possible through its standard format of representation and re-use of existing building data. The ensuing section discusses some of the many applications of BIM in the construction domain.

4.2.1. Applications of BIM in Construction Domain

BIM's tremendous impact on AEC is visible through the rising increase in adoption of BIM applications. Several studies have proposed the use of BIM in different fields of construction to enhance efficiency and quality of projects. In the context of

construction/occupational safety, BIM tools are used to augment safety training and education, safety improvements, and safety management (Fargnoli and Lombardi, 2020). Applications of BIM can also be seen in other activities related to building, such as maintenance of building assets (Antwi-Afari *et al.*, 2018), facility management (Hosseini *et al.*, 2018), enhancing the sustainability of building (Olawumi and Chan, 2018), facility and building equipment management (Fargnoli *et al.*, 2019). In another study, researchers adopted BIM to propose and test the quality of building elements and enhance the complete framework's quality management (Mirshokraei, De Gaetani and Migliaccio, 2019). Likewise, BIM is used to automatically generate logic chains for mechanical/electrical/plumbing (MEP) systems of a building and compare them with actual projects (Xiao, Li, and Hu, 2019).

Several researchers have deliberated on BIM-based solutions to fire safety. Jung, Cha, and Jiang, (2020) proposes an automated system to use three-dimensional visualisation to capture all relevant information for lessening fire disasters. Khan *et al.*, (2020) uses a multi-agent-based approach to visualise and simulate fire safety planning. In another study, a conceptual strategy is proposed to capture and enable exchange of information between Fire Safety Engineering (FSE) tools and BIM (Siddiqui *et al.*, 2021). The researchers aimed to join the buildingSMART international collaboration to enhance the IFC model from FSE perspective. In cost management, Liao, (2021) developed a cost management practice BIM model that proposes countermeasures and guides the entire project cost assessments. In another study, researchers provided detailed analysis on the application of BIM to resolve all cost-related challenges that may be encountered at various stages of a project life cycle (Tang and Liu, 2022). Latest research studies the applicability of BIM alongside geographic information system (GIS) to develop digital twin technologies (Xia *et al.*, 2022).

In Parsamehr *et al.*, (2022), after a review and assessment of construction management challenges was undertaken, BIM-based solutions were proposed for scheduling, cost, quality and safety management. Ferdosi *et al.*, (2022) conducted a thorough state-of-the-art review of recent applications of BIM in key sustainable construction domains including green buildings and energy analysis. Further in depth appraisal of advanced adoption of BIM in construction can be studied in Lee (2022), Cao,

Kamaruzzaman and Aziz, (2022). In the context of this study, much interest focuses on BIM's ability to drive enhanced decision making at every stage of a construction project. With the help of BIM software tools, researchers are able to develop decision models that can be populated and queried in order to evaluate performance parameters of buildings. Haghiri et al., (2021) developed a BIM-based decision support tool that enhances design to fabrication process. In another study, a BIM-based decision-making tool was developed to aid professionals in addressing challenges faced in building condition assessment (Alavi, Bortolini and Forca, 2022). Other researchers have looked towards developing a 4D BIM-based decision support system to aid in scheduling, monitoring risk activities and controlling duration of the construction project (Ayman, Mahfouz and Alhady, 2022).

BIM's multi-platform standard for representing building products in data models is IFC. The IFC standard is specified in the ISO 16739-1:2018 standard for BIM for its ability to facilitate information sharing and exchange among heterogeneous software applications used by several professionals in the AEC domain (Hamil, 2021). Numerous BIM organisations support adoption of IFC. Therefore over 100 BIM software packages that are IFC-compliant have been developed (BuildingSMART, 2022e). IFC is often considered BIM's main hope to achieve complete interoperability. Therefore it is imperative to apply IFC-based BIM research and solutions to provide decision support in the domain of this study. An IFC-based BIM solution would certify that a standard data model that is implementable is developed to support decision making for sustainable building technologies and photovoltaic systems.

The following section provides an overview of IFC, its specifications and its connection to semantic web. Further rationale for its adoption in this study is provided later in this chapter.

4.3. Overview of Industry Foundation Classes (IFC)

IFC is adopted in the context of this study as the standard data schema specification used to represent built assets within the photovoltaic system domain. This study utilises IFC's functionality as a product classification library and taxonomy, to describe and represent detailed products' data. The goals of IFC overlap with the purpose of semantic

web technologies to support data interoperability, distributed data representation and management and application of reusable tools (Pauwels and Terkaj, 2016).

IFC research dates to the early 1990s but was standardised by a buildingSMART open international standard of the International Organisation for Standardisation ISO 16739-1 (Jetlund, et al., 2020). It is currently in version 4.1, and several extensions are under development. Since its emergence, integrated methods of sharing construction information have been developed, thereby urging researchers to create more IFC-based applications (solutions). The ensuing sub-sections discuss IFC formats, specifications, and data schemas that support the development of tools used to export, import, and transmit building data in IFC format.

4.3.1. IFC Formats and Schema Specifications

IFC defines building elements (walls, roofs), geometric features and abstract concepts (such as space and costs) as models using different file formats and schema specifications. These formats and specifications support automated modelling (sharing and re-use) of BIM data across domain applications.

Naturally, an IFC model works like a reference for information exchange between software applications since numerous BIM application software systems support effective data exchange. Primarily, building experts import and export data models to IFC tools using the IFC STEP Physical File Format (IFC-SPF) of the ISO 10303-21 (BuildingSMART, 2022a). However, when experts wish to exchange building data in eXtensible Markup Language (XML) format, the XML Schema Definition (XSD) of ISO 1030-28 is adopted (BuildingSMART, 2022a). An ifcXML file contains the same IFC data. This alternative method of representation gives enhanced benefits to XML adopted software tools that have been primarily used in another domain to support interoperability. IFC data models may also be compressed within a ZIP file format. Nonetheless, the compressed files may be represented in either SPF or XML depending on the file format deemed most compatible to the tools or software applications being used by the user. No file format is "the best", each has benefits and tradeoffs of scalability and software support. However, IFC-SPF is the most widely adopted format in practice for two reasons. Firstly, it is the most compact format that can be read as text

(BuildingSMART, 2022a). Secondly, SPF is based on EXPRESS, which is the official standard specification for IFC models (Jetlund, et al., 2020).

The EXPRESS data modelling language originates from IFC STEP and adopts object-oriented programming. When using EXPRESS modelling language, software experts represent building data according to specifications of their entities, properties, rules and multiple inheritances described in EXPRESS file extensions (.exp). Several of these IFC schema specifications, such as IFC2X3.exp, IFC2X4_ADD1.exp are available according to their release dates and versions (for further overview, see IFC specifications database available at (BuildingSMART, 2022b). As earlier stated, the connecting point between IFC and semantic web is where a successful conversion and implementation is done from EXPRESS to OWL to create an ifcOWL ontology. The result is an IFC data model that may carry both an EXPRESS and XSD schemas of IFC, but is available in directed labelled graphs popularly known as the semantic web's resource description framework (RDF). These ifcOWL ontology can be available in either Terse RDF Triple Language (TURF) or in RDF/XML formats (BuildingSMART, 2022a).

Several specification schemas facilitate information handling for IFC. These include Model View Definitions (MVD), Information Delivery Manuals (IDM), Property Set Definition (PSD), and International Framework Dictionary (IFD) (Jetlund, Onstein, & Huang, 2020; BuildingSMART, 2022c). For easy comparison, let's take a building solar device entity *ifcSolarDevice* as an example. Its specifications are available in different IFC schemas. Every entity is given an identity, an inheritance definition, a general application, and a formal representation. An MVD specification uses IFC entities to define exchange standards for specific use-cases (See, Karlshoej, & Davis, 2011; Jetlund, Onstein, & Huang, 2020; BuildingSMART, 2022c). An mvdXML specification supports validation and generates documentation for IFC models. It is valued for its capacity to filter IFC data and provide software developers with only subsets relevant for their application (Chipman, Liebich, & Weise, 2016; Jetlund, Onstein, & Huang, 2020). The model view definition of *ifcSolarDevice* is:

“<ConceptRoot xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" uuid="058470b8-ffa8-48af-9b3a-42d3049871cf" name="IfcSolarDevice" applicableRootEntity="IfcSolarDevice">” (BuildingSMART, 2022d).

On the other hand, Information Delivery Manuals (IDM) captures detailed user-defined specifications related to the IFC entity's business aspect that needs to be exchanged during a project. Property Set Definition (PSD) allows any type of property field (that is required) to be added. These IFC property sets (which are alphanumeric data) are available in IFC2x3 PSD Schema IFC4 PSD Schema (BuildingSMART, 2022b). Property set definitions are referenced with XML files. Moreover, ifcXML binding provides a syntactically accurate EXPRESS schema that gives software experts the perfect executable code with the building model's fundamental attributes. Figures 4.1 and 4.2 below show XML and EXPRESS specifications of ifcSolarDevice, respectively.

```
<xs:element name="IfcSolarDevice" type="ifc:IfcSolarDevice"
substitutionGroup="ifc:IfcEnergyConversionDevice" nillable="true"/>
<xs:complexType name="IfcSolarDevice">
  <xs:complexContent>
    <xs:extension base="ifc:IfcEnergyConversionDevice">
      <xs:attribute name="PredefinedType" type="ifc:IfcSolarDeviceTypeEnum" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 4.1: XML Specification of ifcSolarDevice (BuildingSMART, 2022d)

```
ENTITY IfcSolarDevice
SUBTYPE OF (IfcEnergyConversionDevice);
PredefinedType : OPTIONAL IfcSolarDeviceTypeEnum;
WHERE
CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR
(PredefinedType <> IfcSolarDeviceTypeEnum.USERDEFINED) OR
((PredefinedType = IfcSolarDeviceTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType));
CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR
('IFCELECTRICDOMAIN.IfSolarDeviceType' IN TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;
```

Figure 4.2: EXPRESS Specification of ifcSolarDevice (buildingSMART, 2022d)

In terms of information storage, the International Framework Dictionary (IFD) is the IFC standard for terminology databases. In the construction industry, most building engineers, material suppliers, and other stakeholders recognize IFD. It uses tables to classify, store and retrieve building information when requested. However, this

classification system has limitations. Classification is naturally hierarchical, but it lacks standard rules of implementation. Users mostly apply it in different ways, thereby making its extension inconsistent. IFD libraries such as OmniClass tackle this limitation by assigning unique identification numbers for each term, thereby making them unambiguous and machine processable (Jetlund, et al., 2020). Through the formats mentioned above, *ifcSolarDevice* can be defined in any BIM environment with details of its meaning and purpose to the building and construction industry.

4.3.2. IFC Data Schema

One of the primary advantages of BIM and IFC is that they facilitate interoperability amongst various BIM professionals. For data to be effectively shared or re-used, it must be represented in a standard format. In IFC, data is represented through the data structure within the IFC data schema. The schema serves as a storage system and codifies data hierarchically by specifying its object definition (*IfcObjectDefinition*) (buildingSMART, 2022f), relationships (*IfcRelationships*) (buildingSMART, 2022g), and properties (*IfcPropertyDefinition*) (buildingSMART, 2022h). These three define a complete IFC model that could be integrated with a large number of entities.

For instance, for *SolarDevice*, an *IfcObjectDefinition* would describe its identity, semantics, and attributes (such as its name, object type, thermal properties). *IfcRelationships* would describe its connections and relations with other entities (such as a building or roof). *IfcPropertyDefinition* would define properties it is associated with such as processes (installation), people involved (suppliers, owners) and abstract concepts (such as costs, performance, warranty). A common feature within all IFC schemas is that their nature of representation facilitates interoperability. Going into details, each schema is broken down into key fundamental concepts or subtypes of the schema class. These details go beyond the scope of this study, therefore, they are summarised in the figure provided below based on the IFC release (buildingSMART, 2022d). Figure 4.3 shows the subclasses of *IfcObjectDefinition*, *IfcRelationships* and *IfcPropertyDefinition* respectively.

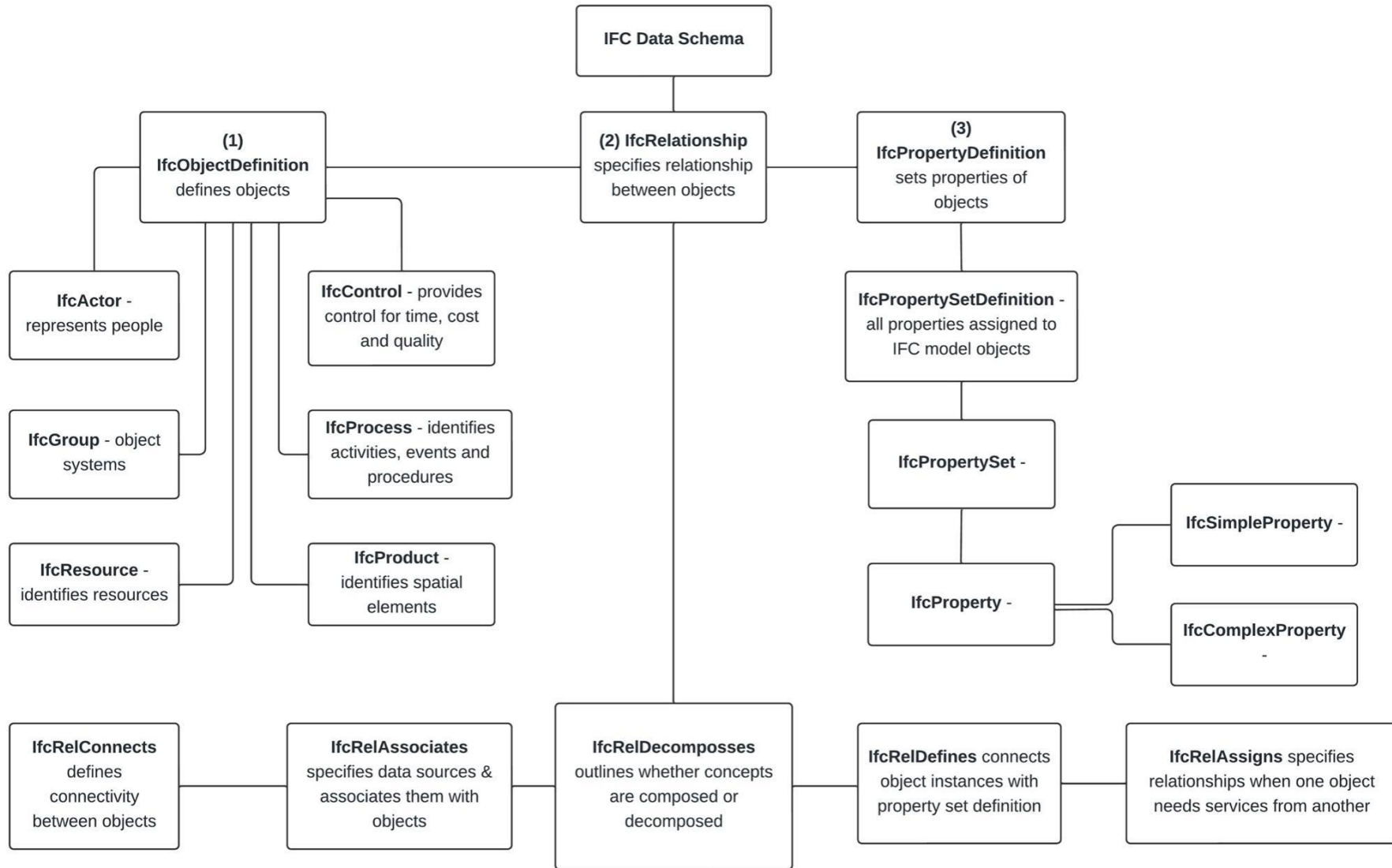


Figure 4.3: Subclasses of *IfcObjectDefinition*, *IfcRelationships* and *IfcPropertyDefinition*

4.4. Industry Foundation Classes (IFC) and Semantic Web

There are numerous similarities between the semantic web's ontology and IFC. Both are adopted within the context of their domain to achieve standardisation goals of data exchange and interoperability. Several researchers have pointed out that the structure of the IFC's EXPRESS specification language is similar to the web ontology language (OWL) of the semantic web (Pauwels & Terkaj, 2016; Petrova *et al.*, 2019). For this reason, researchers have been able to convert IFC data models into RDF models and OWL formats. The question is, why have there been several efforts to integrate semantic web and IFC models? The answer to this question is simple. Firstly, the IFC schema explicitly defines building models with in-depth details of relationships and properties. It is tedious to retrieve such comprehensive data for everyday purposes. Secondly, IFC is non-modular and relies on complex representation through EXPRESS. The IFC schema is also not adaptable when heterogeneous data need to be integrated. Thirdly, IFC has an inadequate set of tools to support complex queries and management of instance data. The semantic web, however, provides the perfect environment for integration, sharing, and query. Several researchers admit there is a considerable added value when semantic web is used on IFC data models to facilitate complex searches, information exchange, and integration (Petrova *et al.*, 2019). One of the most critical limitations of IFC that the semantic web can solve is "semantics." Researchers argue that IFC lacks semantic enrichment and cannot explicitly map entities, their relationships, and rules despite all the benefits (Pauwels and Terkaj, 2016). It cannot provide queries or infer reasoning on data. For these reasons, a group of experts under buildingSMART came together to develop a standard ontology for the IFC named ifcOWL ontology, to provide semantic enrichment for the IFC schema. Once applied, the logic-based nature of OWL provides formal semantics to IFC models and offers enhanced reasoning and meaning to its data. Hence the need to integrate IFC into the current research.

4.4.1. From EXPRESS to Web Ontology Language (OWL), the ifcOWL Ontology

In the past, a tremendous amount of work went into converting EXPRESS into other modelling languages. Several researchers attempted different approaches to convert EXPRESS to OWL over the years. A detailed and thorough appraisal of the previous IFC to OWL conversion approaches is available in Usman et al., (2018). Although IFC instances have a graph-like representation similar to RDF, the EXPRESS schema contains complex elements that vary from OWL's elements. Therefore, a methodological technique must align the distinctive features of the two languages and efficiently convert EXPRESS schema into the corresponding OWL feature. The previous conversion approaches had numerous challenges that hindered wide adoption and standardisation. Firstly, most techniques used OWL 1 instead of OWL 2. OWL 1 lacks the expressive power of OWL 2. It also lacks the advanced first-order logic features within OWL 2 that enable it to handle complex domains. Secondly, most researchers have not published the generated ontology from their approach. Instead, they only presented their ideas in articles. Thirdly, in most of the approaches, the actual richness of the original EXPRESS schema got lost during conversion.

The techniques resulted in private and different implementations of ifcOWL ontology until a standardised approach was developed by Pauwels and Terkaj (2016) and other members of the buildingSMART working group. The main difference between the standard ifcOWL ontology and the previous attempts is its richness in axioms. Its classes, properties, and functionalities are well defined within OWL and "semantically" very similar to the original EXPRESS schema. The developers have provided a comprehensive outline of the differences (Pauwels and Terkaj, 2016). Once the standardised ifcOWL ontology has been published, researchers attempted to enhance its specifications with an alternative way of representing geometric data (Pauwels *et al.*, 2017). Furthermore, a methodology is proposed in (Terkaj and Pauwels, 2017) to develop a less complicated ifcOWL ontology that is modular and could be easily used for domain applications. Similarly, an enhanced EXPRESS to OWL converter has also been designed

(Bonduel *et al.*, 2018). Others attempted to validate ifcOWL ontology using SHACL (a standard for authenticating declared RDF data (Stolk & McGlinn, 2020).

As stated in chapter one, one of this study's objectives is to extend the standard ifcOWL ontology to propose a more straightforward and modular ontology for photovoltaic systems that can be extended towards other domains.

4.4.2. Future of IFC: EXPRESS to Unified Modelling Language

Although several schema specifications exist within IFC, the official standard specification currently for IFC models is EXPRESS (Jetlund, et al., 2020), standardised in ISO 10303 STEP. EXPRESS has a lexical syntax, but its graphical notation is in EXPRESS-G. On the other hand, the unified modelling language (UML) is a de-facto modelling language for software development. It uses an integrated group of diagrams to give standard visualisation of the design of a system. EXPRESS is similar to UML because it adopts object-oriented programming. The rationale behind this mapping is to combine the best of the two worlds of graphical representation of models.

Recent announcements by buildingSMART state their intentions to move from EXPRESS to UML (in the near future) as the official modelling language for IFC (VanBerlo, 2019). It is believed that this move would create wider adoption of IFC and increased community contributions towards its future. It is intended to keep the conceptual model in UML and produce implementation schemas for EXPRESS, XML, and OWL (Jetlund, et al., 2020). The buildingSMART next-generation IFC working group has made available a draft conversion of IFC EXPRESS to UML (Krijnen, et al., 2020). However; the resulting model is still heterogeneous. It would take a while to develop a standardised and stable version.

4.5. Industry Foundation Classes (IFC) in the Context of this Research

IFC is often considered as BIM's main hope to achieve complete interoperability. In the context of this research, we propose that to ensure interoperability in the

domain of sustainable building technology products, three essential conditions must be met. First, there should exist a standard domain data model. Secondly, software implementations of the data model must be feasible. Thirdly, deployment. IFC is BIM's multi-platform standard for representing building products in data models; therefore, IFC makes the first condition.

As discussed in section 4.3.1 above, IFC files have different formats. For instance, files in .IFCXML and .IFCZIP are simply extensions to show the file is XML or compressed as a ZIP file, respectively. Software programs have import features that allow them to import and export IFC files. These include Autodesk's Revit (Autodesk, 2018; Autodesk, 2021), ARCHLine.XP (ARCHLine, 2021), GRAPHISOFT's ARCHICAD (Graphisoft, 2021), SketchUp (with the help of a plugin called IFC2SKP) (Trimble, 2020), Adobe Acrobat (Adobe, 2021), etc. Once an IFC file is imported into these program applications, a complete 3D model is displayed. The programs may wish to modify (work on the building model) and export in IFC format. In ARCHLine, the user can select which design phase to export (ARCHLine, 2021). From Revit and SketchUp in particular, the programs can generate and export 3D PDF files (Autodesk, 2018). An open-source IFC toolkit and geometry engine called ifcOpenShell (ifcOpenShell, 2020) has recently been developed. It imports IFC files and converts them into explicit geometry that Computer-Aided Design (CAD) modelling software can understand. IfcOpenShell can convert IFC-SCF files to OBJ, STP, XML, IGS, SVG. Relatedly, for IFC-XML files, an XML editor is required to access the content of the IFC files for further use. In ontology editors such as protégé-OWL, IFC goes through a conversion process from the EXPRESS schema into an OWL ontology. As soon as an ontology is obtainable, the user can modify, re-use, or extend the IFC ontology (within the editor) for further use. RDF graphs compliant to OWL ontology language can also be built (Pauwels and Terkaj, 2016). Although this process is not straightforward, it is a prevailing and effective way of porting IFC data into an RDF data model. These software applications, programs, or editors that can import, use, and export IFC make our second condition.

We have now demonstrated that IFC is indeed BIM's multi-platform "standard" for representing building products' data. The Import and export features available in the software applications mentioned above allow architectural elements to be transferred from one program to another while preserving the original IFC design's geometry and structure. It is important to note that they do not appear only as 3D models and as the building structural elements they have been designed to be (e.g., walls, roofs, beams). Similarly, IfcOpenShell's ability to convert IFC into several extensions widens its implementation in the domain. This adoption of IFC as the standard for building data exchange by software developers, domain experts, and other users fulfils our third condition.

In the context of this research, the three conditions to ensure interoperability are met. The primary requisite behind effective data exchange and re-use is to have a standard domain-specific online product taxonomy compatible with all software applications. This research recognizes IFC as the de-facto standard data model for the construction domain (condition one). Furthermore, it imports the ifcOWL ontology within the ontology editing software – protégé. This research then re-uses and extends IFC (alongside the conceptual and theoretical knowledge model of the representative problem domain already established) to develop an enhanced knowledge model (a unique ontology) for photovoltaic systems (condition two). Thirdly, this research uses the resulting ontology to propose a web-based application that aids users in making informed decisions on the use of photovoltaic systems (third condition). The decision to extend the standard taxonomy is logical because only continuous use can make standards advance. The proposed application would be compatible, unrestricted, and accessible. It would be virtually suitable for re-use for future studies.

4.6. Applications of Semantic Web with BIM and IFC in Construction

In this study, we observed an indisputable trend in applications of SW based on BIM standards. BIM's initial purpose in the construction domain was to improve collaboration and effective exchange of data in design and construction stages.

Undoubtedly, information management has largely been improved by emerging BIM tools and technologies (Boje, Guerriero, Kubicki, & Rezgui, 2020). The increased adoption of BIM has led to many more indubitable benefits. These include an outsized increase in construction data, enhanced knowledge interoperability amongst construction stakeholders, and increased cost reduction in design and construction. BIM integrates interdisciplinary knowledge into a single coordinated building model. BIM's drive towards improved interoperability in diverse domain applications naturally pulls it towards an integration with SW technologies and its tools. SW however, simply captures data from different BIM models, represents information in a structured format and allows semantic reasoning to be applied.

During this study, a growing trend is observed in the adoption of semantic web-based solutions that incorporate BIM models and tools. In most studies, semantic web represents data captured from BIM models and subjects them to semantic reasoning. Several researchers have attempted to transform the structure of BIM into a semantic web-like configuration to allow inferencing using semantic web reasoners. A BIM-based semantic web research is recorded in areas of construction management and defect management (Park et al., 2013). In a study by Lee, Kim and Yu, (2014), a BIM-based semantic approach was used to tackle challenges of construction cost estimation. Zhang, Boukamp and Teizer, (2015) proposed to use ontology, SWRL and rule engines to visualise compliance checking results based on detailed data obtained from BIM-based hazard detection. Ding et al., (2016) developed an ontology for construction risk knowledge management based on knowledge obtained from BIM models.

Abanda, Fogueu and Tah (2017), developed an ontology representing standard cost estimation measuring methods in machine-processable format for BIM software. This semantic web and BIM approach uses DL-based reasoners to validate proposed ontology. In other studies, researchers attempted to refine the structure of BIM into an RDF-graph-like nature so it can be integrated as linked semantic data (Jeong, 2018 and Bonduel et al., 2018). In a similar research,

authors proposed to convert BIM data into ontology as part of developing BIM applications that support reasoning (Ma & Liu, 2018). Both researchers attempt to re-define BIM structure using SW technologies for inference purposes. In Niknam, Jalaei and Karshenas (2019), BIM product models are converted into RDF-graph-like structures so they can be integrated as Linked data and exchanged with other data that have been semantically represented. The latest project in the BIM-based realm is creating vivid as-is Building Information Models (BIMs) which capture precisely and in detail the exact current state of a building using autonomous robotics (Adan et al., 2020). Where BIM is involved, it is imperative to bring up the data schema and the exchange file format that represents BIM data – Industry Foundation Classes (IFC).

Since the emergence of IFC, the construction industry has adopted more IFC SW integrated methods of sharing construction information. Belsky, et al., (2015), developed an IFC semantic enrichment framework that uses semantic rules to create new IFC files that include the inferred information it has generated. In another study, Bloch & Sacks, (2018) proposed to use a machine learning approach to achieve semantic enrichment on an IFC classification model. While the rule based approach in the former study (Belsky, et al., 2015) was equally effective, the authors claim machine learning approach has added advantages when dealing with indefinite knowledge concepts. Zhang, Beetz and De Vries (2018), proposed an approach called BimSPARQL where the RDF SPARQL query language is extended to query building data represented in IFC. The resulting data were represented in the form of RDF graphs for further queries. In a similar study, SPARQL is applied to reduce the sizes of IFC instance models to make them 2-3 times smaller than RDF databases (Krijnen & Beetz, 2018).

Since the IFC standard provides a central conceptual data schema for BIM data, the translation of its EXPRESS schema to OWL ontology (Pauwels and Terkaj, 2016) offers the domain a more structured representation of building data enhanced reasoning, more effective semantic searches and improved interoperability. Since the emergence of the standard ifcOWL, several studies

have attempted to re-use the approach to propose software applications. In González *et al.*, (2020), the authors used the framework of ifcOWL ontology to code information vital to indoor navigations. In another study, researchers attempted to develop an ifcOWL-DfMA ontology to become the core vocabulary for offsite construction and design management (Kalemi et al., 2020). This project aims to represent domain knowledge in a machine-processable format. However, ifcOWL-DfMA was developed independently of the standard ifcOWL ontology, although they use aligned terminology.

4.7. Conclusion

This chapter presents an overview of BIM and IFC and discusses their relevance to achieving the objectives of this research. Although BIM offers solutions to key challenges within the construction domain, its main relevance to this research lies in its ability to enhance collaboration and support decision-making at every stage of a construction project, including the choice of sustainable technology to acquire. Hence, it is essential to apply BIM-based research and solutions to provide decision support in the domain of this study. Similarly, being the BIM's multi-platform standard for representing built assets, this study utilises IFC's functionality as a product classification library and taxonomy, to describe and represent, exchange, and re-use required photovoltaic technology data models.

This chapter overviewed the overlapping goals of IFC and the semantic web and discussed how the standard ifcOWL ontology came to be. The logic behind the adoption of IFC in the context of this research is provided in three folds. Firstly, re-use of IFC is done because it is the de-facto standard data model for the construction domain. Secondly, extending the standard ifcOWL ontology within ontology editing software – protégé ensures an enhanced knowledge model is developed. Thirdly, this proposes to use the resulting ontology to propose a web-based application that aids users in making informed decisions on the use of photovoltaic systems. These three ensure interoperability is met. The decision to extend the standard taxonomy is logical because only continuous use can make

standards advance. The chapter further elaborated on the integration of BIM, IFC, and semantic web and gave an appraisal of their numerous applications in the construction domain. The next chapter focuses on the methodological framework used in this study.

5. Research Design and Methodology

5.1. Introduction

Research methodology provides the techniques used in establishing a research. It outlines the processes used to identify, obtain, study and analyse information within the research topic in order to achieve the objectives of the research. This chapter first reviews the most used research methods and highlights their strengths and limitations. Based on these, the most appropriate framework for the research is selected. The chapter further justifies the methodological framework adopted. This chapter also elaborates on how to accomplish the study's aim and objectives, as explained in *chapter 1* using the selected methodology. This chapter provides a brief overview of research methodologies in 5.2. The methodologies adopted for this research are selected and justified in 5.3. The research methodological framework is extensively explained in 5.4. The conclusion presents a summary of the chapter in 5.5.

5.2. Overview of Research Methodologies

Research methods split broadly in large volumes of literature into quantitative, qualitative, and mixed methods. This decision to adopt any method largely depends on their distinctive differences in research techniques. While it may seem easy to decide to use numerical data versus words, techniques used to analyse the data collected are more effective factors to consider when deciding which research approach to use (Bager-Charleson and McBeath, 2023). Conversely, some studies adopt mixed methods by collecting data qualitatively and analyse them quantitatively (Morgan, 2014; Culha, 2023). Therefore, the choice between the two approaches can go beyond both formats of data and techniques for analysis.

Qualitative research examines human reasoning and what triggers or controls decisions. It aims to understand individual behaviour depending on dynamics, situations, and contextual or social beliefs (Bazeley, 2015; Mohajan, 2018;

Samuels, 2019; Borgstede and Scholz, 2021). The qualitative study brings out intricate details about a situation through ideas, opinions, experiences through interviews, surveys, focus groups, case studies and many more (Hameed, 2020). It describes phenomena extensively and in detail, thereby revealing unique and unexpected circumstances. A very detailed overview of qualitative research can be found in Ugwu and Eze, (2023).

On the other hand, quantitative research primarily focuses on testing the certainty or falsehood of a theory (Bloomfield & Fisher, 2019). It generates numerical data that are statistically analysed (Rhodes, 2014; USC, 2020; Borgstede and Scholz, 2021). Both methods aim at identifying research problems. However, since each process has distinctive research approaches, researchers often debate which method best suits their work needs. This study does not involve social sciences and therefore will not contest in detail the deliberation of which method is best. However, table 5.1 provides a summary of the differences between the two methods based on a thorough comparison of the techniques found in the literature (Morgan, 2014; Ahmad et al., 2019; Hameed, 2020; Borgstede and Scholz, 2021; Culha, 2023; Ugwu and Eze, 2023; Bager-Charleson and McBeath, 2023). This summary can serve as a guide to select which research paradigm to adopt.

Table 5.1 Differences between Qualitative and Quantitative Research

Features	Qualitative Research	Quantitative Research
Sample Size	<ul style="list-style-type: none"> ● Small and purposeful ● Not statistically projectable 	<ul style="list-style-type: none"> ● Large and random sample size ● Statistically projectable
Data Collection Methods	<ul style="list-style-type: none"> ● Interviews ● Focus Groups ● Case studies ● Ethnography 	<ul style="list-style-type: none"> ● Surveys ● Experiments ● Observations
Data Analysis	<ul style="list-style-type: none"> ● Inductive analysis by the researcher ● Non-statistical, Non-numerical ● Focuses on concepts 	<ul style="list-style-type: none"> ● Deductive analysis based on statistical methods ● Numerical and Statistical
Data Outcome	<ul style="list-style-type: none"> ● The result is a basis for future research 	<ul style="list-style-type: none"> ● It guides business decisions and actions.
Advantages	<ul style="list-style-type: none"> ● It facilitates a deep understanding of theory/application. 	<ul style="list-style-type: none"> ● It collects information from many participants. Thus, results can be

	<ul style="list-style-type: none"> • The outcome provides sound rationale and justification of individual perceptions and experiences. • It helps to identify new and raw phenomena, as well as unique processes of action. 	<p>easily generalised to the broader population of interest.</p> <ul style="list-style-type: none"> • Research can be done in groups for comparative analysis. • Provides statistical information; therefore, communication is easy to understand.
Limitations	<ul style="list-style-type: none"> • It collects information from a small sample of participants. Thus, findings cannot be generalised to the broader population of interest. • Statistical methods cannot be applied. 	<ul style="list-style-type: none"> • New and unique phenomena cannot be found. • Participants are selected at random; thus, results may not represent information from the target of interest (people with actual knowledge of the domain).

Once a research methodology is adopted, data collection techniques that aim to elicit information from several literature sources are chosen (Bager-Charleson and McBeath, 2023). The review of literature is one of the most fundamental approaches to data collection. The examination conducted in this research was primarily from a pool of existing scholarly works related to the research domain. Literature sources identified during this study include peer-reviewed publications, books, reports and other electronic sources with significant results, models, and theories on the domain, as well as studies that respond to major works and theories. Other sources include analysis of phenomena through a medium of direct interaction with domain experts.

As earlier stated, this study does not involve the social sciences. However, a single qualitative approach to data collection and evaluation known as case study is considered in this study. An exploratory case study approach could enhance the depth of comprehension of knowledge on the problem domain (Coombs, 2022). It would provide an in-depth understanding of sustainable building technology products and photovoltaic systems in its natural real-life context. Through descriptive case studies, knowledge could be elicited from identified domain experts to understand the underlying factors that trigger decisions that are made on the choice of sustainable building technology products. Although domain knowledge could be easily gathered from several non-human sources such as workbooks and technical manuals, practising experts provide a more significant

comprehension of the problem. Furthermore case studies are beneficial for the validation of research prototypes and the testing of theories. Therefore, an exploratory case study approach is applied within the methodological framework of this research. Further basis for this choice is provided in the subsequent section.

5.2.1. Justification for Exploratory Case-Study Research Methods

An exploratory research method is adopted to investigate both the semantic web technologies domain and sustainable building technology domain. This method is selected for several reasons. Firstly, although broad research has been conducted recently on semantic web technologies, the domain is still evolving. Its adoption with building information modelling and industry foundation classes involves using tools and technologies that need to be more conclusive and standardised. Similarly, the domain of the photovoltaic system could still be further researched. The use of exploratory research on photovoltaic systems helps lay the foundation for further investigation on other sustainable building technology products in the future. Secondly, it gives an in-depth understanding of existing problems or paradigms within the domains (Coombs, 2022). Its findings provide direction to the subsequent procedures involved in the study. This study benefits from the flexibility that exploratory research has because the study adapts to changes as the investigation progresses. Thirdly, in the experimental method, research evidence is obtained to support findings. In the context of this study, a software prototype is developed to serve as evidence to research results. In information communication technology, the case study approach is one of the best practices for validating research prototypes developed based on exploratory research (Rashid et al., 2019). This study chooses to validate its prototype using real-world scenarios and circumstances from case studies rather than simulated data.

Case studies elicit knowledge from the domain experts to understand the human behaviour, motivations, and the underlying opinions that trigger decision-making strategies (Coombs, 2022). In the case study approach, you cannot manipulate

the experts' conduct or knowledge since its main aim is to examine why individual decisions are made. For instance, in this study, semantic web technologies are explored to provide decision support to individuals on the choice of photovoltaic systems technology. A case study supports this study by providing answers to critical research questions: (a) What are the current issues concerning selecting and installing PV-systems? (b) Why is a computerised recommendation process of PV needed? (c) How is the choice of PV-system technologies made? (d) what are the exact conditions that are relevant or that guide selection of PV technology? In the semantic web context, the following questions justify adopting the case study approach: (e) Why should semantic web technologies be used to recommend PV technology? (f) How can semantic web model knowledge on PV? (g) What is the extent to which semantic web technologies can be used? The answers to these questions give the research an efficient way to model PV technology knowledge using suitable semantic web technologies. This in-depth, comprehensive knowledge and decision points can only be obtained from the exploratory case-study. It is incorporated into the main research framework discussed in section 5.4.

The process of knowledge elicitation and conversion into useful knowledge models cuts across knowledge engineering, software engineering and ontology engineering because ontology development is a sub-task of knowledge-based system (KBS) development. The basics of ontological and software engineering domains have been discussed in chapter 3. The fundamental techniques of knowledge engineering in the context of this research is provided in the next section (5.3). Based on the assessments of the most popular methodologies under each engineering field, this study selects the methodologies that are most appropriate to meet the objectives of this study. The methodologies adopted and a justification for their selection is provided in section 5.4.

5.3. Knowledge Engineering

To understand knowledge engineering, the main keywords associated with it such as data, information, knowledge, domain, and knowledge-based systems

should be analysed. In a simple terminology, data is simply unprocessed and raw details about a subject, while information is processed data that enables understanding. Knowledge is basically a thorough understanding of a subject either through theoretical or practical methods. Therefore, knowledge is processed information while information is processed data (Nicholas-Institute, 2018). A knowledge-based system uses an adequate amount of knowledge and expertise to solve problems often related to the user's area of interest, a domain. To develop a knowledge-based system, a series of steps that involve knowledge acquisition, representation, specification, inference, and evaluation are conducted to develop the system (Shadbolt and Smart, 2015; Yun et al., 2021). These steps make up knowledge engineering.

Although there exist many knowledge engineering methodologies, for decades, there has not been one general way of collecting, analysing, or representing knowledge to build applications. Earlier researches have introduced the processes of ontology-based knowledge modelling, thereby establishing the relationship between knowledge engineering and ontology engineering. Liu and Zarate, (2014), introduced knowledge modelling processes for ontology representation and provided technologies used for domain applications. Fussl et al., (2015), studied the relationship between knowledge modelling and ontology development with detailed research on modelling tools, technologies, and domain applications developed to aid decision making. Gayathri and Uma (2018) only studied reasoning as part of knowledge modelling based on ontology development. A more recent study provided a comprehensive review on the methods used, processes involved in, and techniques used in knowledge modelling (Yun et al., 2021). The review was aimed at helping developers in selecting methodologies most suitable for their work.

Some common knowledge engineering methodologies include Methodology for Knowledge-Based Engineering Applications (MOKA) (Sandberg, 2015), Model-based and Incremental Knowledge Engineering (MIKE) (Studer et al., 1998), knowledge network (Beckmann, 1995), two-hemisphere model driven

(2HMD) (Nikiforova and Kirikova, 2004), Common Knowledge Acquisition and Documentation Structuring (commonKADS) (CommonKADS, 2021). Adoption of any of the methodologies would most likely give the same results, however, each has its benefits and limitations. CommonKADS is selected to be the principal methodology in this study. This choice is because commonKADS is the leading standard in knowledge intensive analysis and system developments (Schreiber et al., 2000). Further justification for this choice is provided in the next section.

CommonKADS achieves its goals through a suite of 3 models. Firstly, the organisational environment analyses a knowledge organisation to identify bottlenecks (through the organisational model), it describes tasks to be performed (within the *task model*), describes the tasks that agents would execute and models communication between agents (in the *agent model* and *communication model* respectively) (Yun et al., 2021). Secondly, the *Expert knowledge model* provides specification for knowledge involved and thirdly, the *design model* describes the structure of the system to be developed (CommonKADS, 2021). These models are complex and for that reason, this study will not elaborate the activities involved in each. However, all models are incorporated in three main steps that make up the framework of CommonKADS. These are knowledge identification, knowledge specification and knowledge refinement (Schreiber et al., 2000).

The main aim of the knowledge identification stage is to obtain knowledge that would make up the knowledge model (Shadbolt and Smart, 2015). This is done through investigations from literature sources, and knowledge elicitation (Shadbolt et al., 2015) through interactions with domain experts. Similarly, commonKADS categorises knowledge into domain knowledge, inference knowledge or task knowledge for specific purposes in the knowledge-intensive framework (CommonKADS, 2020; Yun et al., 2021). Obtained knowledge is then specified in the knowledge specification stage where an appropriate task template is selected (Schreiber et al., 2000). At this stage, knowledge conceptualization is conducted. It is at this point that re-use of existing

components is done to decrease complexity and to ensure standardisation. Once the conceptual model is obtained, knowledge is refined through validation in an iterative manner until the desired system is attained. These stages prove that CommonKADS supports development of proficient knowledge-based applications by providing an exact route to implementation (Schreiber et al., 2000). Prototype development is often recommended rather than a full version since the knowledge-based system is developed in an iterative manner (Kent *et al.*, 2021). Further details on the commonKADS framework as applied in this study can be found later in this chapter (section 5.5).

Some applications of commonKADS can be seen in Guillen and Maceda (2014), where authors applied commonKADS and an ontology to develop a knowledge base for a prototype development in the field of veterinary diagnosis system. Yang et al., (2016) used commonKADS with software engineering to enhance information management in their study. Martins et al., (2018) applied commonKADS and a qualitative research approach to develop integrated methods to support e-government project management. In Santirojanakul (2018), commonKADS was used to develop a sports knowledge management system. Saleh et al., (2018) applied a service-oriented architecture to the commonKADS methodology to improve its reusability.

5.3.1. Justification for CommonKADS Knowledge Engineering

Methodology

CommonKADS is adopted as the main methodological framework of this study for several reasons. Although, several knowledge engineering techniques have been developed and applied over the years (Shadbolt & Burton, 1995; Cooke, 1994, Cooke, 1999; Shadbolt, 2005; Milton, 2012); however, CommonKADS is the leading de-facto standard for knowledge modelling supported by concrete guidelines and tools (Schreiber et al., 2000; Shadbolt and Smart, 2015). It is more artificial intelligence related than the other existing methodologies. It also has a wide range of standardised reusable components (domain, inference, and task templates) that improve the elicitation process' proficiency (Shadbolt and Smart,

2015; Yun et al., 2021).

Similarly, unlike other knowledge engineering methodologies, CommonKADS has two distinctive qualities. Firstly, it can be easily modified; it is one of the most effective and flexible knowledge modelling methods (Yun et al., 2021). Secondly, it provides a unique connection (through notations and tools) to other engineering methodologies (CommonKADS, 2021). It can be either adopted as a whole or partly in combination with other research methodologies. In the context of this research, CommonKADS unambiguously integrates its knowledge engineering techniques to the other selected methodologies (exploratory case-study methods, ontology development 101 and agile software engineering guides). Furthermore, CommonKADS is unique because it provides an easy connection to modern object-oriented paradigm such as UML. CommonKADS' structure that enables effective integration with different core engineering methodologies makes it appealing and fitting to be used for this proposed system. Semantic web and sustainable building technology are domains with large knowledge bases. CommonKADS is suitable for such large domains since it provides standard methods that support knowledge-intensive systems development (Yun et al., 2021). Furthermore, once the knowledge engineer provides a formal specification of the knowledge model, CommonKADS offers an exact route to implementation. This makes it suitable for developing the prototype in this research. Surveys strengthen our choice and show that CommonKADS is the leading methodological standard for knowledge-intensive system development (Studer et al., 1998; Schreiber et al., 2000; Yun et al., 2021; CommonKADS, 2021).

5.4. Methodologies Adopted in the Research

Based on the research methodologies examined and reviewed above and in (Morgan, 2014; Burch and Heinrich, 2015; Corbin and Strauss, 2015), exploratory case-study research methods are believed most appropriate and thereby chosen for use in this research for the reasons provided in section 5.2.1 above. Furthermore, as earlier established, this study involves research and

development of an ontology based knowledge model (prototype), to implement sustainable building technology products (domain). It cuts across three engineering approaches: *Knowledge Engineering*, *Ontology Engineering*, and *Software Engineering*. It is for this reason that the knowledge engineering methodology CommonKADS is adopted as the primary research framework and extended to include the "ontology development 101" guide (Noy & McGuiness, 2001), the Agile Software Engineering methodology (Hughes, 2015) and a case-study approach (Rashid et al., 2019). There are two bases for the connection between these three engineering fields. Firstly, the process of developing an ontology through any of the reviewed ontology development methodologies (in chapter 2) involves knowledge elicitation and representation techniques (knowledge engineering). Similarly, ontology development also entails writing computer programs within application software for systematic design, development, and maintenance of the ontology. Therefore, knowledge, ontology, and software engineering are interlined to achieve the aims of this study.

Section 5.4.1 and 5.4.2 justify the choices for ontology development 101, and agile software engineering methodology, respectively. These sections indicate how the selected methods fit the proposed framework of this research.

5.4.1. Justification for Agile Software Development Methodology

Several software development methodologies have been reviewed in chapter 2. Based on the outcome of the review, the agile software development methodology is deemed most suitable to be adopted in this study for the following reasons.

Firstly, the agile framework has been an emerging choice of methodology to software engineers for the past decade and half (Govil and Sharma, 2022). Amongst all other methodologies, agile methodology stands uniquely due to its distinctive flexible nature (Javed, Khan and Tubbassum, 2019). Several of the most popular software development methodologies borrow a few best practices from agile software development. Scrum uses an agile approach to have a close

collaboration between development teams. Extreme programming follows an agile approach of delivering software in sprints frequently and allows changes to be made when required. The iterative nature of agile is also imitated feature driven development (Pocsova et al., 2020). Agile methodology is most liked for its incremental delivery of the software, where the outcome at the end of each stage of implementation can be assessed and the software engineer can address inaccuracies in the software (Saeedi & Visvizi, 2021). The approaches in agile methodology have been critically analysed and credibly validated through experienced decision-makers and Fuzzy TOPSI mathematical models in Govil and Sharma, (2022). Agile is ranked number 1 and the most ideal methodology when compared to other software development methodologies.

In the context of this study, agile methodology is highly capable of handling the complexity and variability involved in integrating knowledge-based, ontology and software engineering approaches. In research studies that involve development of knowledge models, prototype development can be an iterative process applied during any stage of the development. Since prototype development involves an iterative process of getting feedback on the design methods and findings in the research, the software development approach most suitable for this purpose is the agile software methodology. Furthermore, since ontology development could be a continuous process of knowledge specification and refinement as they are elicited, the agile framework would support an iterative introduction of new facts and requirements into the ontology as they emerge until a well-structured prototype is derived. In an agile approach, a developed software application can also be expanded and improved upon after feedback is provided. Since this study provides a solution to the domain of photovoltaic systems with possible expansion to other sustainable building technology products, it would be easily expanded for future developments under agile. Furthermore, the domain is evolving in terms of new technologies and standards. Therefore, the agile method is fitting since it has an adaptive approach to handle any shifts in technology and allow recurrent alterations. Similarly, this study aims to develop a web-based application that facilitates decision making

which at the end should satisfy the user. User satisfaction is one of the biggest priorities of agile. Where user satisfaction is not obtained, agile methodology detects and resolves incompatible expectations untimely.

5.4.2. Justification for Ontology Development 101

Several ontology development methodologies have been reviewed in chapter 2. The review conducted will not be repeated in this section. However, based on the outcome of the appraisal done in chapter 2. Ontology development 101 is deemed most suitable to be adopted for this study for the following reasons.

Ontology development 101 is the most widely applied ontology development methodology in research projects in a wide range of domains (Abanda, Tah and Duce, 2013; Khosrojerdi, Gagnon, & Valverde, 2021). It has an explicit specification of vital techniques in the development process. It has straightforward steps that are easy to follow. Ontology Development 101 focuses on the development process' collaborative aspects, which is a fundamental quality for this study (Zeginis et al., 2012). Similarly, this study relies on the benefits of re-using existing knowledge models to increase interoperability, reduce time and cost of development, prevent 're-inventing the wheel,' and guarantee consistency. This makes ontology development 101 fitting since its techniques support effective re-use of existing knowledge models. Based on these, it is perceived as the most mature of all development methodologies (Noy and McGuinness, 2001; Yun et al., 2021). Likewise, it is highly compatible with the ontology editor of choice (protégé-OWL). For these reasons, Ontology Development 101 is the preferred choice of ontology development methodology for the proposed research.

5.5. Research Methodological Framework

It has been established that CommonKADS is adopted as the main methodological framework for this study. Figure 5.1 to 5.3 below presents the research framework and distinguishes between the three engineering fields and the case study approach. The following sections discuss the four main stages of

knowledge engineering in CommonKADS – knowledge identification in 5.4.1, knowledge specification in 5.4.2, knowledge enhancement in 5.4.3. and prototype development in 5.4.4 and recommendations for use in practice and future research in 5.55.

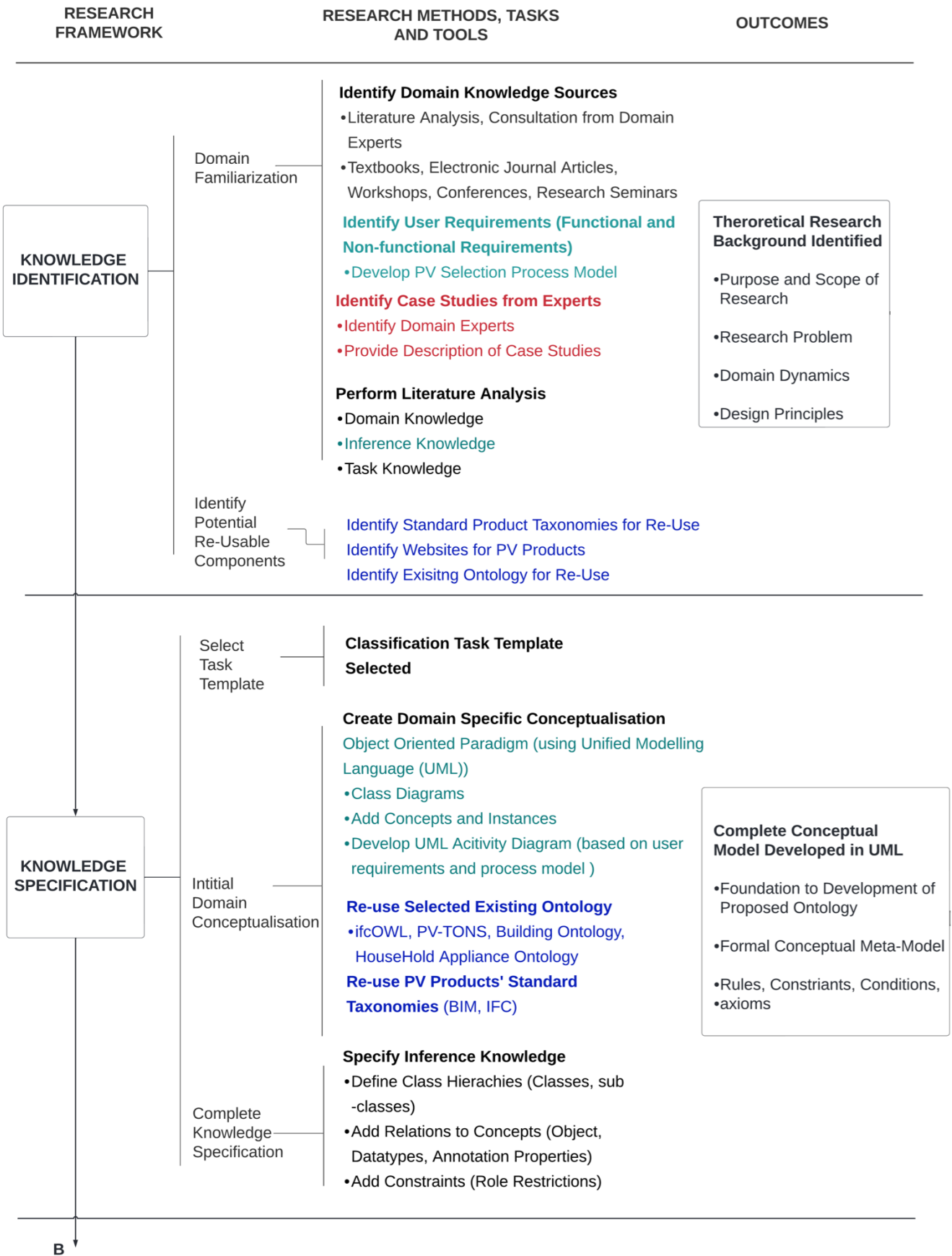
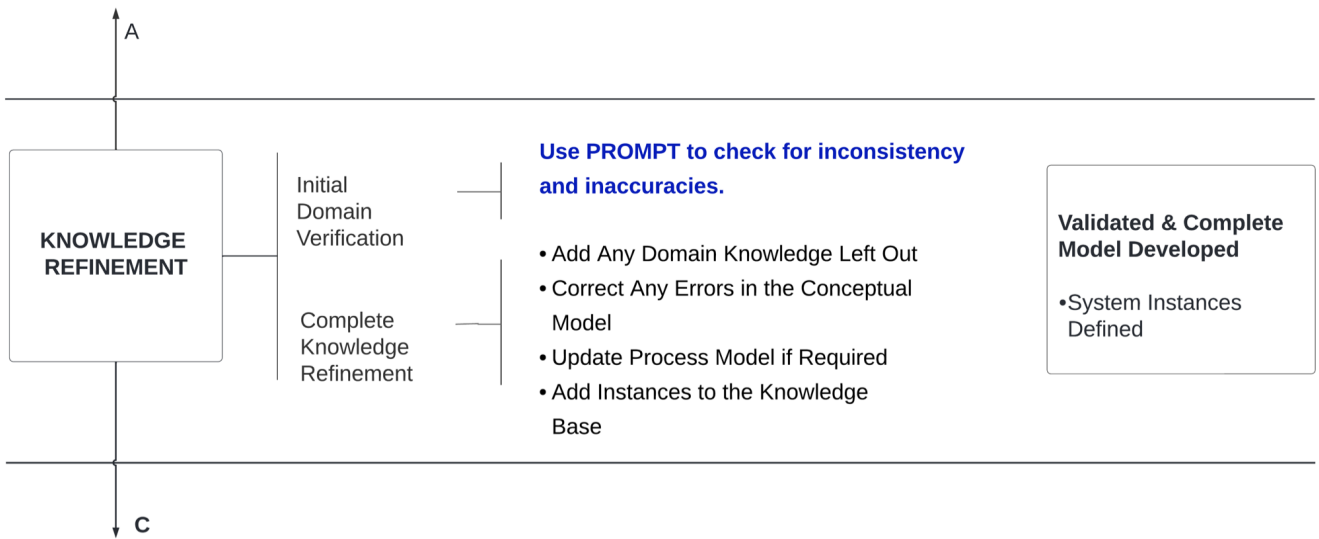


Figure 5.1: Research Framework (Part A)



KEY

CommonKADS Methodology

Qualitative Case Study Approach

Ontology Development Methodologies

Agile Software Engineering Methodology

Figure 5.2: Research Framework (Part B)

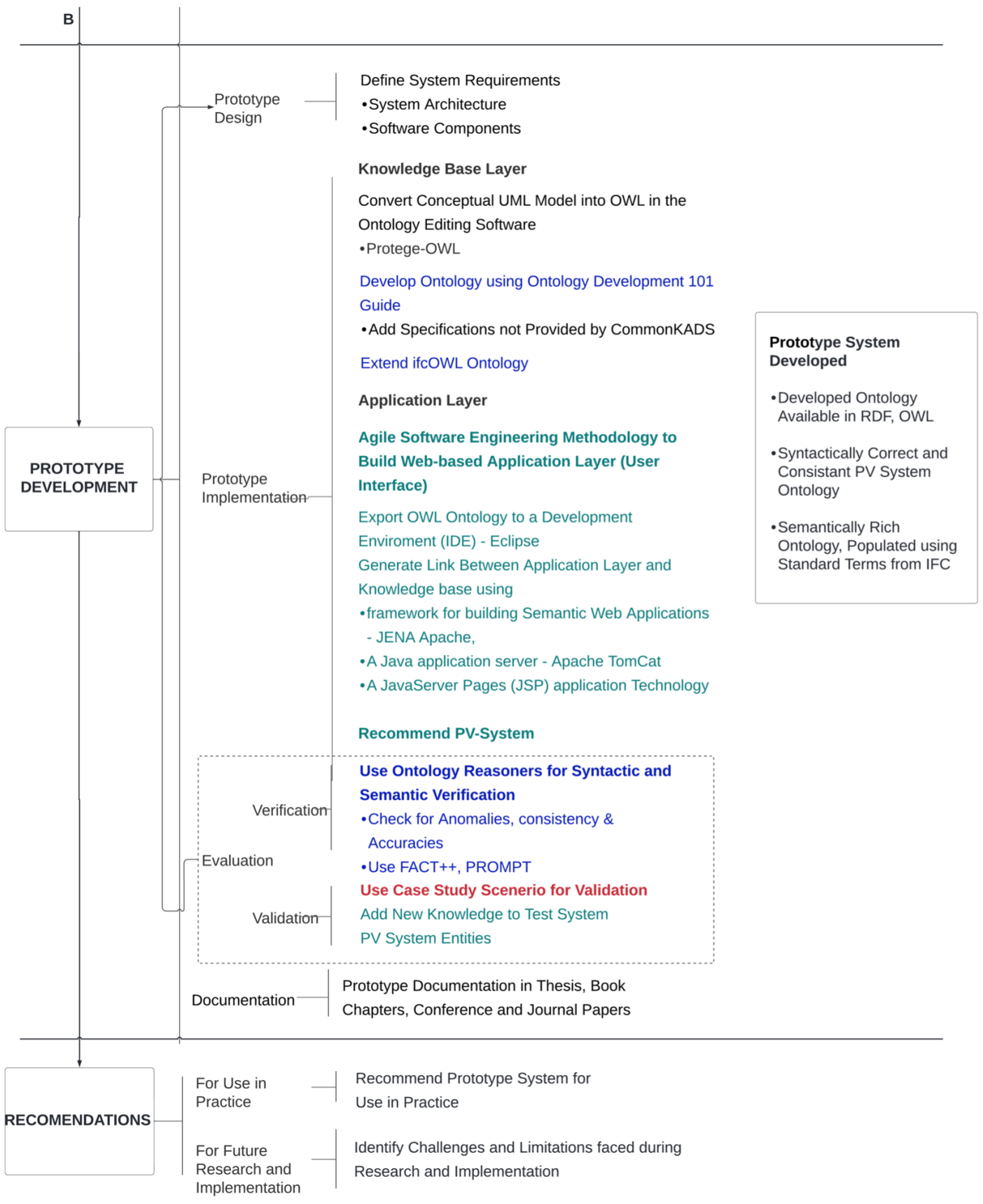


Figure 5.3: Research Framework (Part C)

5.5.1. Knowledge Identification

The main aim of the knowledge identification stage is to obtain a thorough survey of the domain knowledge (including scope, research problems, domain dynamics, and design principles) and to prepare the knowledge for its specification. These objectives are achieved in the next sections through domain familiarisation from literature review conducted on semantic web technologies and sustainable building technologies (see chapters 2 and 3) and the reuse of domain knowledge. Once the research domain's theoretical background is identified, obtained knowledge is prepared for specification based on the CommonKADS framework.

5.5.1.1. Domain Familiarisation

The knowledge identification stage begins with domain familiarisation (by identifying knowledge sources) and analysing knowledge obtained. This research deals with two emerging domains. Semantic Web Technology is an enabling technology applied to provide solutions to the problem domain; Sustainable Building Technology Products with a specific focus on photovoltaic systems. Extensive literature reviews were undertaken. Knowledge about Semantic Web technology was obtained from different knowledge sources such as research seminars and training workshops at Oxford Brookes University, American University of Nigeria, and Udemy online training. Other sources include popular semantic web websites such as World Wide Web Consortium (W3C, 2020a); the Semantic Web Company (SWC, 2020); Protégé (StanfordUniversity, 2020). Semantic Web experts provided clarity and guidance since the semantic web is an evolving technology. The review on sustainable building technology products was undertaken primarily from vital knowledge sources such as peer-reviewed publications, books, and other electronic sources. The researcher also authored some peer-reviewed papers. Feedback received from conference reviewers and comments from journal reviewers enhanced the quality of this research. Furthermore, domain knowledge was abstracted through consultations with domain experts. These experts are research practitioners proficient in academia in regards to the PV domain and consultants who deal with PV applications in practice. Likewise, more knowledge was identified from domain websites such as Energy Savings Trust (EnergySavingTrust, 2022), Carbon Trust

(CarbonTrust, 2022), sustainable energy technology products classified by Building Research Establishment (BRE, 2022a). It is fundamental to identify the current knowledge gap to pinpoint where and what to contribute to the research domain.

An extensive domain analysis done provided a critical assessment of obtained knowledge used in making critical research decisions. This includes what are the dynamics of domain knowledge on SBTP? What critical areas of SBTPs to focus on? What are the main founding theories? Who are the renowned authors? What are the latest arguments in academia in the domain? Etc. Furthermore, knowledge is categorised into a domain, inference, and task knowledge for specification purposes. The outcome of familiarisation and analysis on both research domains reveal approaches, tools, and technologies to be used. Furthermore, it informs the conceptual framework of this research.

5.5.1.2. Identify Potential Reusable Components

The second step of knowledge identification is to identify potentially reusable components. Reuse of domain components has numerous advantages. The first is interoperability. The motivation behind the increase in research and development of semantic web applications in the construction domain is motivated by the target to solve the domain's ongoing interoperability concerns. Reuse of existing knowledge components improves interoperability, efficiency, consistency, and value of any knowledge model. Additionally, reuse prevents re-inventing the wheel and reduces development time (Hu *et al.*, 2021). Furthermore, because validated knowledge is acquired, reuse leads to the standardisation of knowledge models. In this study, firstly, standard construction product taxonomies were identified for reuse. These include BIM (which uses standard terminology to represent elements of a building) and IFC (Pauwels and Terkaj, 2016) (a standardised, open, independent, digital representation of the built environment). IFC provides standard language for attribute definitions and property sets. Secondly, popular websites with detailed knowledge on sustainable building technology and PV products are identified, and reusable knowledge components were retrieved (GreenGlobes, 2021). These include building construction elements and products classified by Uniclass (NBS, 2020), Green Book Live

(GreenBookLive, 2020), building construction techniques organised by BRE (BRE, 2022b) carbon trust (CarbonTrust, 2022), LEED (LEED, 2023), etc as discussed in chapter 2.6. Ultimately, existing PV domain ontologies were ascertained for possible reuse. This includes Photovoltaic Technology Ontology Systems (PV-Tons) (Abanda, Tah, and Duce, 2013), ifcOWL ontology (Pauwels & Terkaj, 2016), and frameworks used to develop other domain ontologies such as Building Element Ontology (Pauwels, 2018), IoT ontology (Agarwal, et al., 2018), the Building Product Ontology (BPO) (Wagner and Rüppel, 2019), The Smart Applications REference (SAREF) ontology (SAREF, 2020), Open Energy Ontology (OEO) (OpenEnergyFamily, 2020).

5.5.2. Knowledge Specification

The knowledge specification stage completes the knowledge model. This involves adding properties and relations to domain attributes and concepts that have been previously established (in the knowledge identification stage). The knowledge specification stage of commonKADS consists of *selecting a task template, initial domain conceptualization, and complete knowledge specification*. The subsequent sections discuss each method.

5.5.2.1. Select Task Template

The first step is selecting a suitable task template for knowledge specification. CommonKADS has catalogues of several task templates of knowledge models that support reuse. Based on the review of the different task templates by (Schreiber et al., 2000; CommonKADS, 2020), each of these templates is specific for task types. For this research, the classification task template was chosen.

The justification for this choice is as follows. Firstly, the classification task template is most suitable for domains where product categorization is done (Schreiber et al., 2000). For instance, in the field of photovoltaic systems, objects can be hierarchically classified, such as PV component classifications. Secondly, it has an established domain schema model that explicitly and correctly defines classes for the objects that have been classified (Schreiber et al., 2000). It has a straightforward pruning method where it catalogues all things to respective classes, specifies attributes to each object, gives

values to all attributes, and analyses modelled knowledge to remove all inconsistent classes. The developer may decide to use a decision tree, information theory, or user control when selecting attributes (CommonKADS, 2020). Thirdly, the nature of its inference structure is appropriate for applications that aim to assist in decision-making. Queries on photovoltaic systems can be easily deduced using any object-oriented-based inference engine. It is one of the most straightforward templates, but its task methods are very analytical and goal oriented.

5.5.2.2. Initial Domain Conceptualization

The *initial knowledge conceptualization* stage involves primary commonKADS methodologies for knowledge capture, an ontology development methodology, and the agile software engineering methodology. Knowledge capture requires a knowledge engineer to elicit domain knowledge from peer-reviewed publications (and other literature sources), domain experts, and potential reusable knowledge models, as formerly mentioned in section 4.5.1. Literature sources have proven highly useful in providing vital attributes of sustainable building technology products and semantic web technologies. Similarly, case study scenarios from knowledge specialists with extensive experience in the research domain contributed useful insights into knowledge not easily captured from literature sources. Additionally, the use of potential reusable knowledge from domain websites, standard taxonomies, and existing ontologies into the knowledge model incites the need to use ontology alignment techniques and tools from the ontology development methodology established by Noy and Musen (2002). This ontology development methodology and the case study approach of knowledge elicitation both use qualitative knowledge analysis. Furthermore, to have a structured representation of elicited knowledge, a hierarchical conceptualization of domain attributes is developed in the form of an object-oriented paradigm using Unified Modelling Language (UML). The agile software engineering methodology is used throughout the domain conceptualization stage as a loop where knowledge is captured, analysed, and represented in a single circle. Where additional knowledge is obtained, the loop is reiterated.

5.5.2.3. Complete Knowledge Specification

The *complete knowledge specification* stage involves stipulating the inference knowledge. Here, relationships between previously defined concepts in 4.5.2.2 are established. Names and values of object properties are defined. Inverse functional properties are also presented. Annotation properties are then set to allow annotations to be placed on class names, property definitions, and ontology names (so knowledge can be imported from other ontologies when required).

5.5.3. Knowledge Refinement

The *knowledge refinement* stage involves two stages. The first stage involves the initial semantic verification of the research knowledge model, while during the second stage, a complete refinement of the knowledge model is done by adding all domain knowledge instances.

5.5.3.1. Initial Domain Verification

During this stage, the knowledge model is checked for anomalies, inconsistencies, and inaccuracies. PROMPT – the ontology alignment tool was chosen for this research (as agreed in section 1.7.1), which also provides support for semantic verification in ontology development and is used for semantic checking of the knowledge model. This verification process could be iterated back to the *knowledge specification* stage in 4.5.2 above if required. Once all errors are identified and corrected, the knowledge model is refined as explained in the next section.

5.5.3.2. Complete Knowledge Refinement

Knowledge refinement stage checks for any domain knowledge that may be left out during knowledge specification and adds this missing knowledge to complete the knowledge model. Domain instances are also added to the knowledge model. Previously developed process models and conceptual models are then updated to contain all subsequently added knowledge.

5.5.4. Prototype Development

One of the objectives of this research is to design and implement a prototype system with an evident amount of intelligence capable of providing decision support in the selection of photovoltaic systems. This prototype is developed based on the knowledge acquired and represented in the previous sections. The prototype for this research is developed iteratively based on the agile software development methodology. It would be treated as part of a future extensive expert system where the knowledge model and inference of the system would be tested and improved upon. This prototype could also be used to test theories about applications of the semantic web as well as be a tool for further research in the domain. The prototype is designed to be understood by professionals in the construction and sustainable building domain and by end-users with little knowledge of the research domain. The user interface will also be designed to be easily traversed by users with little computer skills. The prototype development phase includes *prototype design*, *prototype implementation*, *prototype evaluation*, and *prototype documentation*, all discussed in the subsequent sections.

5.5.4.1. Prototype Design

The prototype to be developed is a web-based application that will use the proposed ontology as a structured central knowledge base. Therefore, the main aim of the prototype design stage is to establish the system requirements, system architecture, and software components of the proposed application. While UML models developed represent concepts and properties to be implemented for the ontology, functional, non-functional requirements are used to design the web-based application. Software components include Protégé-OWL, PROMPT, Jena API, etc. The system architecture represents the layers unto which software components would be executed. Detailed explanations of the system architecture are provided in chapter 7.

5.5.4.2. Prototype Implementation

The prototype implementation stage involves inputting the knowledge model into the different layers of the system architecture. To develop the knowledge base layer (ontology), the developed conceptual UML model is converted into OWL in the ontology editing software – protégé. These corrected data are then enriched using the ontology

development 101 guides. Specifications not provided by commonKADS are determined and added.

It is important to note that although existing domain ontologies on sustainable building technology products and photovoltaic systems contain domain knowledge that can be reused; however, it is important to mention they are an isolated conceptualization of the domain. Most of these ontologies developed solely on knowledge from human domain experts cannot serve their purpose for long since domain terms used in literature differ from one expert to the other. Likewise, building product terms and descriptions evolve as construction product libraries continuously enlarge. Any semantic web solution developed using this knowledge model would be void of rich domain terms formalised and standardised. A novel approach is to retrieve knowledge from domain-specific online product repositories such as the industry foundation classes to get rich domain ontology. This way, terms are standardised and understood by the majority, rather than only the developers. For this study, the ifcOWL ontology was extended.

SBTPs and PV concepts with their corresponding properties in the conceptual model like concepts in ifcOWL ontology are identified, and the terms used in ifcOWL are used for representation. Concepts that are not common and only exist in our conceptual model maintain their current terms. However, new classes have recently been included in the IFC schema IFC2X4_ADD1.exp after the development of the ifcOWL ontology. These new classes are directly extracted from the IFC schema (through inherited definitions from supertypes) and inputted into the ontology editing software protégé. The proposed ontology is expected to have a high level of reasoning to know the extent to which it can recommend sustainable building technology products.

A web-based application provides an intended user with a functional application to use. The agile software engineering methodology is used to develop the graphical user interface design. It is made interactive and can be easily used by end-users with minimal computer science skills. Since this prototype application aims to represent PV systems technology components using the languages of the semantic web (not HTML), a link is generated between the knowledge base layer and application layer through the

integration of four (4) software tools. A Development Environment (IDE) - eclipse, a Java application server – Apache Tomcat, a JavaServer Pages (JSP) application technology, and an open-source framework for building semantic web applications - Jena API. Methods from the Jena API read the content of the ontology in RDF and parse the values to relevant fields into the user interface using hash maps. The system uses these inputted data alongside built-in formulae to make computations. Resulting values recommend PV components.

5.5.4.3. Prototype Evaluation

Once the knowledge model is complete, an Iterative evaluation of the system must guarantee syntactic and semantic correctness. Evaluation, which includes both syntactic verification and validation, assesses the developed ontology's quality and correctness. Verification checks for anomalies, inconsistencies, and inaccuracies in implementation. For instance, errors with instance checking and incompatible data-type properties. Syntactic checking is done using PROMPT, which provides support and techniques for semantic verification of ontology. Eliminating anomalies are also done using ontology-reasoning tools such as FaCT++, Pellet incorporated in protégé. Verification is an iterative process until all errors are removed or minimised.

Validation ensures that developed ontology models precisely what it is created for. Condition tests through case studies are used to guarantee the system achieves the desired outcome. This is similar to the Turing Test of the software development process. Condition-decision matches are done to check that the right decision is made for the right reasons. Similarly, new knowledge is also added to test the system. Temporary classes and predictable faults would be added deliberately to see how the system would behave. The prototype developed must meet system requirements. It is important to note that prototype evaluation is a repetitive process. It is done continuously until satisfied, and the desired application is attained.

5.5.4.4. Prototype Documentation

Documentation is the last stage of the prototype development. It is vital to facilitate reuse, maintenance, and knowledge sharing. It is also necessary for further

implementations or improvements. Ontology has been published online and available on GitHub. Requirements for correctly publishing an ontology on the web have been adhered to. These include using a stable URI, providing metadata, labels and descriptions, links to ontologies that have been reused. Final documentation of the prototype is presented in conferences, journal publications, and the research thesis.

5.5.5. Recommendations for Use in Practice

The proposed prototype system is recommended to be used in practice. Previous studies have developed desktop-based solutions using legacy database languages such as SQL, which do not handle unstructured data well. This study develops a web-based prototype that is scalable and is recommended for use globally. Furthermore, this prototype has a knowledge base rich in domain knowledge. Standard taxonomies were reused and extended during development. The developed application is highly functional, and the graphical user interface is friendly and applicable. Challenges and limitations faced during the research and prototype development are recorded for future research studies and implementations. This prototype could be extended to recommend more sustainable building technology products.

5.6. Conclusion

This chapter has reviewed the most used research methods and developed the research framework to be adopted in the thesis. This chapter elaborated on how this research framework would accomplish the research's aim and objectives, as explained in chapter 1. An exploratory approach is selected to obtain comprehensive knowledge on the domain of sustainable building technology products and the semantic web. The main research framework adopted, commonKADS, incorporates knowledge engineering, ontology engineering, and software engineering to effectively understand how semantic web technologies can manage knowledge on photovoltaic systems. These methodologies were used in the development of a prototype system, the development of a photovoltaic systems ontology, and a PV-selection/recommendation application. The details of the application of these methodologies are presented in the subsequent chapter.

6. Knowledge Modelling (Acquisition and Representation) of Sustainable Building Technology Products and Photovoltaic Technology

6.1. Introduction

An appraisal of sustainable building technology products and photovoltaic technology was conducted in chapter 2. The challenges and possible solutions to barriers in the domain were highlighted. Based on that, an investigation of the extent to which semantic web technologies can offer solutions to the problem domain were examined in chapter 3. This led to the establishment of a methodological framework in chapter 5 that applies ontology development methodologies and software engineering methodologies within commonKADS knowledge engineering framework to demonstrate how the aim and objectives of this study would be achieved. The resulting objective of this study is to develop a semantic web prototype system based on an established knowledge model. A clear understanding of a problem domain is the first step towards development of a knowledge-based system. It is thereby vital to comprehend the complex and vast knowledge available and simplify it into clear concepts.

This chapter follows the justified research methodological framework established in chapter 5 to first develop a generic conceptual model of sustainable building technology products using the Unified Modelling Language (UML). This is followed by a detailed representation of photovoltaic systems technology models which is the focus of this study. Insights gained can be applicable to other areas of sustainable building technologies as part of future research. The developed models in this chapter would be implemented in a software development environment in chapter 7. The following sections elaborate on knowledge acquisition through the three main activities involved in CommonKADS: knowledge identification, knowledge specification, and knowledge refinement.

6.2. Knowledge Identification

Knowledge identification is the first stage of acquisition. The two main activities involved in this stage are domain familiarisation, knowledge analysis, and identification of re-usable knowledge components. The following sections explain each process.

6.2.1. Domain Familiarisation

Domain familiarisation involves processes that ensure that every aspect of the domain is fully investigated. This study reviewed old and new knowledge from four primary information sources to get wholly familiarised. Firstly, literature sources that include books, peer-reviewed publications, dissertations, e.t.c. Secondly, a review of products published by domain standards such as Energy Savings Trust (EnergySavingTrust, 2022), Carbon Trust (CarbonTrust, 2022), sustainable energy technology products classified by Building Research Establishment (BRE, 2022a), and many more, as discussed in chapter 2. The third information sources were seminars and training workshops at Oxford Brookes University, American University of Nigeria and Udemy online training (Udemy, 2022). The fourth form of familiarisation is through consultations with domain experts, who provide in-depth insights to knowledge and standard practices in ways that books cannot specify. Once a thorough understanding of the domain was developed, the knowledge obtained was analysed.

6.2.2. Knowledge Analysis

The purpose of knowledge analysis is to critically assess domain knowledge to fully comprehend and outline the research's essential parts. For example, certain research questions need to be answered. These include how diverse is the domain of sustainable building technology products with focus on photovoltaic technology? What critical areas should this research focus on? What are the founding theories? What are the latest developments and practices? What are the latest arguments in academia on photovoltaic technology products? The answers to these questions are discussed in chapter 2.

Likewise, it is crucial to distinguish and categorise identified knowledge into the domain, inference, and task knowledge (Kingston, Shadbolt, and Tate, 1996; Schreiber et al., 2002; Khankasikam, 2010): *Domain Knowledge* represents explanatory knowledge that gives an in-depth understanding of the dynamics of the domain of research. These are concepts, properties, and relations of all concepts relating to sustainable building technology products. They are relevant and static. *Inference Knowledge* represents concepts that make up the inference layer of the prototype. Knowledge in the layer is inferences that would use domain knowledge to solve the research problem. It also

represents tasks to be performed to make decisions. This knowledge would mostly be retrieved from concept maps that model in-depth knowledge on deductions and conclusions made in the domain, e.g., case studies. *Task Knowledge* outlines procedures that inferences would follow. They are simple tasks, and subtasks are followed to achieve goals. Task knowledge is goal-oriented knowledge.

The outcome of domain familiarisation and analysis divulges the right tools and approaches of domain knowledge to the research's conceptual framework.

6.2.3. Identify Potential Knowledge Components for Reuse

The last activity involved in the knowledge identification stage is to identify potential knowledge models to be re-used. The ontology development 101 methodology selected for this study allows re-use of existing knowledge models. It is also very similar to the development workflow of open-source software, where the development of ontology is made easy, reliable, and 'repeatable.' Relatedly, in a popular article on 'ten simple rules' to ontology development, Courtot, et al. (2016) indicated that the best approach to ontology development is to re-use existing knowledge content as required as possible to avoid duplication of efforts. Furthermore, as explained in chapter 5.7.1.2, re-use of existing knowledge models increases interoperability, reduces time and cost of development, prevents 're-inventing the wheel,' and increases efficiency and consistency. Furthermore, since validated knowledge is identified, their re-use may lead to standardisation. This research identified and reviewed domain taxonomies (in chapter 2) and existing ontologies in sustainable building technology (in chapter 3) to establish the extent to which their validated knowledge can be potentially re-used.

Domain Taxonomies - In chapter 2, an overview of leading research institutions that publish important information on the domain of sustainable building technology products (SBTPs) was done. These include Building Research Establishment (BRE), Energy Saving Trust, and the Carbon Trust. These agencies can be considered as domain taxonomies since they contain standard certification information on SBTPs.

Under the BREEAM (BRE Environmental Assessment Method) certification schemes, the Green Book Live (Bre, 2022b) provides an A-Z catalogue of approved 'green' products that end users can select to validate their commitment to zero carbon emissions. One limitation of this catalogue of products is that it lacks a comprehensive list of the sustainable construction products itself; it only provides the responsible sourcing certificates issued by BRE (company name, certificate and issue date). To find an approved product, one would need to provide the category, company name, product name and certificate number which is tedious and unscalable. However, its classification of renewable energy sources at the top level is beneficial to this study and thereby adopted. Reused concepts are presented in the initial establishment of concepts for sustainable building technology products in section 6.3.2.1.

The carbon trust footprint label appears on the individual products. It tells you if the product is certified renewable, certified carbon neutral, certified reducing each year or certified lower than other products (within the category) in the market. However, carbon trust fails to provide a comprehensive list of the large number of products it has certified. It only provides the details of certification and a list of certified organisations. Similarly, energy savings trusts also do not have a comprehensive database of products, however, it provides key categories of sustainable standards for sustainable building technologies and renewable energy that are vital for this study. The concepts obtained from these classifications are presented in section 6.3.2.1.

Existing Ontologies - Based on the review of the ontologies identified for re-use in chapter 3, this study considers the digital construction energy systems ontology (OntologySpecificationDraf, 2022), the Smart Applications REference (SAREF) ontology (Daniele, Hartog, and Roes, 2015; SAREF, 2020), the PhotoVoltaic Technology ONtology (PV-Tons) (Abanda, Tah, and Duce, 2013) and the ifcOWL ontology (Pauwels and Terkaj, 2016). The extent to which these ontologies are relevant to this study is presented below.

Concepts are obtained from the Smart Applications REference (SAREF) ontology because it is an extensible industry-oriented standard for vertical markets. It is also

considered the primary standard ontology for the internet of things (IoT) that can be reused to develop other standards and ensure interoperability (Moreira et al., 2020). Since smart home appliances fall within the context of this study, the *saref:Device* class and its associated properties and sub-classes are deemed fit for re-use in this study. A smart home device *saref:Device* has a model and manufacturer (*saref:hasModel*, *saref:hasManufacturer*) that has at least one function (*saref:hasFunction* min 1 *saref:Function*) such as washing and a measure of energy unit *saref:Energy* (SAREF, 2020). In the context of this study, home appliances are relevant when determining the total energy load of the building, therefore, only concepts that define the appliances and its energy measures have been reused. Similarly, since SAREF mainly focuses on the device, the study looks to other ontologies to find other relevant concepts that are not covered in SAREF ontology. The concepts re-used from SAREF are presented during the initial establishment of concepts for sustainable building technology products in section 6.3.2.1 below.

As reviewed in chapter 2, the digital construction ontologies capture the most relevant objects and properties that are related to process or people involved during construction projects. The digital construction energy systems ontology narrows the scope down to parameters that define the energy aspect of construction. Some of its concepts and properties are obtained from BIM models represented in ifcOWL. Furthermore, the digital construction energy systems ontology also extends SAREF smart energy (SAREF4ener) by incorporating relevant concepts that define energy systems, devices, profiles, storage and control (SAREF, 2020; OntologySpecificationDraft, 2022). The resulting ontology therefore covers semantic data required to link IoT and BIM. It is for these reasons that some of its concepts are reused in this study. The reused concepts are presented in the initial establishment of concepts in section 6.3.2.1.

PV-Tons is considered one of the most relevant existing ontologies for re-use in the study for the following reasons. Firstly, it is an ontology developed specifically for the photovoltaic systems domain. It used well established concepts of sustainable building technology products and photovoltaic systems technology to develop a conceptual

model for the domain. Secondly, ontology was developed to investigate the extent to which semantic web technologies can model knowledge on the domain. This study is closely related to PV-Tons since both are ontology-related solutions that offer decision support in the PV domain. For these reasons, some aspects of the classification of concepts used to develop PV-tons is adopted and presented in section 6.3.2.1 and 6.3.2.3.

However, there are features that this study requires that are not covered in PV-Tons. Primarily, PV-Tons is a desktop-based solution. It is confined. Since one of the targets of this study is to encourage product manufacturers to represent their products online in semantic web's RDF, this study proposes a web-based application. This means resulting ontology will be scalable. It also means products represented online can be machine processable, can be added as instances within the ontology and can be compared to other equally processable instances by the application to provide decision support and recommendation to the users. Furthermore, PV-Tons ontology was not developed using standard taxonomies. Recall the benefits of integrating semantic web with BIM and IFC explained in chapter 4. Several researchers admit there is a considerable added value when semantic web is used on IFC data models to facilitate complex searches, information exchange, and integration (Petrova *et al.*, 2019). Therefore, state of the art applications must take advantage of the many benefits of integrating semantic web with BIM and IFC to ensure efficient knowledge modelling in the PV domain. The logic-based nature of OWL provides formal semantics to IFC models and offers enhanced reasoning and meaning to its data. Hence the need to integrate IfcOWL ontology into the current research. Since IFC is a standard data model for the construction domain, it is extended in the study to develop an enhanced knowledge model for photovoltaic systems. This decision is logical because only continuous use can make standards advance. The proposed ontology in this study could be used as an official extension of IFC ontology for the photovoltaic technology domain.

All features and concepts re-used from the selected ontologies are outlined in section 6.3.2.1 and 6.3.2.3.

6.3. Knowledge Specification

There are three main steps involved in knowledge specification. These are selecting a task template, initial capture of domain knowledge, and a complete domain specification. The following sections elaborate on each.

6.3.1. Selecting a Type of Task Template

Following the description of task knowledge in section 6.2.2 above, task templates offer a control structure whereby goal-oriented tasks can be completed. They are the reusable combination of model elements that facilitate domain knowledge modelling (CommonKADS, 2020). Therefore, a knowledge engineer follows the template to develop top-down modelling by specifying inference and task knowledge. CommonKADS contains several task templates, each with unique features that make it suitable for particular task types. They are wide-ranging, and the details of each are not required for these studies. However, based on a review of the different templates available in (CommonKADS, 2020), the classification task template is found most suitable for modelling sustainable building technology domain and thereby adopted for this study.

The justification for this choice is as follows. Firstly, the classification task template is most suitable for domains where product categorization is done. For instance, in the field of photovoltaic systems, objects can be physically available for inspection and classification, such as solar module classifications or batteries classifications. Secondly, it has an established domain schema model that explicitly and correctly defines classes for the objects that have been classified. It has a straightforward pruning method where it catalogues all things to respective classes, specifies attributes to each object, gives values to all attributes, and analyses modelled knowledge to remove all inconsistent classes. The developer may decide to use a decision tree, information theory, or user control when selecting attributes (CommonKADS, 2020). Thirdly, the nature of its inference structure is appropriate for applications that aim to assist in decision-making. Queries on photovoltaic systems can be easily deduced using any object-oriented-based

inference engine. It is one of the most straightforward templates, but its task methods are very analytical and goal oriented.

6.3.2. Initial Domain Knowledge Modelling

The main target of this section is to compile all concepts relating to the domain (of sustainable building technology products with an emphasis on photovoltaic systems technology) into a knowledge model so that the proposed system supports inferencing and decision-making in the choice of the technology. In knowledge engineering, knowledge modelling (a problem-solving process) abstracts concepts in two ways; the knowledge engineer either builds the conceptual knowledge model from scratch or re-uses existing knowledge models. Building the model from scratch necessitates domain experts to validate the obtained model thoroughly, which is time-consuming and expensive. However, re-use of existing models does not require meticulous validation from domain experts. Automatic validation techniques existing within the knowledge acquisition processes suffice. Furthermore, PROMT (Noy and Musen, 2000) is both a knowledge acquisition tool and a popular ontology evaluation tool that, when used, would ensure the obtained model is semantically correct. For these reasons, this study re-uses the identified reusable knowledge models discussed in section 6.2.3 above.

The subsequent sections take up an initial knowledge modelling of the domain first from identified existing models, then subsequently from re-usable domain ontologies.

6.3.2.1. Initial Establishment of Concepts for Sustainable Building Technology Products

Initial conceptual modelling is obtained from existing knowledge obtained from domain sources, literature sources, and domain experts, as identified and discussed in 6.2.1. The main concepts within the domain of sustainable building technology emphasising photovoltaic systems are *building* and *energy*. It is therefore imperative to consider all components of building and energy that promote sustainability. A total of eight components have been classified based on their roles to foster sustainable construction. Three of these components are mainly related to sustainable building, *building* itself as a concept, *construction building materials*, and *building construction elements*. Other

identified components such as *building energy (power installations and consumptions)*, *renewable energy technologies*, *household appliances*, *smart home technologies*, and *location* relate to building and energy. *Organisation* connects a building and the building's energy with external stakeholders. These components are discussed further below.

Building: The *building* concept terms the physical structure onto which sustainable building technology products are mounted. Every building has an *address*, thereby relating it to a *geographicalLocation*. Similarly, every building has an amount of *energyLoad*, which links it to the *energy* concept. Furthermore, a building consists of spatial structuring elements within the *buildingConstructionElements* concept is discussed in the following sub-section.

Building's Energy: A building's energy encompasses the overall energy that comes into a building, its consumptions, losses, and everything in-between. Building energy consumptions are mainly heating, ventilation, lighting, and cooling. Buildings and the construction industry combined account for one-third of the total global energy consumption and CO2 emission contributions. Environmental sustainability is key in the research; therefore, the building's energy is considered concerning sustainable building technology products. Although there can potentially be a lot of concepts under the building's energy, the study chooses to abstract only the key concepts that connect a building to its electric energy needs. Where a building's energy is considered, the key concepts are the structural parts of the building onto which the energy technology would be installed (*installation*), the type of energy technology used to power the building (*sustainableBuildingTechnologyProducts*, *generator*, *grid*) the building's electric energy connection type (*gridConnection*, *nonGridConnection*), the building's energy load (*energyLoad*, *energyOffset*, *energyConsumption*), and the energy performance of the appliances used in the building (*energyPerformance*, *householdAppliances*). Concepts obtained from the digital construction energy systems ontology under building's energy are *energyConsumption*, *dieselGenerator* (OntologySpecificationDraft, 2022). These concepts are adequate to develop the initial representation of concepts related to the building's energy.

Sustainable Building Technology Products: Sustainability context of a building relates to the design and management of buildings, processes involved, and technologies used in ensuring good performance, energy and resource efficiency, monitoring and maintenance, and every other aspect that aims to satisfy a construction project without compromising human health, the current environment or future generations. Different categories of sustainable building technology products demonstrate different degrees of sustainability. Each category has unique characteristics such as a high percentage of fossil fuel elimination when used; some contain large amounts of recycled materials, renewable materials, some increase water efficiency, and some largely reduce carbon emissions. The categories of sustainable standards provided by BREEAM are regarding *ecology, pollution, co2 emission, waste, water, energy, nature of materials* (BRE, 2022a). Categories of sustainable standards by energy savings trust involve *supplyChain, wasteManagement, rawMaterials, renewableProducts* (EnergySavingsTrust, 2022). Other top-level concepts of sustainable building technology products (not listed above) that aim to reduce the negative environmental impacts include *biodegradableMaterials, offSiteConstructionMaterials, greenInsulation, waterConservationTechnologies (dualPlumbing, greyWaterReuse, rainwaterHarvesting, drainPipes, electricShowers, waterSavingTaps), wasteMinimization, heatRecovery, coolRoofs, ventilationTechnologies, energySaving, renewableEnergyTechnologies, lowEmittingMaterials, electrochromicSmartGlass, sustainableIndoorEnvironmentTechnologies, rammedEarthBrick, smartSystemTechnology, panellisedSystem.*

Renewable Energy Technologies: This research emphasises renewable energy technologies which generate energy from diverse and abundant sources. Energy savings trust provides a renewable energy classification that includes *windTurbines, solarPanels, biomassHeating, hydroelectricity technologies, microCombinedHeadAndPower technologies* (EnergySavingTrust, 2022). From the A-Z catalogue of green book live (Bre, 2022b), a classification of renewable energy provided include *solarEnergy (solarThermal, solarCollector, photovoltaicTechnology, heatPumps), geothermal, biomass (pelletBoilers, pelletStoves, logBoilers, logStoves, combinedHeat, combinedPower), HydroEnergy (hydroDams, pipes, hydroTurbines), tidal, windEnergy (windTurbines)*

energy systems. Concepts obtained from the digital construction energy systems ontology under renewable energy are *maintenanceCost*, *operationalCost*, *rooftopPVs*, *deviceInstallation*, *renewableEnergySupplyShare*, *capitalCost*, *lifeCycleCost*, *deviceInstallation*, *deviceMaintenance*, *deviceLifecycleCost* (OntologySpecificationDraft, 2022).

Household Electrical Appliances: one of the several methods of determining the energy requirements of any building is through a cumulative sum of the energy consumptions of the household electrical appliances. This study adopts the classification structure of household appliances as used from an enumerated *IfcElectricApplianceTypeEnum* list in IFC2X4_ADDI.exp specification (BuildingSMART, 2022d). These concepts are *dishwashers*, *electriccooker*, *electricheater*, *freestandingfan*, *freezer*, *fridge_freezer*, *handryer*, *microwave*, *refrigerator*, *telephone*, *tumbledryer*, *tv*, *washingmachine*, *watercooler*, *waterheater*. Concepts obtained from the digital construction energy systems ontology under household appliances are *batteryStorage*, *load*, *energyConsumptionQuality*, *lifeCycleCost*. **Smart Home Technologies** use sensors that increase energy management and living comfort in homes. They are mostly computer-based, capable of making automated intelligent decisions. Concepts here include *internet*, *motionDetection*, *security*, *camera*. From the SAREF ontology (SAREF, 2020), actuating concepts such as *doorSwitch*, *lightSwitch*, *motion* and the unit of measure for power, temperature and smoke (*powerUnit*, *temperatureUnit*, *smokeSensor*) are reused. These technologies use context-awareness to reduce energy consumption when needed and can perform real-time data analysis.

Building Construction Elements: The building element is defined by buildingSMART as a single or combination of physically existing, tangible structures or spaces that fulfil distinctive functions within the building (buildingSMART, 2022a). A classification of the main building elements is abstracted from the green guide to housing specification (BRE, 2022b) and comprises *carcass* and *openings*, *circulation and storage*, and *floors*, *walls*, *stairs*, *roof*, *windows*, *insulation*, *landscaping* and *foundation*.

Location: the geographical location of a building is one of the first considerations when one wishes to adopt sustainable building technology products. Most renewable energy sources are intermittent in nature. For instance, hydroelectricity is suitable in locations with massive volumes of water, wind turbines in areas near hilltops or coasts which have frequent robust winds while solar technology is better suited in locations where solar radiation is high and not areas towards the arctic. Furthermore, location dependent concepts are required in determining the solar irradiance of a location. Concepts here include the *longitude, latitude, solarIrradiance, temperature, wind, shading,* and all other *meteorological*

Organisation: The *organisation* concept has been selected and re-used from PV-Tons (Abanda, Tah, and Duce, 2013) as a concept that defines all external components (actors or stakeholders) related to sustainable building technology and photovoltaic systems. For instance, the sub-concepts under the *organisation* would be photovoltaic technology *manufacturer, installer, designer, and client.* However, the study does not include its *business* and *research* concepts because a technology *designer* could be involved both as a researcher and for a business purpose.

The initial establishment of concepts of sustainable building technology products is represented in the figure 6.1 below.

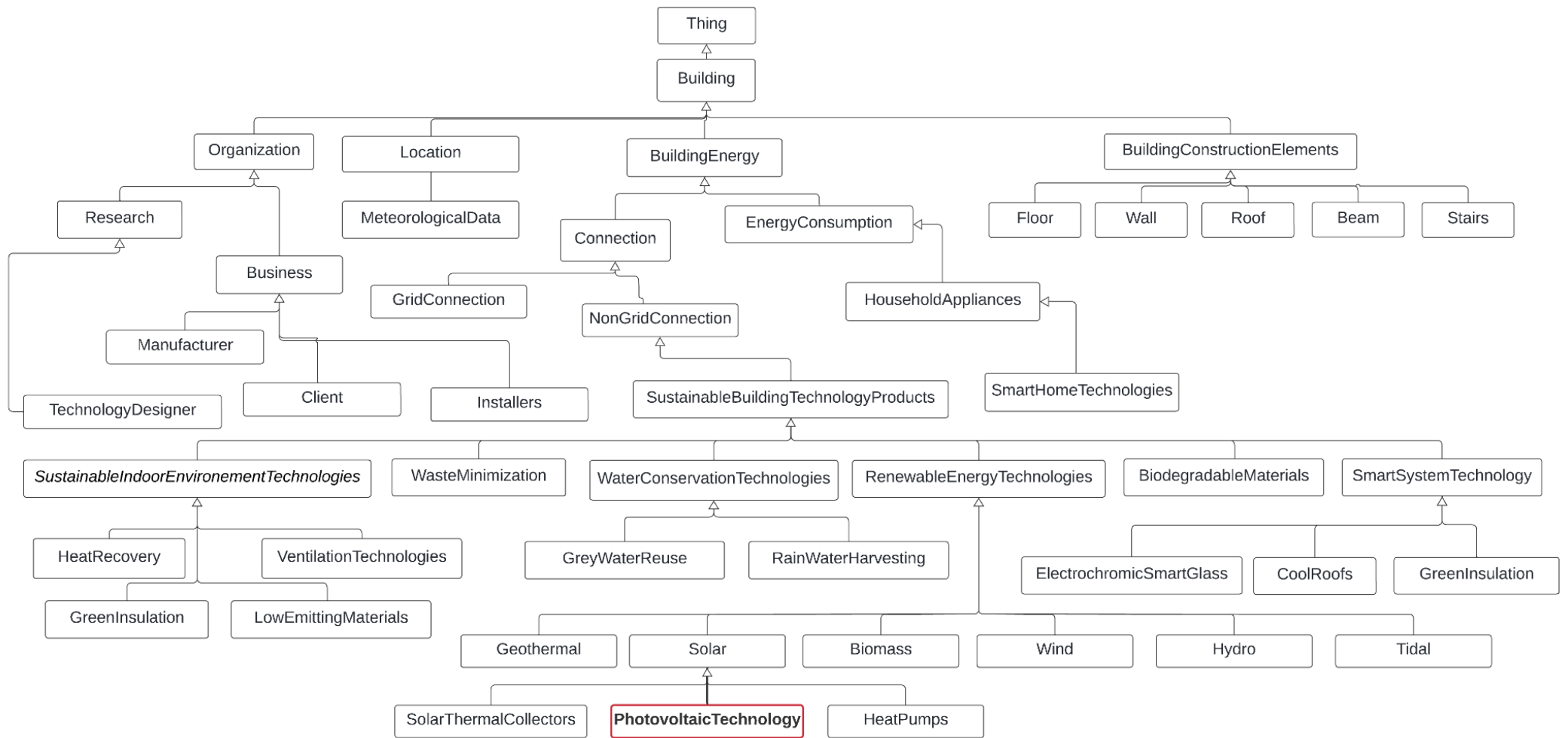


Figure 6.1: Partial Representation of Concepts of Sustainable Building Technology Products Domain

6.3.2.2. Initial Establishment of Concepts for Photovoltaic Systems

Photovoltaic systems technology is the focus of this study; therefore, it is imperative to establish concepts unique to PV-systems to include the design and selection of its several components. It is crucial to outline the knowledge requirements involved in PV-system component design and selections to define the concepts. PV-system output often varies due to different component characteristics (size, efficiency, and performance). Therefore, its design methods must include a detailed understanding of the selection and performance of each component. Furthermore, different permutations for the combination of components must be analysed because various permutations could result in dissimilar performance estimates. The key decision points for the notes in selecting and sizing PV-systems are the location of the building onto which PV technology would be installed, the energy load of the building to be covered and the tradeoff between mounting space, budget and efficiency of the technology.

Firstly, the geographical location of the proposed building is the primary consideration as it determines whether adopting a PV-system would be likely or not. PV-technology has an intermittent nature such that it is highly suitable for locations around the equator and other areas with huge solar radiation. Some locations in countries towards the arctic with little solar radiation may find solar energy inaccessible.

The second vital consideration is the energy load of the building the PV-system would cover. This is because the general size of the PV-system is dependent on the energy requirements or offset. The several methods of determining the building energy load have been reviewed in chapter 2. These methods specify the peak, average and minimum (daily, weekly or monthly) energy load of the building as illustrated (by a sample energy load map of a household) in figure 6.2 below. An energy load guide is available as an appendix A. A key research question is what energy offset should the PV system cover? Will the proposed PV system cover the peak or average energy load of the building?

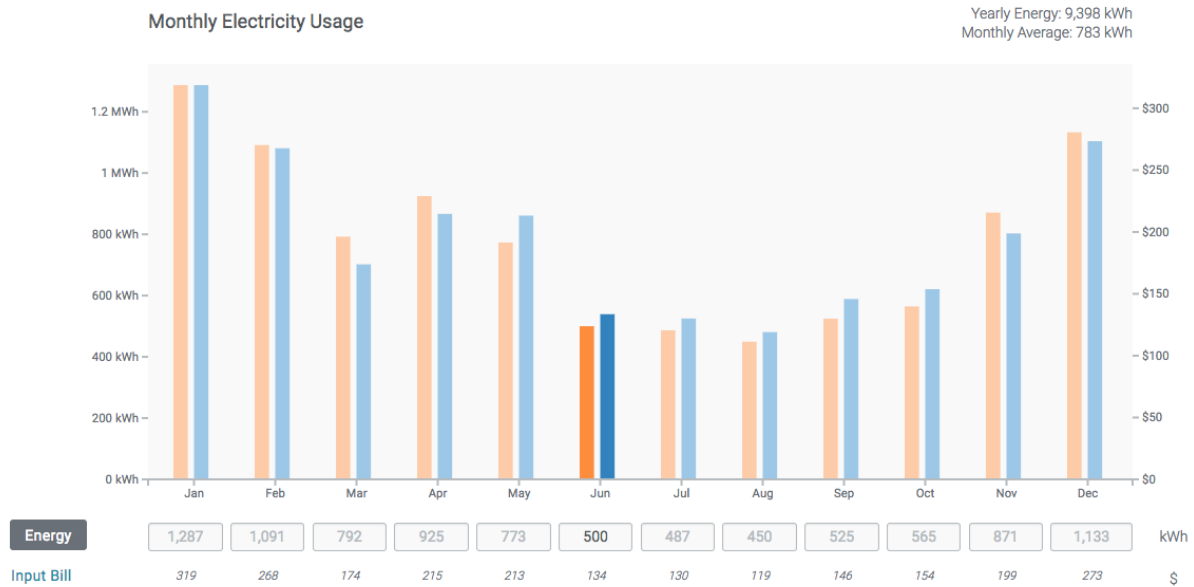


Figure 6.2: Sample Energy Load Map (Aurora, 2023)

The PV components (module, inverter, battery and controller) are proportional to each other and their sizes are dependent on the energy offset selected. Using figure 6.2 above as reference, if a PV-system is designed to meet the peak energy load, then the energy needs of the building would be satisfied a hundred percent the entire year. However, this means apart from the month of January, there would always be excess electrical energy that would require a large battery bank to store. Similarly, the entire PV-system will be oversized and expensive. If a PV-system is designed to meet the minimum load, then the PV-system will be small and cheap but an auxiliary energy source (such as the grid or a diesel generator) will be required for an extensive period. If the proposed PV-system aims to satisfy the average energy load, then energy generated would be sufficient for about seven months out of the year. Little excesses generated from the months above average energy need could be stored in a battery bank to use where the energy load of the building is higher than the electric energy generated by the PV-system. The amount of energy to be offset is entirely up to the user. The proposed system in this study establishes the minimum components' sizes required to generate the electrical energy output the user chooses.

The third decision point is the tradeoff between efficiency of the technology, the budget and the mounting surface area. High efficiency modules are often adopted because they

generate more electric output than a similar sized module of lower efficiency. They also require smaller mounting space. However, the higher the efficiency of the technology, the more expensive it is. Users with large mounting space often opt for less efficient modules to cut down price but where the mounting surface area is small, the most efficient modules are selected. Price, space and efficiency trade-offs are inevitable when sizing PV components.

This study proposes a comprehensive process/decision model (figure 6.3 A, B, C, and D below) with detailed steps essential to the design and selection of PV-systems. The developed decision model has a unique advantage; it is backed by mathematical computations that support the design process that has been developed and is presented. Details of these calculations are presented below the process model while applications of the computations are available in the implementation chapter. These calculations would have been unmanageable to capture through the routine procedures in ontology development. The researcher authored publications (Usman *et al.*, 2020a; Usman *et al.*, 2020b) and critically evaluated the numerous existing design methods used for PV-systems selection and their effects on the performance of installed PV-systems. The publications presented case studies and justification that attest to the efficiency of the proposed selection approach in this study.

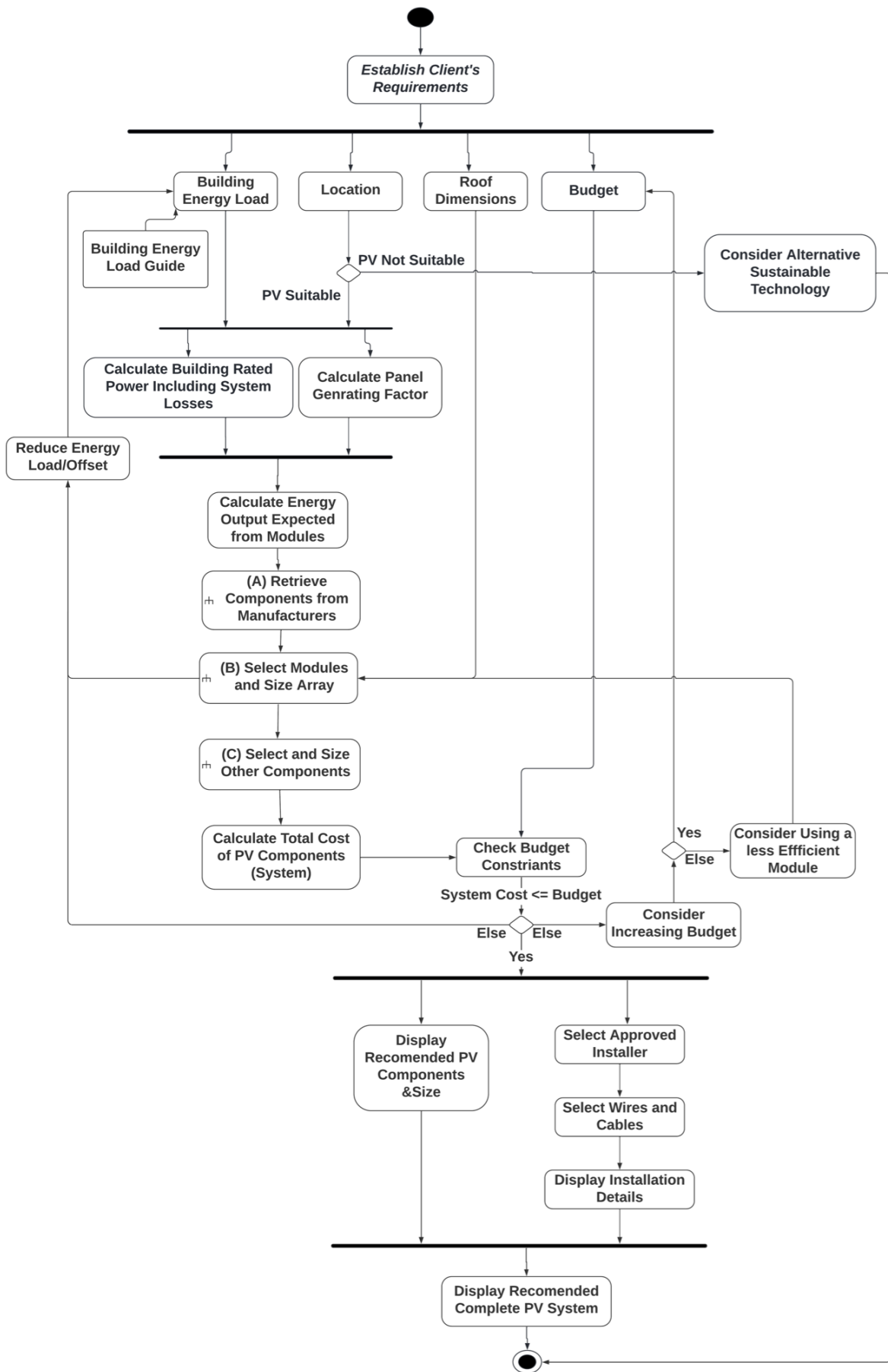


Figure 6.3 (A): Decision/Process Model for Selecting Suitable PV-Systems

**(A) Retrieve Components from
Manufacturers**

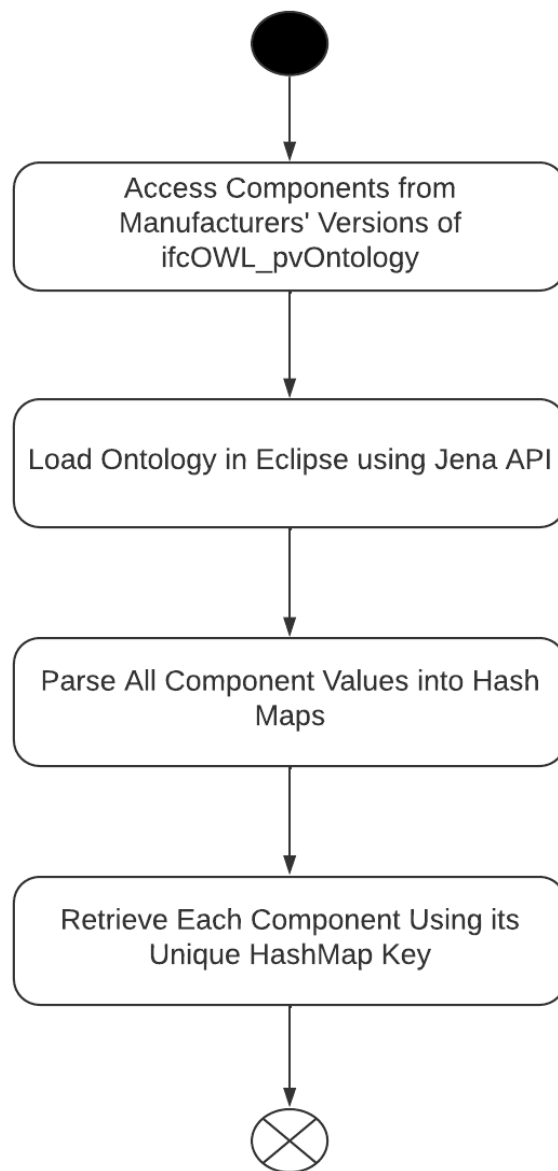


Figure 6.3 (B): Retrieving Components from Manufacturers

(B) Select Modules and Size Array

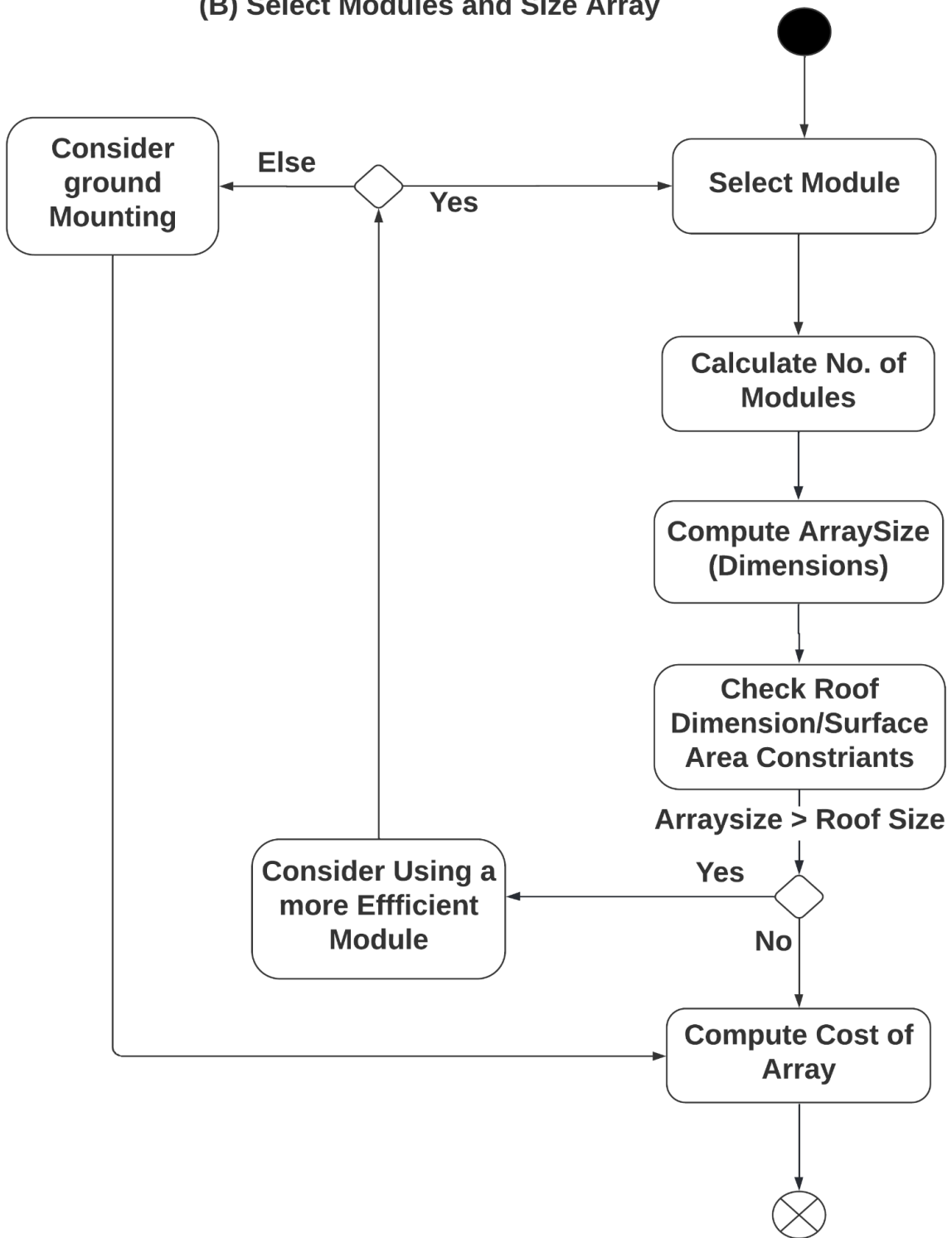


Figure 6.3 (C) : Selecting and Sizing of Modules and Array

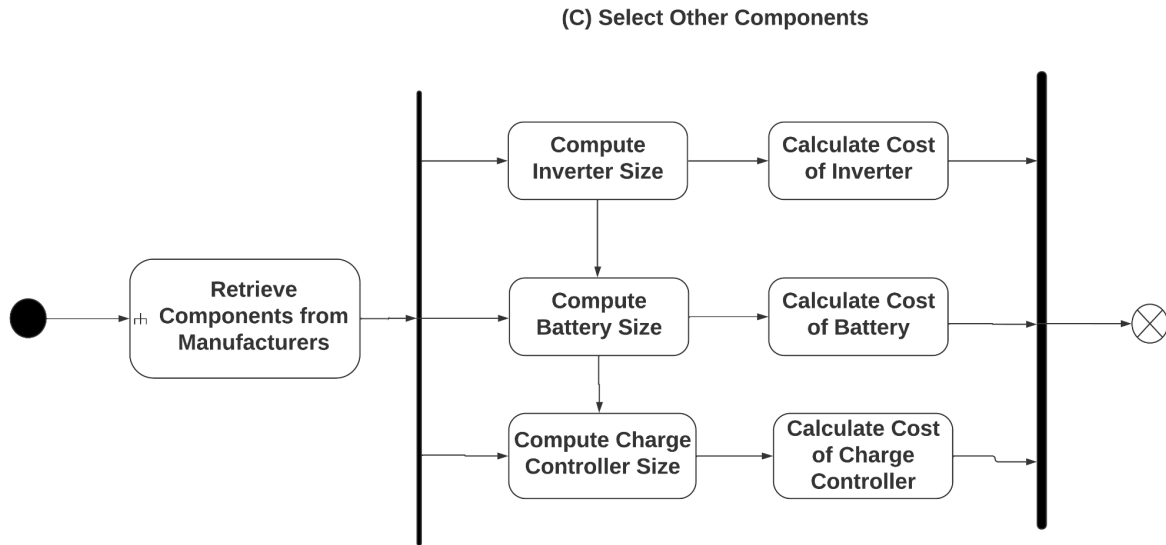


Figure 6.3 (D) : Selecting Other Components

The mathematical computations derived to support the process model are based on the thorough design recommendations provided in chapter 2. Watt-Hours needed from modules based on building energy requirements;

Watt - hours needed from PV modules (W_M) = building energy load $\times 10^3 \times 1.3$ (where 1.3 accounts for system losses), ... (iv)

Watt Peak rating of modules needed;

Panel Generating Factor (PGF) = Solar irradiance \times (Losses due temperature, dirt, poor radiation and ageing), ... (v)

Total watt-peak rating (W_p) = Watt-hours needed from PV modules (W_M) \div Panel Generation Factor (PGF), ... (vi)

Array Size;

Array size = Total watt-peak rating (W_p) \div Watt Rating of Selected Module (W) ... (vii)

Inverter Size;

Following the inverter design recommendations in (in chapter) inverters should be 25–30% more than the energy load.

Inverter Size = buildingEnergyLoad × 1.25 (kVA), ... (viii)

Battery capacity = (buildingEnergyLoad × days of autonomy) ÷ (nominal voltage × depth of discharge × 0.85) (Ah); where an 85% performance rate is expected of the battery due to system losses in the system. ... (ix)

The proposed decision model laid the foundation for the founding of the main concepts for the PV-system domain. These include top-level concepts such as *building, roof, location, Installation, manufacturer, installer* and a group of PV technology components such as *module, panel, inverter, battery, chargeController, wires, cables and fuse*. These concepts established from the process model led to the development of the initial conceptual model of PV-systems.

Furthermore, to bolster the PV-system domain knowledge model, concepts were extracted from existing domain ontologies. The main components identified and abstracted play additional roles in sizing of components and improving performance of PV systems as explained in the ensuing section. These components make up a group of classes, and sub-classes that have been modelled in a UML diagram (figure 6.4) shown below. The UML diagram provides a hierarchical representation with clear visualisation of the concepts and the relationships they have to each other.

6.3.2.3. Concepts Reused from Existing Domain Ontologies

In chapter 3, an overview of several existing domain ontologies with the potential to be re-used in this study was provided. In section 6.3.2, reasons are provided for the choice to reuse knowledge from existing domain ontologies. In this subsection, concepts reused from PV-Tons ontology are provided. Furthermore, the pattern of extending the ifcOWL ontology is introduced.

PV-Tons has a scenic graphical representation of the relationship between modules, panels and arrays. We have established from the review of literature that a solar module

is a small unit of a solar panel. Therefore, concepts such as *solarPanel*, *subarray*, *Array*, *ArraySubField*, are retrieved. Similarly, all characteristics of a solar module that distinguishes one from another becomes the features of the resulting panel. The *organisation* concept is adopted from PV-Tons with details of the external components related to the PV Technology domain such as *manufacturers* and *installers*. Other concepts obtained include *householdAppliances*, *maxPowerPointTracker*, *earthing*, *mounting*, *client*, *technologyDesigner*, and *meteorologicalData*. The hierarchical classification of these concepts adopted from PV-Tons are illustrated in blue colour in figure 6.4 below.

The main contribution of IFC to this study is its standard taxonomy and unique standardised representation of building's data (i.e., its implementation structure within the ontology). In terms of its standard terminology, previously established concepts such as *roof*, *building*, *householdAppliances* and *buildingConstructionElements* are modified to *IfcRoof*, *IfcBuilding*, *IfcElectricAppliance* and *IfcBuildingElement* respectively which are the standard terms in IFC. Adoption of IFC terms means interoperability would be ensured in the proposed system application. There would be effective data exchange and re-use since standard taxonomies are compatible within the majority software applications.

IFC's unique and standardised representation of building's data can be observed in the latest IFC specification IFC2X4_ADD1.exp (buildingSMART, 2022d). It is important to note that IFC's hierarchical specification of solar technologies stops at *IfcSolarDevice* (IFC, 2021). The *IfcSolarDeviceType* defined is meant to state the different types of solar devices (technologies). IFC enumerates these types as *solarcollector*, *solarpanel*, *userdefined* and *notdefined*. However, this part of the taxonomy causes a challenge for us.

Firstly, we have established in previous sections that a *solarpanel* is a PV technology component containing an arrangement of several solar modules. Similarly, we have also established that the PV technology has several other components aside modules and panels such as inverters and batteries. Representing the *solarpanel* as an immediate type of *IfcSolarDevice* does not cover the entire scope of PV technologies, and their

several other types and components. It therefore narrows the scope of the domain. Secondly, IFC only “lists” *solarpanel* as a predefined type of *IfcSolarDevice* and only describes it in its documentation as a port nesting without representing it as a subclass. IFC delimits ports to “indicate possible connections to other objects according to specified system types...” (IFC, 2021b). Consequently, listing *solarpanel* as a type of *IfcSolarDevice* is subject to variation. Thirdly, and fortunately, the *userdefined* function allows a user to define and therefore extend IFC specifications.

Hence, due to the three vital reasons mentioned above, this study proposes to map *PvTechnology* as a type and subclass of *IfcSolarDevice* and modifies *solarpanel* to be a type and subclass of *PvTechnology*. This is required in order to fit the context of this study. This is shown in the conceptual model in figure 6.4 below. Further details of this representation can be found in the implementation chapter. All subsequent concepts defined under *IfcSolarDevice* are unique to this study in the hope that the additions form an extension to the standard IFC specification and becomes a foundation to develop a PV technology ontology inspired by the IFC standard. Concepts re-used from IFC are illustrated in red in figure 6.4 below.

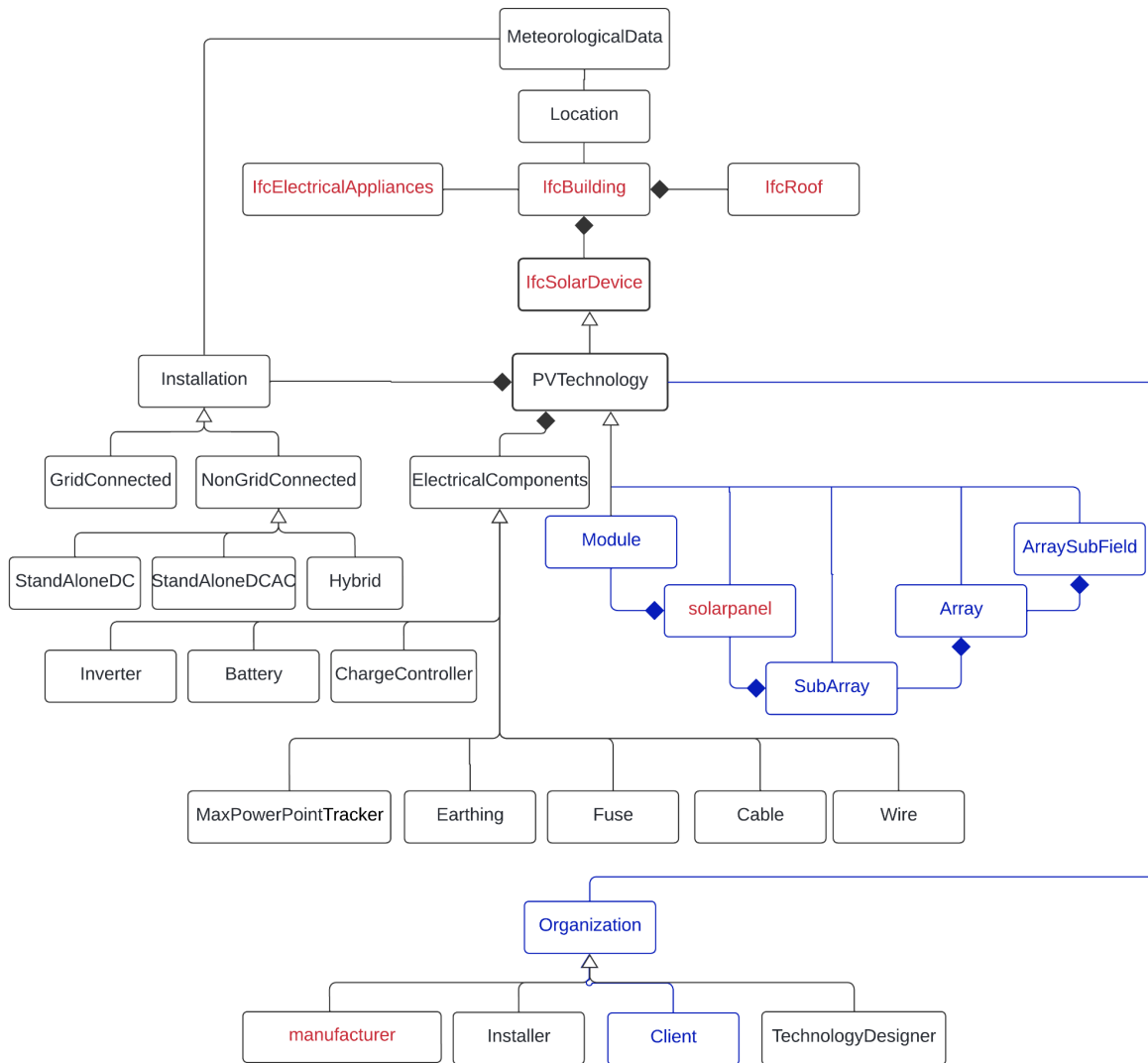


Figure 6.4: Partial Conceptual Modelling of PV-System

6.4. Complete Conceptual Knowledge Specification for the Domain

This stage consists of three main steps; determination of annotation, object, and data properties to complete the conceptual knowledge model of the sustainable building domain with greater emphasis on PV-systems.

6.4.1.1. Establish Annotation Properties

One of the many advantages of the semantic web is that it allows the representation of knowledge-based annotations of web resources. The OWL language used to express ontology languages itself is a web resource and requires annotation. Annotation on OWL ontology is used to document authorship and all other information related to the

ontology, classes, object and data properties, imports, and many more. Protégé-OWL editor has some existing annotation properties built-in, such as labels, and has an efficient way of adding more annotation properties and adequately tag them to the suitable ontological component. The annotation properties for this research will be added during the prototype implementation stage, discussed in chapter 7.

6.4.1.2. Determine Object Properties

Object properties in Protégé-owl are defined as an instance of class owl:ObjectProperty. It describes the relationship that links one concept to another. Using figure 6.4 as a reference, object properties tell us the relationship that the *PVTechnology* has with the *organisation*, the *building*, the *roof*, or *PVInstallation*. For instance, a *roof* is *aPartOf* a *building*, and a "*building* *instals* *PVTechnology*." However, it can be described as a *building* that also *consumes* the resources generated by the *PVTechnology*. Therefore, we select a property definition that best describes the relationship between the two concepts. Furthermore, it is essential to note properties when it comes to inherited concepts. For example, a technology designer *designs*, a manufacturer *manufactures*, an installer *instals*, and a client *deploys* *PVTechnology*. Given that these are sub-classes of *organisation*, it is imperative to use an object definition that would suit all subclasses, such as *isInvolvedIn*, which suggests the *organisation* relates to the *building* through several ways. The next consideration when defining object properties is inverse properties. For instance, a "*building* *contains* *householdAppliance*," the inverse property definition would be "*householdAppliance* *isContainedIn* *building*".

6.4.1.3. Determine Data Properties

We have proven how object properties described the relationships between concepts. On the other hand, data type properties describe properties of concepts established within the knowledge model. Data-type properties will play a vital role in this study as they form the basis to which the client may use to make an informed decision on selecting PV-systems. These properties have literal defined data types such as strings, integers, booleans, e.t.c., therefore their values would be compared, and decisions would be taken based on the values obtained. For instance, the *PVTechnology* concept could have a data-type property *hasWarranty*. In plain English, this could be "A *PVTechnology*

hasWarranty of 10 years". Similarly, this value given to the warranty may be a decision factor compared to other PVTechnology with a smaller number of years of warranty.

Several data-type properties are polyonymous; they are known by various names. Firstly, property definitions of products differ from manufacturer to manufacturer. Most often, some manufacturers have more properties of a PVmodule, than another. Therefore, it became necessary to merge captured properties together. Secondly, there were issues regarding synonyms where different names with the same meaning were used to denote a particular property. For instance, in terms of the energy output of the PVTechnology considered in this study, manufacturers had different names such as rated output, rated power, product maximum power, and many more. This study adopts rated power. These properties are merged based on the semantic aggregation proposal by Noy and Musen (2002). For instance, consider a situation where two manufacturers A and B, have two different defined properties.

Manufacturer A: PVTechnology A = {name, type, ratedPower, cost}

Manufacturer B: PVTechnology B = {name, type, efficiency, weight, expectedServiceLife, moduleSurfaceArea}

Using the semantic aggregation proposal by Noy and Musen (2002), a union of the properties from both manufacturers is expected. Therefore,

PVTechnology A \cup PVTechnology B = {name, type, ratedPower, cost, efficiency, weight, expectedServiceLife, moduleSurfaceArea}

It is imperative to note that sub-classes would inherit all object and data-type properties of their parent classes. For instance, *PvTechnology* would inherit all properties of *IfcSolarDevice*.

6.5. Alignment of Datatype Properties using IFC Schema Specification

The next stage is to align established datatype properties with re-usable datatype properties defined in IFC schema. We have already established the benefits of re-using the IFC schema in section 6.3 above. Four main considerations were undertaken below.

First and foremost, concepts and their corresponding properties that have already been established are directly compared to concepts within the IFC schema using the following three steps:

- i. Data type properties that are similar are identified and the terms used in IFC are used for representation. In Figure 6.5 (A) below, all datatype properties in black are properties that have been initially established. In (B), datatype properties in red are properties that exist in both the initial conceptualization and in IFC schema; but the terms used in IFC have been adopted. For example, *weight*, *warranty* and *expectedServiceLife* have been dropped to adopt *solarDeviceWeight*, *warrantyPeriod*, and *serviceLifeDuration* respectively, which are the terms used for their representation in IFC.
- ii. Properties that are not common and only exist in our initial conceptualization maintain their current terms. In Figure 6.5 (B), it can be noted that names of properties such as *name*, *ratedPower*, *cost*, *manufacturer*, *componentEfficiency*, *manufacturer*, *moduleSurfaceArea* (denoted in black) are maintained. These properties are not defined within the IFC schema but are highly relevant within the context of the study.
- iii. New properties (non-existent in our initial conceptualization) but available in IFC specifications that are relevant, are adopted and re-used. These are *assessmentCondition* and *environmentalImpactIndicators*. The assessment of the product's condition and its contribution to hazardous waste, greenhouse gas emissions, or inert waste generated are vital.

Secondly, IFC gives formal type definitions with standard international units. It defines basic and simple values under *IfcSimpleValue*, a select type for basic types of ISO 10303-41 under *IfcMeasureValue* and types that are derived through formula under *IfcDerivedMeasureValue* (buildingSMART, 2022d). IFC's data type definitions are standardised; therefore, this study adopts the literal type values of IFC throughout its conceptualization. For instance, the literal value for *moduleSurfaceArea* in Figure 6.5 (A) has been modified from *double* to *IfcQuantityArea* in B and the value for *ratedPower* from *double* to *IfcEnergyMeasure*. The *IfcEnergyMeasure* is a real value that indicates a

measure of energy either required, generated or used by the product (buildingSMART, 2022d) measured in Joules (J, Nm).

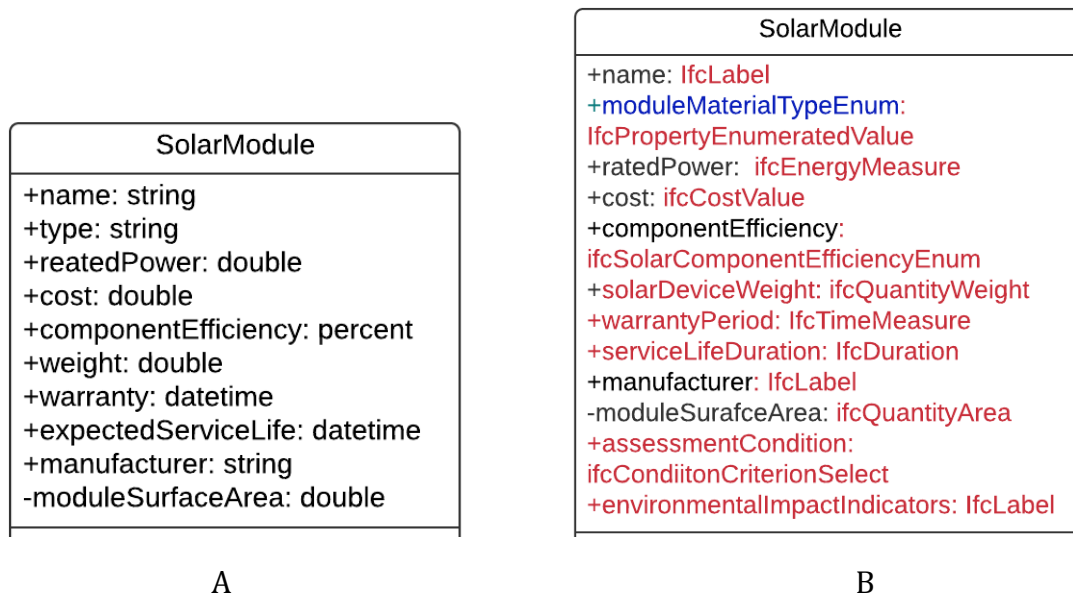


Figure 6.5: (A) Initial Conceptualisation of Solar Module (B) Conceptualisation after extending IFC specifications.

Thirdly, IFC’s data type definitions are comprehensive and highly detailed. An example of such many instances is the property set definition for *warranty* in IFC2X4_ADD1.exp (buildingSMART, 2022d). Several sub properties are defined such as *warrantyIdentifier*, *warrantyStartDate*, *warrantyEndDate*, *isExtendedWarranty*, *warrantyPeriod*, *warrantyContent* and *exclusions*. The study adopts *warrantyPeriod: IfcTimeMeasure* to describe the warranty of the PV technologies defined. *WarrantyPeriod* encompasses *warrantyStartDate* and *warrantyEndDate*. Additionally, the other sub properties such as *exclusions* are not required for the purpose of this research.

Fourthly, this study defines some data type properties (that are not available in IFC) in a similar manner to IFC definitions. IFC’s specification of solar technologies stops at *IfcSolarDevice* and does not define the various components and types of PV technology. Therefore, this study defines several classes and corresponding properties relevant to the PV technology but following the IFC context of representation. An example is an IFC “enumeration” property defined to represent the various types of solar module materials *moduleMaterialTypeEnum* denoted in blue in Figure 6.5 (B) above.

IfcPropertyEnumeration provides a list of unique values that are either numeric or descriptive. Therefore, *moduleMaterialTypeEnum: IfcPropertyEnumeratedValue* defines a unique enumeration list of module types such as amorphous, monocrystalline, polycrystalline e.t.c. Similarly, the ifcOWL ontology is a general building model ontology, additional concepts defined in this study would make the resulting ontology modular and unique to PV technology and its applications. Figure 6.6 below shows a top-level conceptual model showing both object and data properties.

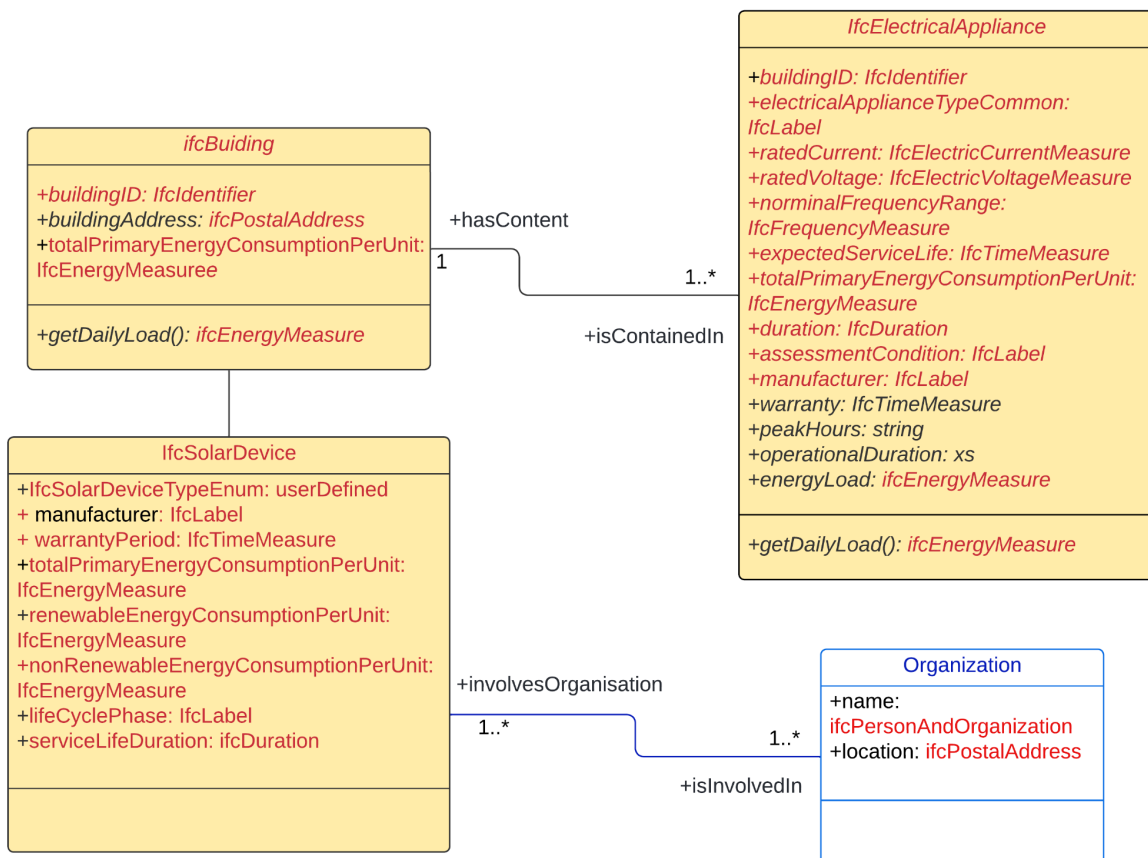


Figure 6.6: Top - Level Conceptual Model Showing Object Properties and Data Properties

The complete conceptual model of PV technology is provided in Figure 6.7 below. Details for these definitions are available in the implementation section in chapter 7. It is our hope that additional definitions in this study are considered by the buildingSMART working group to be added as an extension of the standardised IFC.

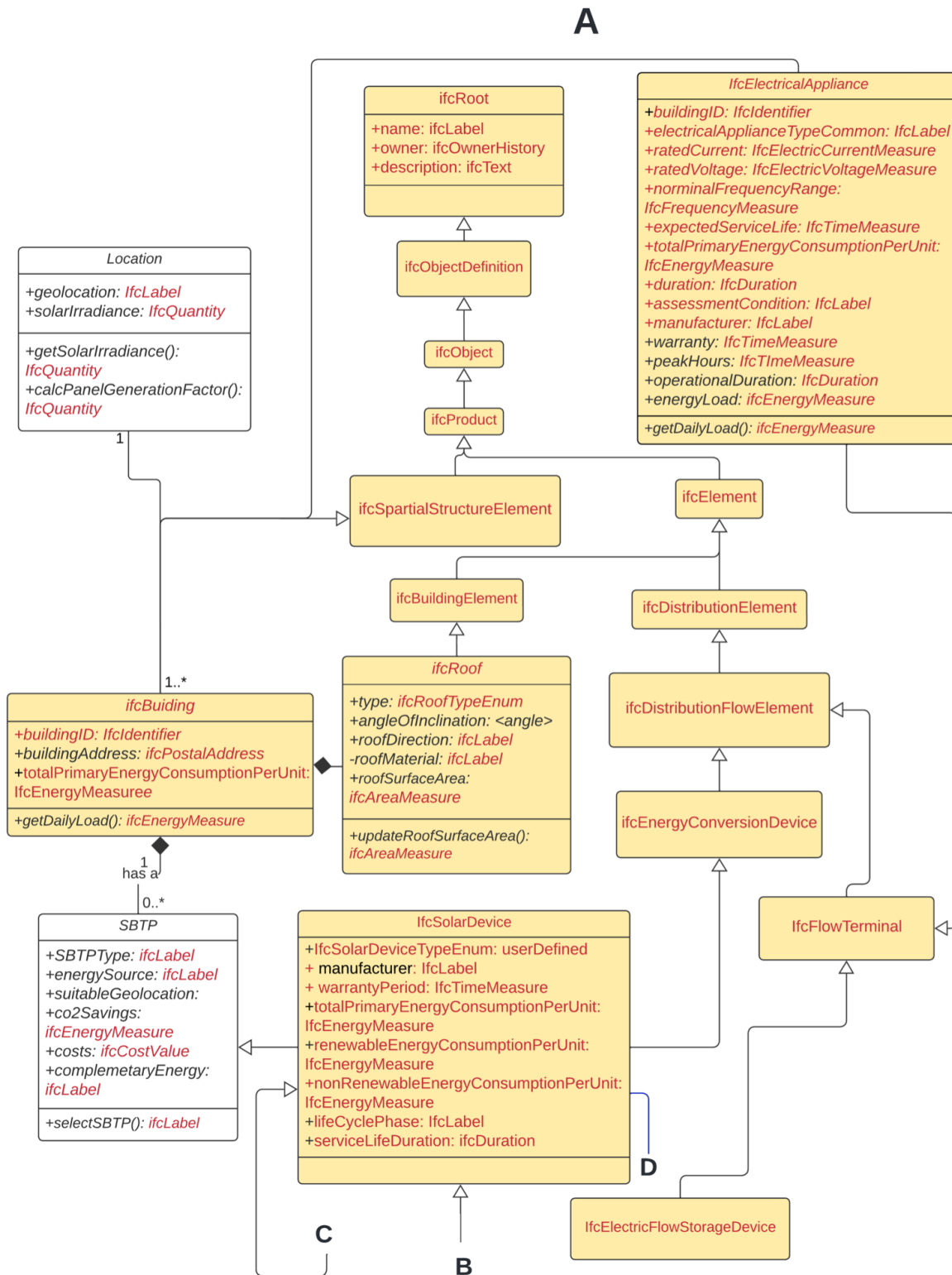


Figure 6.7 (A): Complete Conceptual Model of PV-Technology

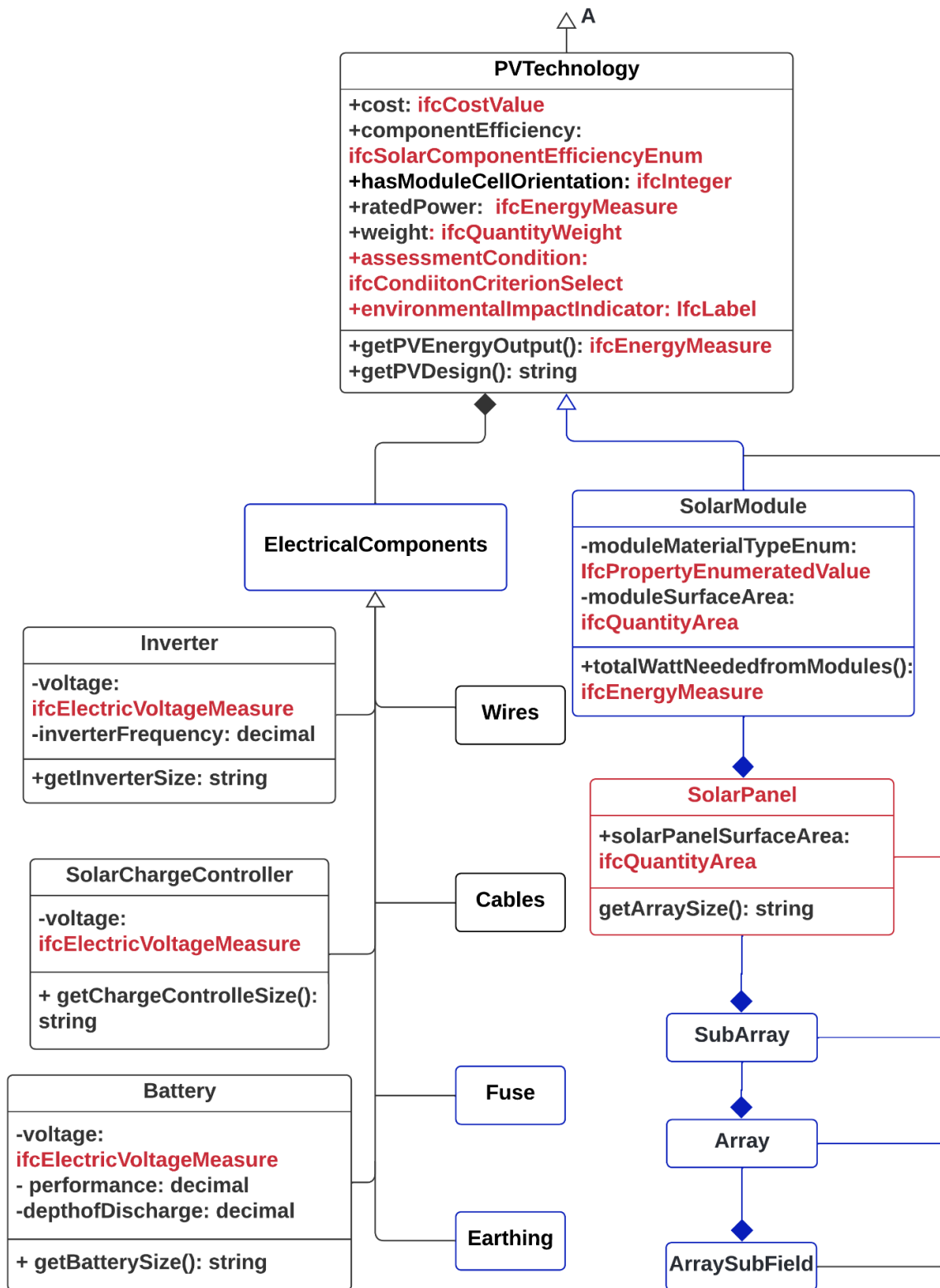


Figure 6.7 (B): Complete Conceptual model of PV-Technology

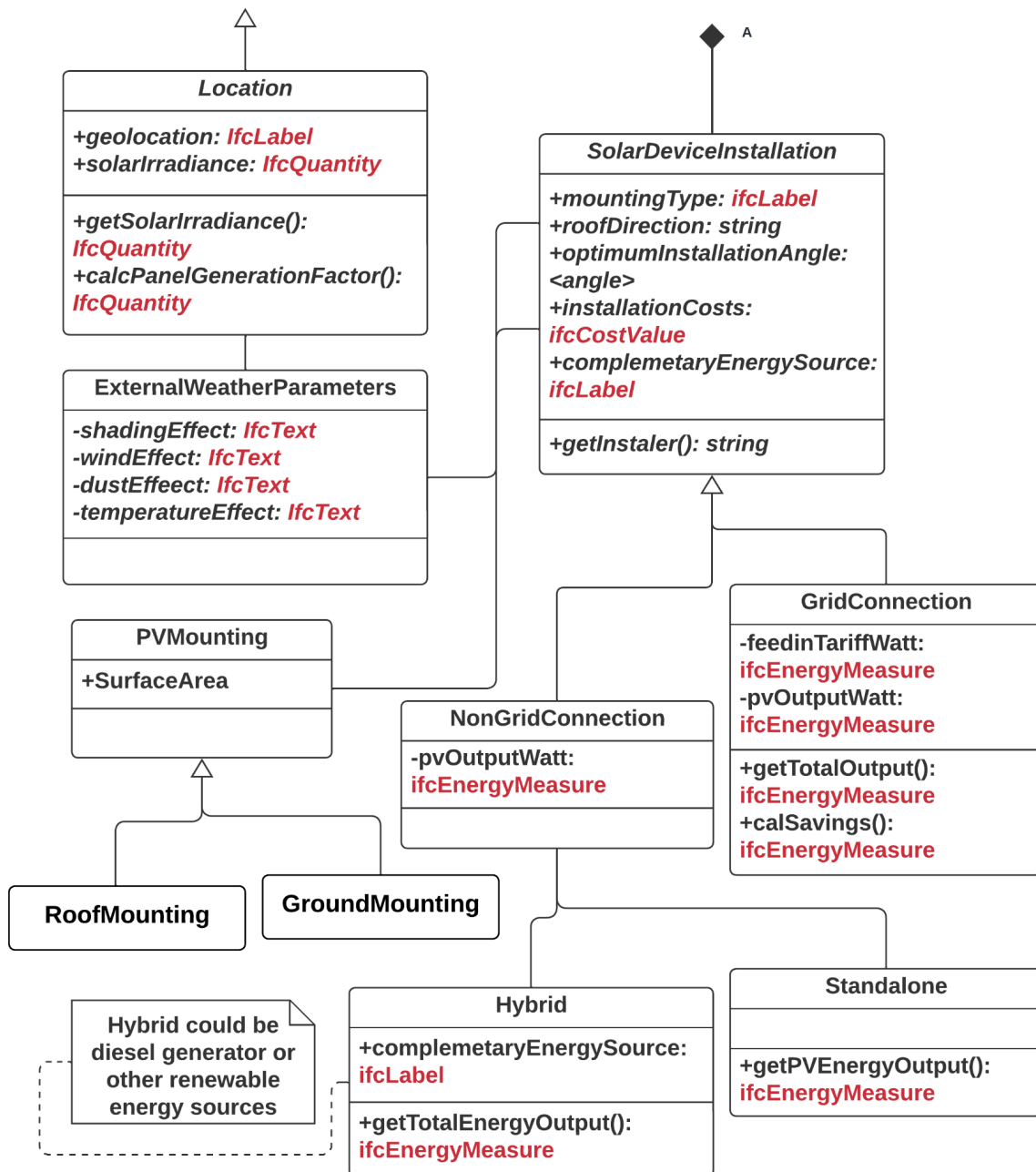


Figure 6.7 (C): Complete Conceptual model of PV-Technology

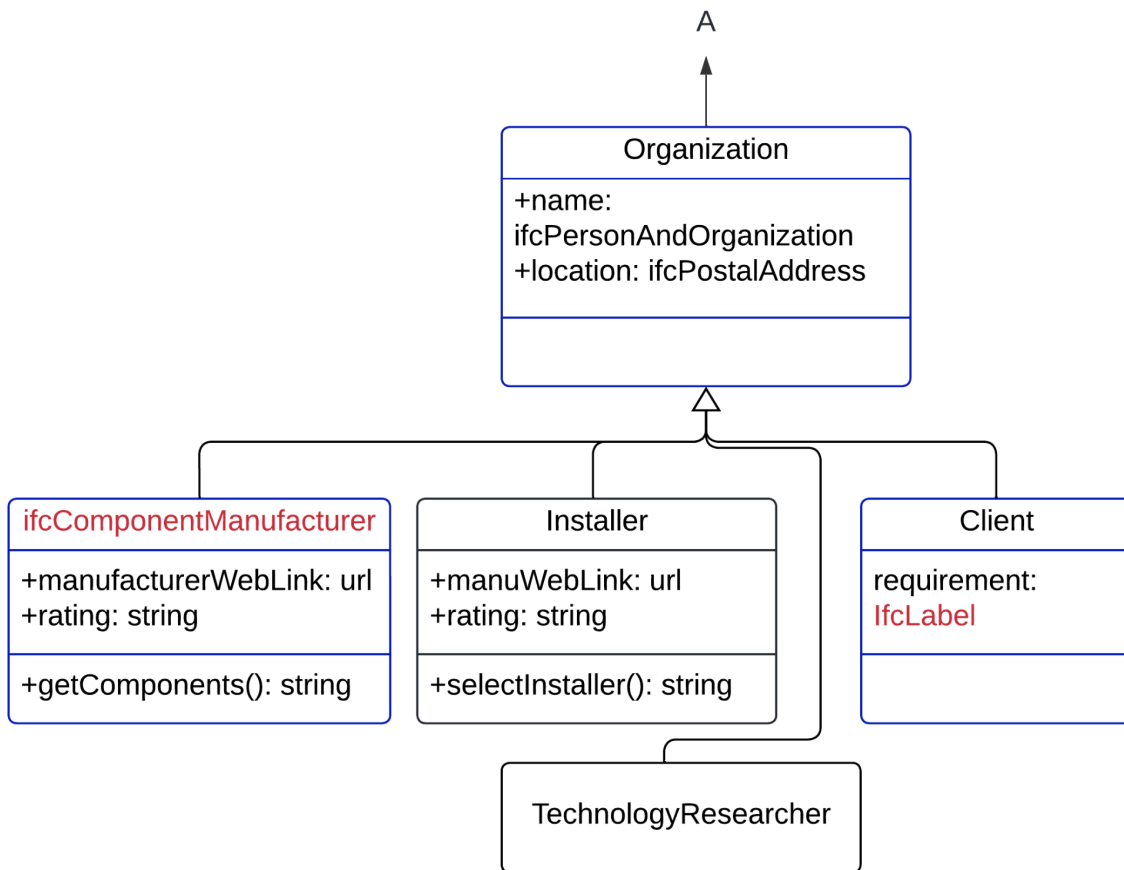


Figure 6.7 (D): Complete Conceptual model of PV-Technology

6.6. Knowledge Refinement

As elaborated in the research methodological framework in chapter 5.7, two main activities are involved in the knowledge refinement stage. These are syntactic and semantic domain verification and a complete knowledge refinement that involves three steps; adding any left-out knowledge, correcting errors found during verification (if there are) and populating the knowledge model with instances. Semantic and syntactic verification of the knowledge model is done at the implementation stage in chapter 7. Implementation will be done in a software environment, therefore populating the knowledge model with instances will be efficient and applicable.

6.7. Instance Acquisition for Ontology Development

Instances in an ontology are objects within the domain. For example, M1 may be a solar module which has unique specifications and properties that are defined by its (solar device component) manufacturer. Other instances could be the different manufacturers and installers of solar device components. Upon completion of ontology development, instances of the domain would be populated into the proposed ontology through the use of the protege-OWL editor in chapter 7.

One of the targets of this study is to encourage manufacturers to represent their products using standard terms, in machine processable formats. Therefore, the proposed ontology is made available online so that different product manufacturers can populate the ontology with their instances. Once instances are added, further implementation can be conducted to provide decision support and recommendation to the users based on the instances added by the manufacturers.

6.8. Conclusion

The knowledge acquisition chapter followed the CommonKADS knowledge engineering methodology to develop the conceptual model of sustainable building technology products and subsequently photovoltaic systems technology. The unified modelling language UML was used to capture the conceptual models. The PV technology conceptual model was developed by extracting concepts from an established process

model that outlined vividly the steps involved in sizing and selecting PV-systems. The process model enabled the acquisition of concepts that would have been challenging to obtain from ontology development methods. Other concepts were extracted from reusable components from BRE, carbon trust and energy saving trust and finally from IFC schema specifications. Furthermore, the chapter shows how this study re-uses IFC's standard terminology on established concepts, property names and values. IFC's unique and standardised representation of PV technology *IfcSolarDevice* has been adopted and extended to contain a unique hierarchical specification of PV technology and its several components. It is our hope that additional definitions in this study are considered by the buildingSMART standards working group to be added as an extension of the standardised IFC.

The UML models developed in this chapter would be converted into OWL in an ontology editing software; protégé. The Annotation, data and object properties acquired through the commonKADS methodology would also be modelled in the software editor because ontological components are best handled within a software environment. Numerous challenges arise during implementation. These include restrictions and specification of rules on data type properties and their values. Furthermore, challenges develop when instances are abstracted from manufacturer websites. This is because manufacturers represent their technology (products) using HTML. In this format, the instances cannot be semantically exploited. How can the study have PV technology represented in RDF instead of HTML by manufacturers? How will instances be added? How can the system recommend PV technology from amongst the ones represented on a manufacturer's website or amongst several manufacturer websites? These challenges would be addressed in chapter 7.

7. DESIGN AND DEVELOPMENT OF PROTOTYPE SYSTEM

7.1. Introduction

In chapter 6, the conceptual knowledge model of sustainable building technology with emphasis on photovoltaic systems was developed and presented using Unified Modelling Language (UML) diagrams. This chapter focuses on developing a complete semantic knowledge model for the problem domain. The target of the implementation in this study is 2-fold. Firstly, a modular ontology is developed for building data that can be extended towards the PV domain and inspired by the IFC standard. This ontology is named the *IfcOWL_pvOntology*. Secondly, a web-based application named the Semantic IFC-inspired Photovoltaic System (SIPS) is proposed, which reads in the developed ontology and provides recommendations on photovoltaic technology products to the system users. This chapter concentrates on developing *IfcOWL_pvOntology* while chapter 8 focusses on implementation of the prototype system.

Development of any software application entails establishment of system requirements. This chapter presents the desired functionalities and requirements expected from the system. Software components and tools adopted for use are also presented. A comprehensive system architecture is developed which can be referenced for each stage of the implementation. Based on this architecture, the chapter shows how the ontology is developed in an ontology editing software protégé-OWL. Because proposed ontology is IFC based, this chapter explains processes involved in integrating IFC and semantic web. The IFC schema has several segments made up of a group of entities that focus on specific domains. These are called Model View Definitions (MVD). MVDs play a vital role in IFC implementation and specifically in this study. This chapter explains the several MVDs and presents the MVD adopted for this study. Other unique IFC specifications and file types that support application development were also presented. The chapter further presented methods of integration of IFC and semantic web through these specifications. These include conversion of IFC's EXPRESS schema into OWL; generation of RDF graphs from the IFC STEP Physical File Format (IFC-SPF) files of the ISO 10303-21 that are mainly adopted for BIM data exchange (discussed in chapter 4); and use of IFC to ontology converters.

The ontology was developed by adopting the steps outlined within the ontology development 101 methodology. One of the challenges is to effectively transform the conceptual model developed in chapter 6 into corresponding OWL entities. Therefore, a set of standard transformation rules are adopted to transform UML models into OWL. Processes involved include techniques used to import and extend IfcOWL ontology, mapping of corresponding OWL classes and property values, adding facets of classes, enhancing the knowledge model with restrictions and cardinalities, and addition of instances. The main challenge faced is correctly extending the IfcOWL ontology with our established knowledge, making sure to follow the IFC structure. Decisions had to be made regarding terms, value types, and enumerations that needed to be maintained and terms that needed to be changed to follow the context of IFC.

Due to the large size of the OWL model, only screenshots of key aspects of the ontology development are provided in the chapter. The complete RDF/XML syntax file of the ontology is available at the Github site for this study - https://github.com/zainabusman/IfcOWL_pvOntology.

7.2. System Requirements and Specifications

Development of any knowledge-based system entails outlining specifications of the configuration that the system must have to perform efficiently and to serve the purpose of its development. The system requirements of any knowledge-based application explicitly identify the purpose, scope, intended users, the functionalities essential to satisfy the intended user requests, functional and non-functional requirements of the proposed software application, and any challenges that may arise. The establishment of the system requirements answers questions such as why is a PV ontology being built? What is the intended use of the software product? What is the software product supposed to look like? Who are the users? What specific requirements must it fulfil? What is the baseline for validation and verification?

Following the knowledge requirements specified by the CommonKADS methodology highlighted in chapters 5 and 6 and using the system requirements format proposed by the agile software development methodology (Pocsova et al., 2020), the features,

behaviour, and requirements to be fulfilled by ifcOWL-PvOntology are identified and outlined in table 7.1 below.

Table 7.1: System Requirements for IfcOWL-PvOntology

<p>Purpose</p> <ul style="list-style-type: none">• IfcOWL_pvOntology is proposed to contain a structured semantic representation of knowledge of our problem domain.
<p>Scope</p> <ul style="list-style-type: none">• The domain of the ontology is sustainable building technology products (SBTP) at a high level and then narrows the scope down to photovoltaic technology due to the vastness of the SBTP domains.
<p>Intended Users</p> <ul style="list-style-type: none">• Anyone looking to select and install PV technology• SBTP manufacturers who are looking to represent their products online using semantic web languages. This is so their products can be machine processable and can be factored within the prototype system to provide recommendations to users on the choice of product selection.
<p>General Requirements</p> <p>Knowledge base:</p> <ul style="list-style-type: none">• Ontology should be the central knowledge base that holds knowledge on sustainable building technology products and PV technology, using the structure of IFC standard terms.• Information should be easily stored and retrieved from the knowledge base. <p>Software product:</p>

- Ontology should be open source: ifcOWL-PvOntology should be in standard semantic web languages so other developers can exploit and re-use the knowledge model using any software tools or environment of their choice.
- Ontology development process should be agile: to achieve this, we have chosen to build the ontology using methodologies such as the flexible agile software development methodology and the unified modelling language (UML) which allows relationship of classes to be dynamically modified.
- The system should support reasoning: rules, constraints and logic will be applied.

Functional Requirements

Main Requirement: Ontology must be developed from rich domain knowledge and must align with and extend standards such as IFC.

- Extend IfcOWL ontology to develop IfcOWL-PvOntology
- Allow product manufacturers to add their PV components and specifications to the ontology as instances.
- Export ontology into the web-based application development environment and develop a link between knowledge base and application layer.

The ensuing section outlines the different software components that have been adopted in the study to achieve the system requirements elaborated above.

7.3. Software Components and Tools

Implementation of the proposed ifcOWL-PvOntology required the use of several software technologies and tools. In reference to the technologies reviewed in chapter 3, the selected semantic web technologies used in developing IfcOWL-PvOntology are summarised below.

7.3.1. Platform and Development Framework

In software engineering, developers normally write base codes on top of platforms such as Linux, macOS, Windows, Android, etc. The development environments or editors used in this study are available in several platform-compatible formats. Therefore, they

were installed based on the platform being used by the developer. The development environment used to develop IfcOWL_pvOntology is protégé-owl. Justification for selecting protege-OWL over alternative ontology editors is provided in chapter 3.

- Protégé-OWL 5.5.0 - this is the open-source ontology development editor used to develop IfcOWL-PvOntology. It comes with several plugins and tools that allow the developer to create, visualise and modify ontologies. It is flexible for swift prototyping (See chapter 3 for more on Protégé-OWL).

7.3.2. Programming Languages

The programming language used in this study is specific to the purpose of the implementation. As reviewed in chapter 3, OWL is the adopted language for the development of our ontology. It supports explicit and formal conceptualization of the PV domain. It offers well-defined syntax and semantics, adequate expressive power and sufficient reasoning support. The SPARQL Protocol and RDF Query Language is used in this study to query RDF knowledge within the ontology from the web-based application. It allows the web-based application to retrieve and manipulate the values stored in RDF format. More information on OWL and SPARQL can be found in chapter 3.

7.3.3. Plugins

Consider a plug-in as an add-on software component. They are added to existing software development editors or computer programs to provide specific extended functionalities. The plugins used in this study are itemised below.

- OntoGraf – is also a Protégé-OWL plugin that allows incremental navigation and comparison of asserted and inferred class hierarchies. It enables visualisation of the entire class hierarchies within an ontology.
- FaCT++ and Pellet – are two ontology reasoners used in this study incrementally to provide reasoning services to the developed ontology. More information on these reasoners can be found in chapter 3.

7.4. System Architecture

The software architecture developed for the proposed prototype system involves connections and communication between multiple software components and layers. This architecture contains two main components that are further divided into smaller categories. The main components are the front-end and back-end. Figure 7.2 (A) below shows the back-end of the system architecture. The back-end shows knowledge modelling achieved in chapter 6 (knowledge acquisition from literature sources, domain taxonomies, reuse of concepts from existing domain ontologies; knowledge refinement and representation). It also depicts the several components of ontology development in protégé-OWL, which is the target of this chapter. Figure 7.2 (A) also shows how the software components and tools explained in section 7.3 above have been used. Figure 7.2 (B) shows the front-end architecture developed to fit the context of this study. The front-end architecture covers the process of reading, querying, and processing the content of the ontology to perform computations and recommend PV technology products. The details of the front-end of the architecture is explained in chapter 8.

BACK END

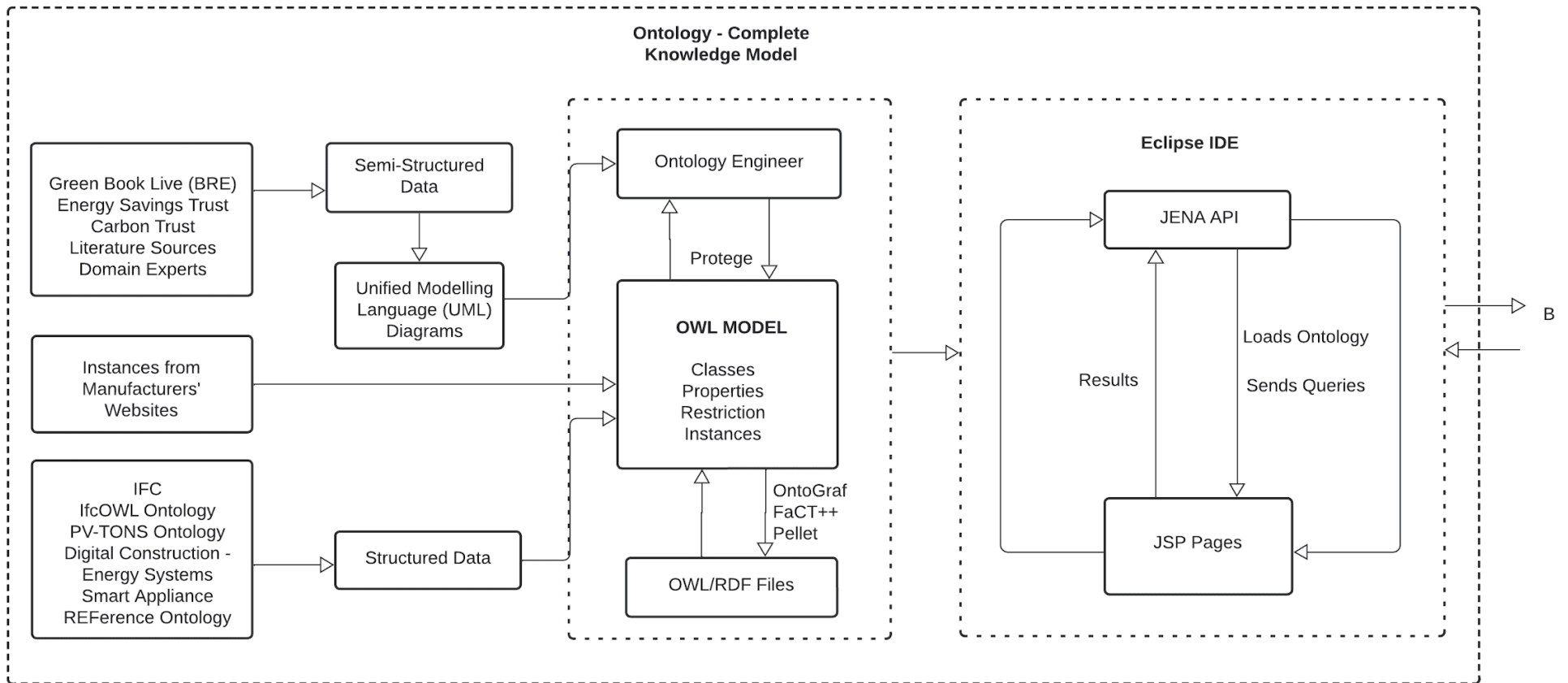


Figure 7.2 (A): Back-End of the System Architecture

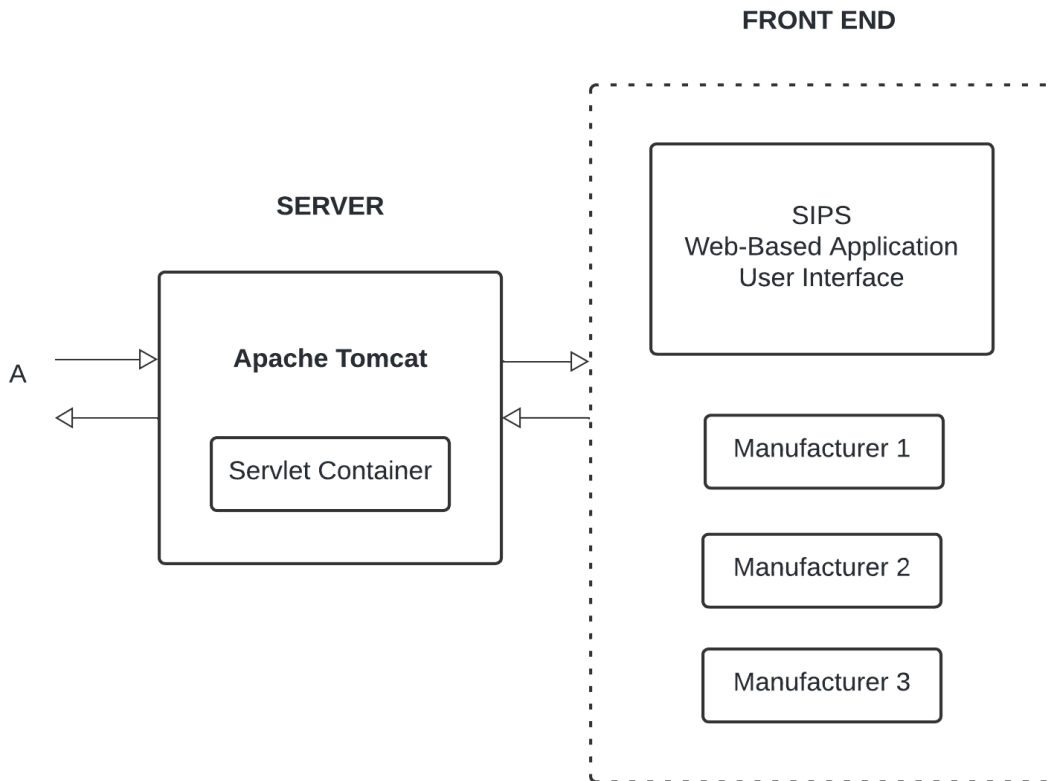


Figure 7.2 (B): Front-End of the System Architecture

7.5. Ontology Development

One of the objectives of this study is to demonstrate the extent to which semantic web technologies can provide decision support for photovoltaic technology. Based on the conceptual model developed in chapter 6, an ontology is developed in this chapter. The ontology development 101 methodology adopted for this specifies that an ontology could either be developed from scratch or other ontologies could be re-used and extended. As shown in chapter 6, the conceptual model proposed for this implementation contains re-usable concepts from PV-Tons (Abanda, Tah, and Duce, 2013) and IfcOWL ontology (Pauwels and Terkaj, 2016). We have also established that the main contribution of IFC ontology to this study is its standard taxonomy and unique standardised representation of building's data (i.e., its implementation structure within the ontology). For this reason, the ontology proposed for this study will be implemented on top of the standardised ifcOWL ontology. The main challenge faced in this implementation is to ensure ontology developed is simple, modular, and specific to

photovoltaic technology, that can be extended towards other sustainable building technology products in further studies.

The ensuing sections describe integration of IFC and ontology and its conversions. The steps involved in extending the ifcOWL ontology, model view definitions and subsequently developing the proposed PV technology ontology based on the conceptual model in chapter 6 is well elaborated.

7.5.1. Proposed Approach to Integrating Semantic Web and IFC

Chapter 4 explained the uniqueness of IfcOWL ontology, the numerous benefits of re-using and extending it and how it fits into the context of this research, there would be no need to repeat it here.

7.5.1.1. Model View Definition (MVD) of IFC

Recall that IFC is a huge collection of definitions, relations, and properties of a wide range of entities within the architecture construction and engineering domain. The idea behind MVD is to model a narrowed down scope of these data represented in IFC. This means the complete IFC schema is segmented into several subsets containing a group of entities (MVDs) to be used for specific purposes or workflows. For instance, one MVD may focus on a single design domain such as buildings, while another on bridges. MVD makes it simpler for application developers to use IFC, because instead of exploiting the whole IFC schema in their implementation, the developer can simply select an MVD depending on the requirements of the application to be developed. As explained in greater detail in chapter 4, the buildingSMART working group uses an Information Delivery Manual (IDM) to define which requirements are mapped in specific MVDs. This study will not accentuate the details behind the formation of MVDs, however, all defined MVDs are available in a library published by buildingSMART, (2021b).

The MVDs that best suit this study are Energy Analysis View (EV) and Product Library View (LV 0.1). EV includes a group of entities on energy usage and cost while LV focusses on construction products, their manufacturers and information regarding products and how they are configured (buildingSMART, 2022b). Fortunately, these

MVDs are available within one IFC schema; IFC4 ADD2 TC1. The schema covers entirely, the *IfcObjectDefinition*, *IfcPropertyDefinition* and *IfcRelationship* of all IFC entities that define energy-related construction and building products, including the entity of interest for this study; *IfcSolarDevice*. All software applications that require BIM data to be exported from IFC schema use one form of MVD or another. It is likely that software developers in the future will develop their own MVDs for unique software implementations. However, the MVDs must be interoperable. Further details on MVDs can be found in (buildingSMART, 2022c).

7.5.1.2. Conversion of IFC Specifications into Semantic Web

IFC data can be converted into semantic web in three ways. The first approach is generating an ontology in OWL language from IFC EXPRESS specification. Second method is through generating RDF graphs from IFC-SPF files. The third approach is through existing “IFC to ontology” converters.

Ontology generation through IFC EXPRESS is the most researched approach of converting IFC into semantic web. In chapter 4, we provided a review of attempted approaches available (in literature) to convert IFC’s EXPRESS specifications into OWL. The review gave descriptions of conversion approaches from IFC to OWL and outlined the challenges faced by researchers and how they have been overcome by Pauwels, Terkaj, and the other buildingSMART working group (Pauwels and Terkaj, 2016) to develop the standard IfcOWL ontologies. The code for generating an OWL ontology from the IFC EXPRESS file consist of reusable java codes that ensure ontology generated stay within the context of IFC. These java components are made available by Pauwels (2021b).

The second approach looks into generating RDF graphs from IFC-SPF files. Because BIM models contain large details of most aspects of a building, IFC generates its representation of these BIM details in plain text to make it modular and suitable for BIM data exchange. This variant of IFC is termed IFC-SPF. The SPF format is a subtype of STEP_file. It is the most widely used IFC format because of its compact size, making it a vital option for generating RDF graphs from IFC. The standardised codes and steps

involved in generating RDF graphs from IFC-SPF files are available here (Pauwels, 2021a). To run this code, the initial step is to select an IFC schema depending on the preferred MVD. Next, to run the code on a computer, one needs to load executable JARS that include necessary support for the code to be run. Lastly, the code is run through the command line interface to generate an RDF graph (Pauwels, 2021a).

The third approach considered is through 'IFC to ontology' converters that are being developed by the buildingSMART working group. These converters are meant to transform IFC data into smaller modular ontologies that can be used by software developers for unique purposes.

If IfcOWL ontology wasn't available or standardised, this research would have used one of the three approaches explained above to develop its IFC ontology. This study chooses to reuse a standardised IfcOWL ontology rather than creating its own version. The reused IfcOWL ontology is then extended to develop an enhanced knowledge model of the problem domain, sustainable building technology products and photovoltaic systems. Developing the ontology from scratch is not recommended. It is tedious, and work and time intensive. It can also result in errors and inconsistencies due to the manual nature. Since IfcOWL ontology is standardised, its re-use guarantees correctness. It is scalable. The ifcOWL ontology could be easily parsed, and hierarchies, rules and restrictions of all classes can be retrieved from the ontology Tbox. Once these are retrieved, we can correctly generate new classes and establish relations between them. Furthermore, simple, and sophisticated applications can be built on top of it through parsing. Current IFC specifications do not fully capture all entities of the PV technology domain, so even if this study was converted from scratch, it would still need to make a lot of manual additions to the resulting ontology. The decision to extend the standard taxonomy is logical because only continuous use can make standards advance. The proposed application would be compatible, unrestricted, and suitable for re-use for future studies.

7.5.2. Extending IfcOWL Ontology

Note that there are up to 7 versions of standard ifcOWL ontology that have been published and available for re-use. These versions are available at the developers GitHub site (Pauwels, 2021a). One of these versions named ifcOWL IFC4/ADD2_TC1/OWL, was developed based on the MVD database IFC4 ADD2 TC1 that includes energy analysis view and product library views, which are required in the context of this study. The complete IFC EXPRESS schema including all entities, and properties are contained within the ifcOWL ontology. However, the ontology contains two sub ontologies that define certain OWL classes and properties that are not contained in IFC. These are EXPRESS and LIST ontology.

The data types defined within EXPRESS ontology are specific to EXPRESS schema alone. EXPRESS ontology was developed through a 'genOnto' converter by Terkaj and revised by Pauwels (Terkaj & Pauwels, 2015a). It is serialised using C++ RDF library developed by the same authors. More details on EXPRESS can be found at (Terkaj & Pauwels, 2015a) and (buildingSMART, 2022a). LIST ontology (Terkaj & Pauwels, 2015b) contains a representation of ordered lists that facilitate representations of all lists within both EXPRESS and ifcOWL ontologies.

This begins by directly importing the latest official ifcOWL ontology into protégé (the ontology development and editing software selected for the prototype development). To reuse ifcOWL, the first step is to do a direct import of the ifcOWL IFC4/ADD2_TC1/OWL ontology. ifcOWL ontology indirectly imports only EXPRESS ontology. EXPRESS ontology then imports LIST ontology.

Ontology imports	Ontology Prefixes	General class axioms
Imported ontologies:		
<p><https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL> OWL Ontology IRI: <https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL> Location: https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL</p>		
Indirect Imports		
<p><https://w3id.org/express> express Ontology IRI: <https://w3id.org/express> Location: https://w3id.org/express</p>		
<p><https://w3id.org/list> list Ontology IRI: <https://w3id.org/list> Location: https://w3id.org/list</p>		

Figure 7.3: Imported ontologies in Protégé-OWL 5.5.0

Once the ifcOWL ontology is active, we can trace our class of interest which is *IfcSolarDevice*. Note the parent classes starting from *IfcRoot* within the class hierarchy of the *IfcSolarDevice* in Figure 7.4 below and notice the ifcOWL ontology has no sub-classes for *IfcSolarDevice*.

Firstly, it is imperative to note that there are new concepts within the updated version of the model view definition (MVD) database (IFC4 ADD2 TC1) used in this study that are not available within the standard ifcOWL ontology. Therefore, IFC schema was thoroughly traversed to identify the missing concepts in the ontology. These concepts are added into the ontology following its exact structure and hierarchy as specified in the schema.

The next step is to transform the conceptual model developed in chapter 6 in corresponding OWL classes and properties and built on top of the already imported ifcOWL ontology.

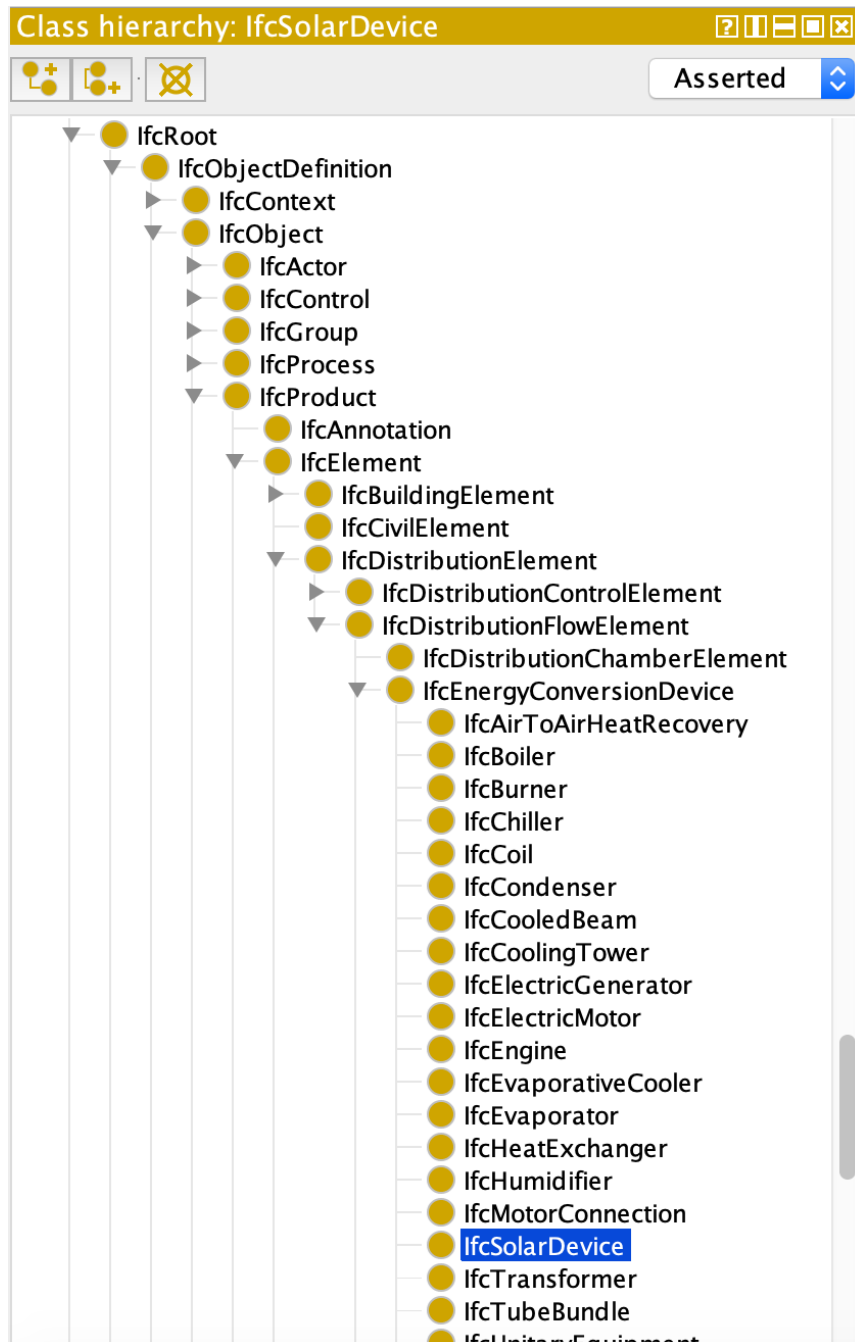


Figure 7.4: Class Hierarchy for IfcSolarDevice

7.5.3. Transformation of the Conceptual Model (in UML) into Corresponding OWL

This section discusses how to convert the object-oriented UML conceptual model in Figure 6.7 into a corresponding knowledge-model within protégé-OWL. Several UML-OWL conversion tools are available and reviewed in literature. Thorough review (Vo and Hoang, 2020) has shown that these tools result in loss of data, which requires the developer to manually correct inaccuracies, add missing data and improve the

resulting knowledge in OWL. The details of the studies on UML-OWL conversion tools and their reviews are not within the scope of this study. However, the most prominent recommendations from these studies are the use of conventional rules to transform UML to OWL successfully. Table 7.2 below shows an outline of the rules used in the study. Details of how each rule is applied is discussed in the ensuing sections. A detailed review of UML – OWL conversion rules are available at Vo and Hoang (2020). It is important to note that the resulting knowledge model in OWL is built on top of the ifcOWL ontology already imported into protégé-OWL.

Table 7.2: Rules for Transforming UML into OWL

	UML	OWL
Rule 1: Each UML class is transformed into a class in OWL.	Class	Class
Rule 2: Attributes in UML should be transformed into primitive data types in OWL.	Attribute	Data-type Property
Rule 3: Association of UML classes should be transformed into object properties.	Association	Object Property
Rule 4: Conventional attributes should be defined as enumeration data types	Conventional Attributes	Enumeration
Rule 5: Generalisation association (“is a) between classes should be transformed to subclasses of the parent class	Generalisation between classes	subClassOf
Rule 6: Inheritance between associations should be transformed into subProperties of the parent properties.	Generalisation between association	subPropertyOf
Rule 7: Multiplicities set for attributes, operations and associations in UML should be transformed into cardinality restrictions in OWL.	Multiplicities	Cardinalities
Rule 8:	Multiplicity Constraints	Functional Property

7.5.3.1. Classes

The first step involves rules 1 and 5 from the Table 7.2 above to transform classes and subclasses. In both UML and OWL, the class description has similar structures and relationships where its attributes define the structure, and instances define behaviour of the class. Similarly, generalisation and specialisation in UML and OWL are comparable. Each class in UML is transformed into an OWL class. Each subclass in UML is transformed into corresponding subclass in OWL to establish the class hierarchy. Note that these classes are built upon the ifcOWL ontology. All entities added will have a suffix “_ext” to show that it is an extension of the ifcOWL ontology.

One of the main challenges faced is determining the proper position within ifcOWL ontology to add classes. While the position of *IfcSolarDevice* and its subsequent subclass are obvious, it is challenging and necessary to know the proper position to place other classes to establish the appropriate class hierarchy. For instance, the *SolarDeviceInstallation* class is derived from the *IfcTask* class. *IfcTask* defines all tasks to be carried out within a construction project. This could include one of the activities within construction, an installation of a product, or any operation related activities. *IfcTask* provides descriptions, process, duration, status, and priority to the defined task. Therefore, it contains the attributes required to build the *SolarDeviceInstallation* class upon. Another instance is the *organisation* class within the UML conceptual model. A major challenge that was faced when deciding the appropriate hierarchy for an *organisation* was either to place it under *IfcActor* or *IfcActorSelect*. *IfcActor* defines all human agents involved within a construction project and allows descriptions for the person, organisation, roles, and addresses (BuildingSMART, 2022i). *IfcActorSelect* allows a *person* and/or *organisation* to be referenced. Therefore, for the context of this study, *IfcActor* is more appropriate for *organisation* and *IfcActorSelect* is used for reference. The EXPRESS specification for *IfcActor* is:

ENTITY IfcActor

SUBTYPE OF (IfcObject);

TheActor : IfcActorSelect;

INVERSE

```

    IsActingUpon : SET OF IfcRelAssignsToActor FOR RelatingActor;
END_ENTITY;

```

```

TYPE IfcActorSelect =

```

```

    SELECT ( IfcOrganization, IfcPerson, IfcPersonAndOrganization ) ;
END_TYPE;

```

where *IfcPerson* (is the person related to the organisation), *IfcOrganization* (to which the person is related) and *IfcPersonAndOrganization* includes all persons and role(s) (played by the person) within the context of an organisation. Therefore, *organisation* in UML has been transformed and built on the *IfcPersonAndOrganization* under *IfcActor* in ifcOWL ontology. Note that the term *organisation* has been substituted to IFC's standard term *IfcPersonAndOrganization*. Figures 7.5, 7.6 and 7.7 show the resulting class hierarchies of *IfcSolarDevice*, *SolarDeviceInstallation_Ext* built under *IfcTask* and *IfcPersonAndOrganization* under *IfcActor* respectively.

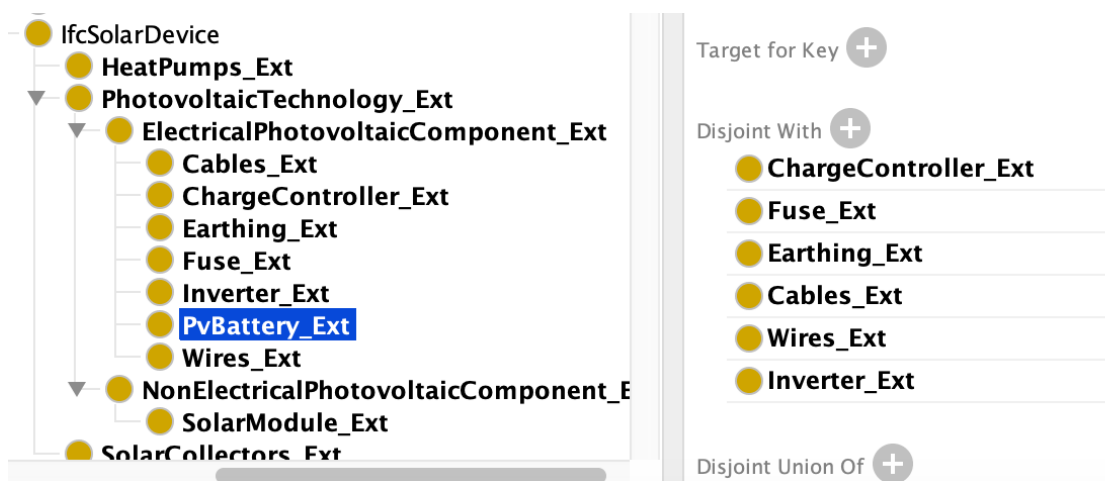


Figure 7.5: Class Hierarchy of IfcSolarDevice Within Protégé

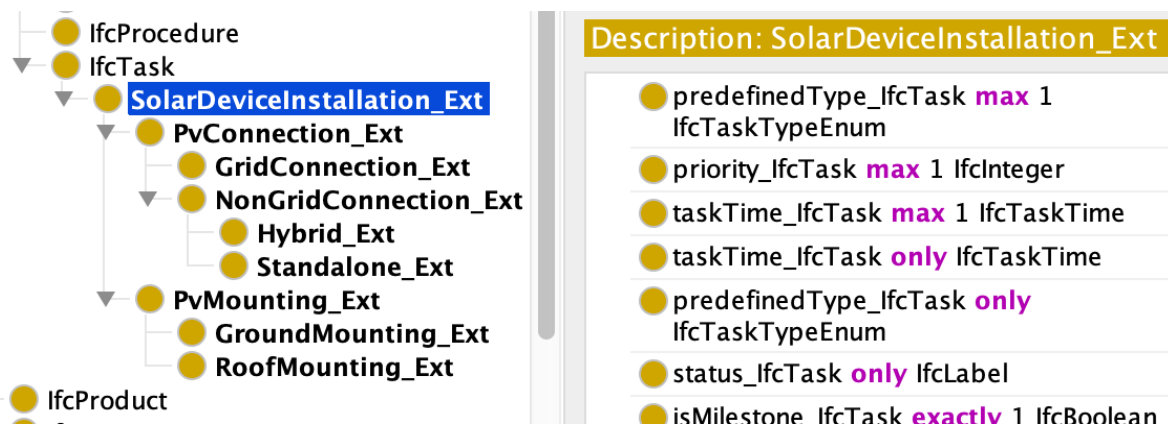


Figure 7.6: Class Hierarchy of SolarDeviceInstallation under IfcTask

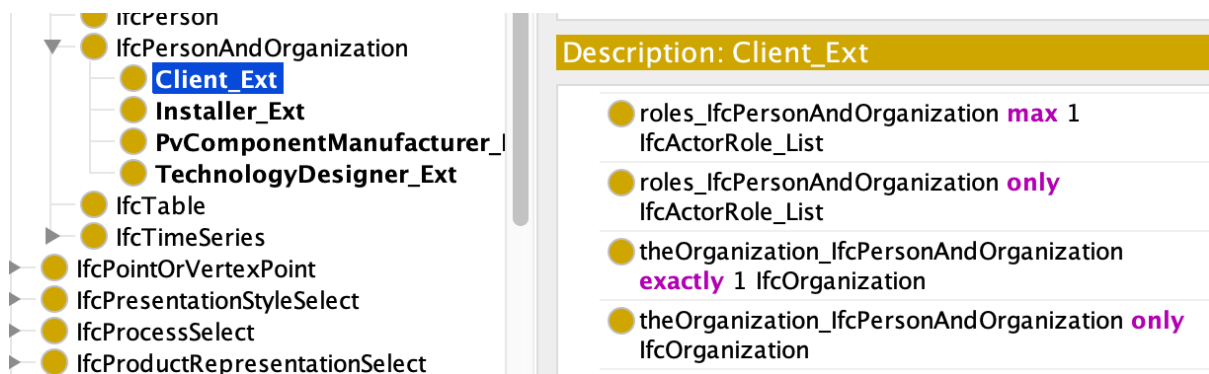


Figure 7.7: Class Hierarchy of Modified IfcPersonAndOrganization

Disjoint classes are defined in protégé to show classes that cannot share an instance. For example, the *electricalPhotovoltaicComponents*'s subclasses in figure 7.5 are disjoint with each other, while the subclasses under *IfcPersonAndOrganization* in figure 7.7 are not disjoint. It is likely that an instance of class *Installer_Ext* may be an instance of *PvComponentManufacturer_Ext* as well. Protégé also allows a class to have multiple parent classes. Note in figure 7.8 below, that *PvBattery_Ext* is a *ElectricalPhotovoltaicComponent_Ext* and a subclass of *IfcElectricFlowStorageDevice* which defines all storage devices that store and progressively release electric energy.

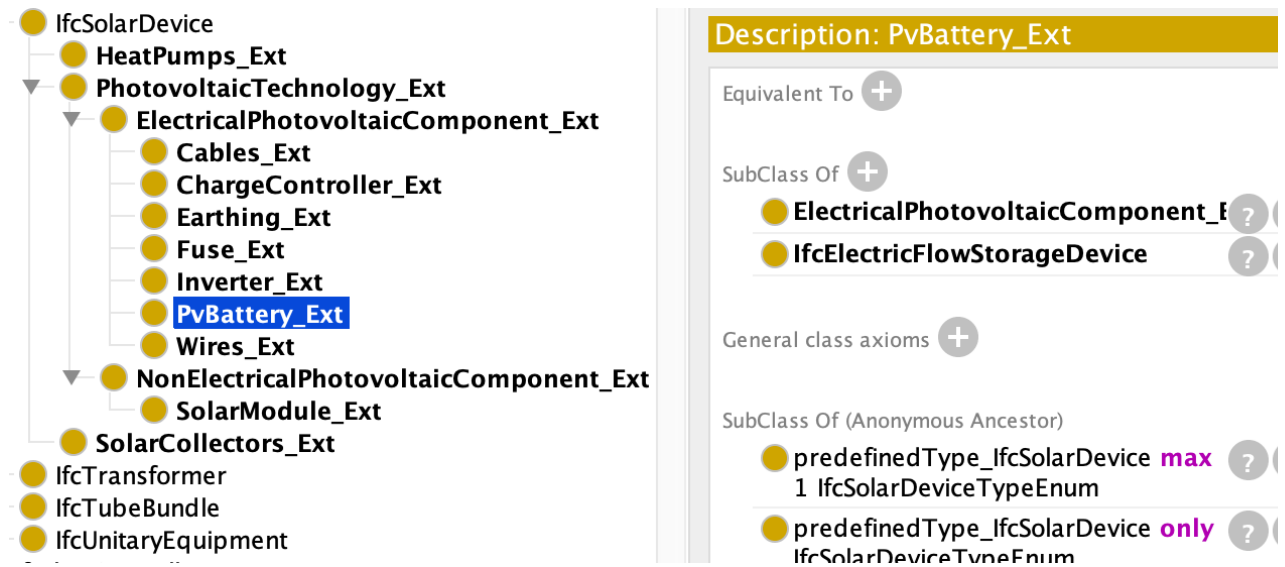


Figure 7.8: Super Classes of IfcBattery_Ext

In the previous chapter, CommonKADS methodology was used to establish properties of defined classes. However, the descriptions of these properties were not defined. In the ensuing sections, properties are transformed from UML to OWL and specific definitions are provided in protégé.

7.5.3.2. Properties of Classes

In ontology modelling, properties are essential because they describe explicit relationships between concepts and instances. These are object, data, and annotation properties. From table 7.2 above, rules 2 and 3 are applied to transform attributes and association of UML classes into data and object properties in OWL in protégé respectively.

Attributes of every single class within the UML model are transformed into data-type properties. For instance, some of the main properties of the *PhotovoltaicTechnology* and its subclasses such as *hasRatedPower*, *hasManufacturer*, *hasCost*, *hasModuleSurfaceArea* properties can be seen as implemented in protégé in figure 7.9 below. A clear distinction seen in extending IfcOWL ontology is that its structure allows enumeration of terms. These terms could be measured values, or an assured list used as alternatives that can be selected when properties are defined. Therefore, we adopted this pattern of enumeration. For instance, the data property *hasModuleMaterialType* has an

enumerated list that contains the different types of solar module materials considered in this study. When an instance of the class *IfcSolarModule* is likely to be added, the material type will have to be from amongst the types defined in the enumerated list. A similar approach is used to define other data types such as *hasBatteryType*, *hasInverterType* and *hasChargeControllerType*. Figure 7.10 shows the enumerated list termed *ModuleMaterialTypeEnum*.

Association of UML classes are transformed into object properties in protégé. Common practice defines ontology properties to seem like regular complete sentences. For instance, the relationship between an *IfcSolarDevice* and an *organisation* is such that an *organisation* *IsInvolvedIn* *IfcSolarDevice*. Notice that *IsInvolvedIn* has several sub properties such as *isInstalledBy*, *isManufacturedBy* because it relates to the *IfcSolarDevice* through several ways. The property *isManufacturedBy* defines the manufacturer or organisation that manufactures solar devices. *isInstalledBy* defines the individuals or companies that handle installations. *IsRequiredBy* defines the client that needs to install the solar device in his home. These transformations are provided below in figure 7.11 within protégé.

Note that properties have several characteristics they can be defined by. They can be functional, inverse, transitive, symmetric or other characteristics. A property is functional if for any given individual, the property will have at least one value. That means there is a definite relationship through this property to that individual. In protégé-OWL, the list of property characteristics is visible within the editor and selected as desired. For example, the object property *isManufacturedBy* is functional with *PhotovoltaicTechnology_Ext* and range *PvComponentManufacturer_Ext*. Therefore, an instance of *PhotovoltaicTechnology_Ext* may also be an instance of *PvComponentManufacturer*. To explain further, an instance of *PhotovoltaicTechnology_Ext* *A* *isManufacturedby* a *PvComponentManufacturer_Ext*. An instance of *A* cannot be manufactured by two different *PvComponentManufacturer_Ext*.

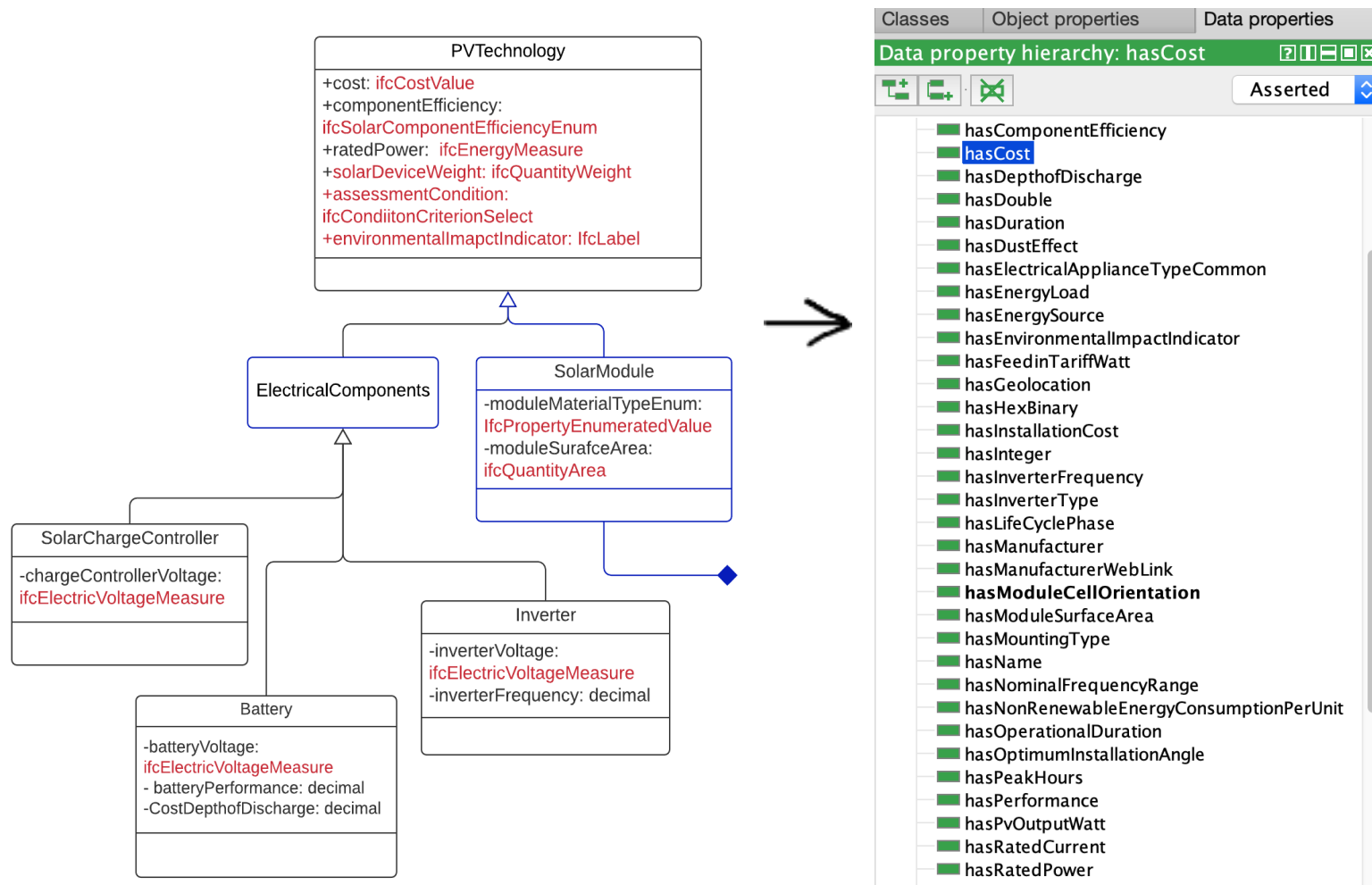


Figure 7.9: Transformation of UML attributes into data-type properties in Protégé-OWL

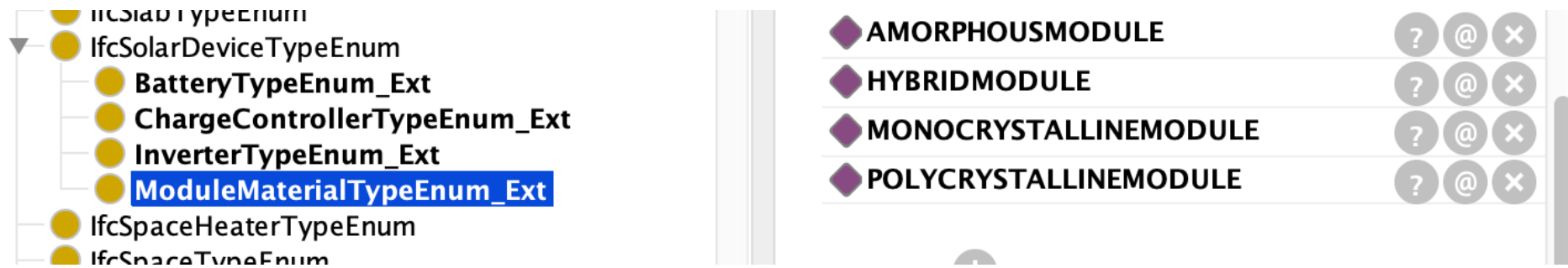


Figure 7.10: An enumerated list of solar module material types (*ModuleMaterialTypeEnum*) in Protégé-OWL editor

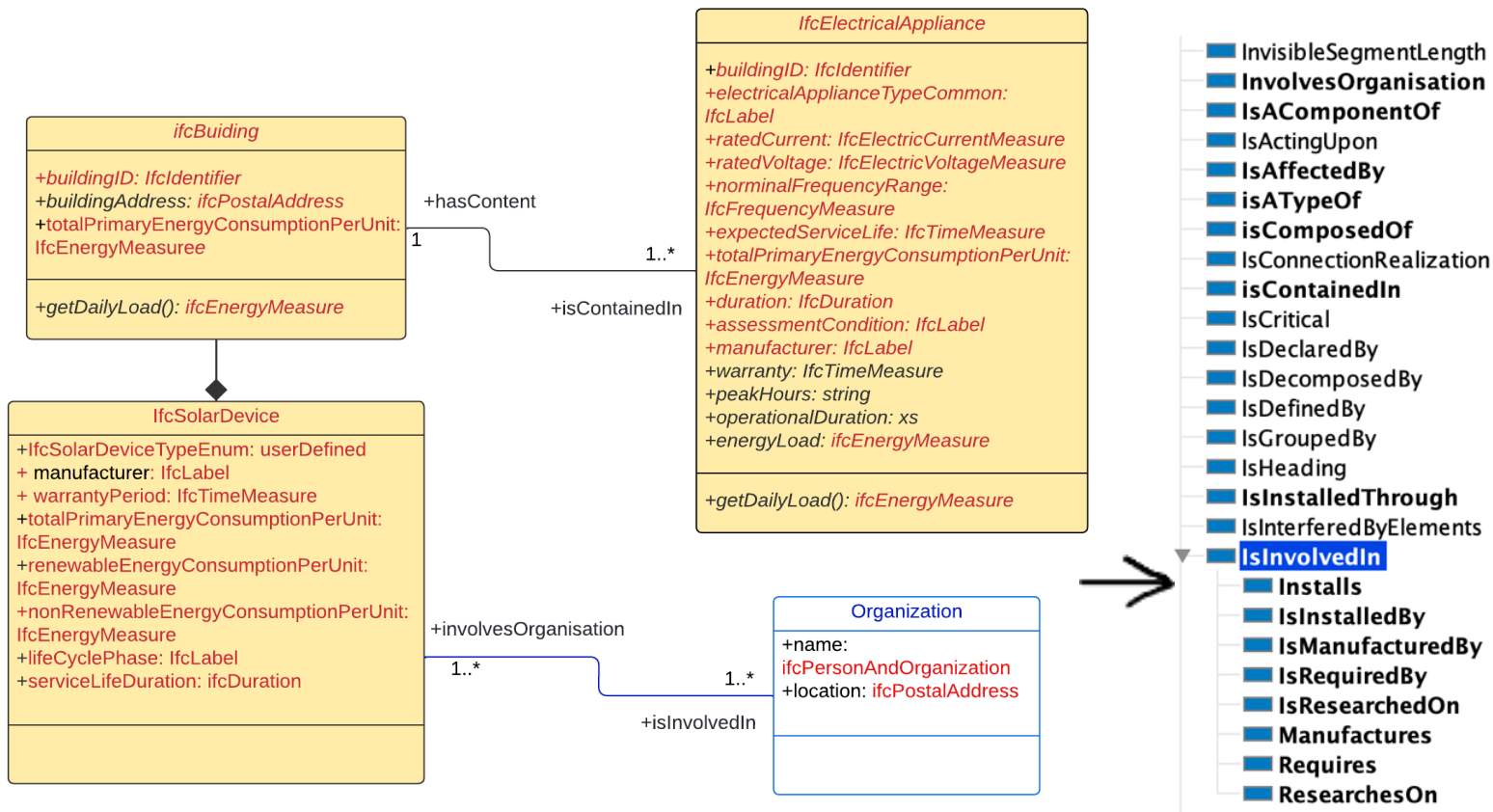


Figure 7.11: Transformation of Association of UML classes into object properties in Protégé-OWL

- isCritical_IfcTaskTime
- isDeclaredBy_IfcObject
- isDecomposedBy_IfcObjectDefinition
- isDefinedBy_IfcPropertySetDefinition
- isGroupedBy_IfcGroup
- isHeading_IfcTableRow
- IsInstalledThrough**
- isInterferedByElements_IfcElement
- IsInvolvedIn**
 - Installs**
 - IsInstalledBy**
 - IsManufacturedBy**
 - IsRequiredBy**
 - IsResearchedBy**
 - Manufactures**
 - Requires**
 - ResearchesOn**
- isLinear_IfcStructuralResultGroup
- isMilestone_IfcTask

Characteris ? | | | | |
Description: IsManufacturedBy

- Functional**
- Inverse functional
- Transitive
- Symmetric
- Asymmetric
- Reflexive
- Irreflexive

Equivalent To +

SubProperty Of +

IsInvolvedIn

Inverse Of +

Manufactures

Domains (intersection) +

PhotovoltaicTechnology_Ext

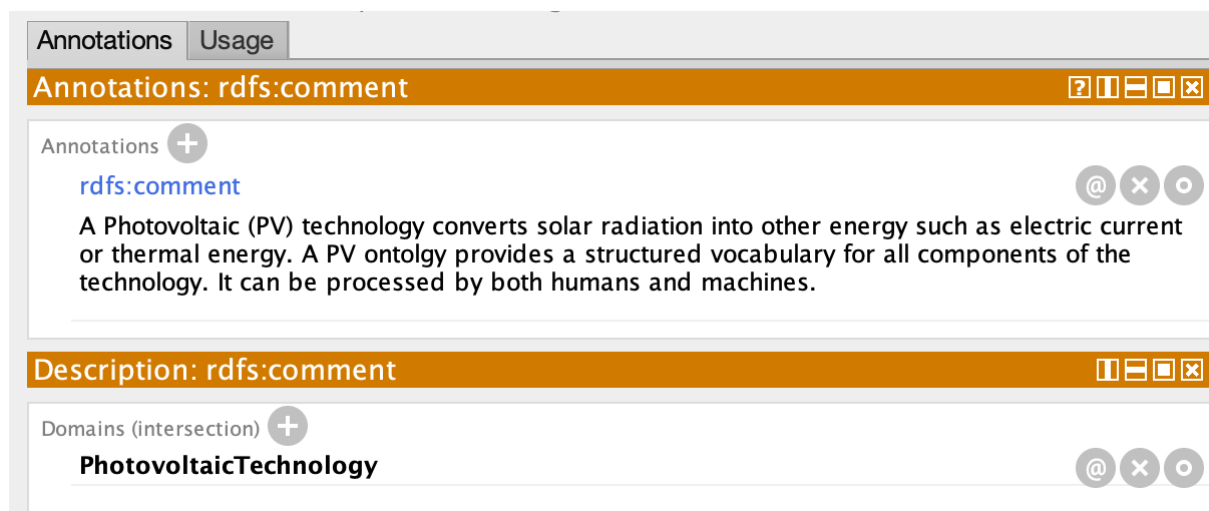
Ranges (intersection) +

PvComponentManufacturer_Ext

Figure 7.12: Characteristics of Properties defined in Protégé-OWL

Figure 7.12 above depicts functional property defined in protégé. Observe that properties have a bidirectional relationship with individuals. From figure 7.12 above, if an individual A of *PhotovoltaicTechnology_Ext* is linked to individual B of *PvComponentManufacturer_Ext* through a functional property *isManufacturedBy*, then B is also linked to A through an inverse property *Manufactures*. It is important to note that an inverse property of a selected functional property is also a functional property. A property is transitive if the particular property creates a connection between individual A to B, and connects B to C, then it is inferred that A is related to C. A symmetric property is an inverse of itself. If Individual A is related to B, then automatically, B must also be related to A along the same property. Asymmetric properties on the other hand have no bidirectional relationship for individual A and B defined along the same property. Defining a property to be reflexive means every individual declared is related to itself via that property while irreflexive means individuals are not related to themselves via that property.

Since annotation properties are defined to provide additional explanation, this study defines annotation properties to classes, individuals, or properties where further clarification is required. In ifcOWL ontology, annotation properties include *rdfs:comment*, *rdfs:label*, *cc:licence*, and other types In protégé-OWL some of our annotation properties were defined as *rdf:comment*. For instance, figure 7.12 shows annotation and usage for class *PhotovoltaicTechnology*.



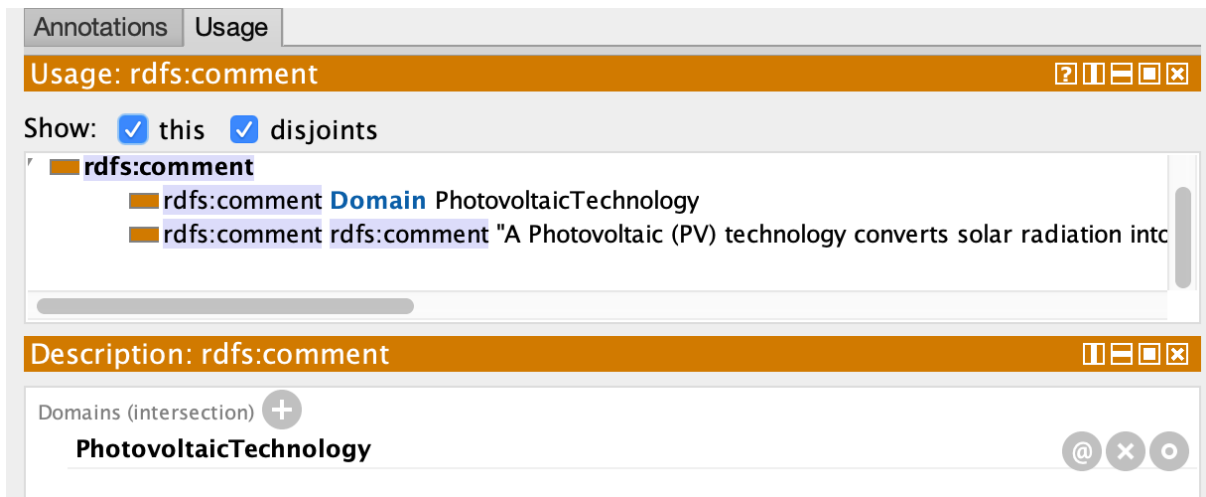


Figure 7.12: Definition of Annotation properties within Protégé-OWL

7.5.3.3. Facets of Properties

This study used facets to place restrictions on properties defined in section 7.5.3.2 above. These could be value types, allowed values or cardinality. Following the structure of the ifcOWL ontology being extended here, we set certain values to follow the IFC taxonomy. For instance, *hasRatedPower* has the value *IfcEnergyMeasure*. Furthermore, recall the data property *hasModuleMaterialType*, represents the different types of materials used in making solar modules. These have been reviewed in chapter 2 and listed as monocrystalline, polycrystalline, amorphous and hybrid solar material types. These materials are well researched, and they are highly important when making informed decisions about PV technologies. Therefore, we defined them in protégé as allowed values. Other data properties *such as hasInverterType, hasBatteryType, hasChargeControllerType* also have allowed values defined. Allowed values are defined in this study as enumeration lists as shown in figure 7.11 above. They are used together with cardinality restrictions.

Cardinality restrictions set the quantity of a value. It could be an exact cardinality denoting exactly one amount (e.g., exactly 1), or maximum and minimum qualified restrictions. Other restrictions often used are existential restriction (some) or universal restrictions (only) as seen in figure 7.13A below. Figure 7.13B shows cardinality

restrictions specified for *hasModuleMaterialType*. Note that restrictions are set across them but can only be visible when the class or property is selected.

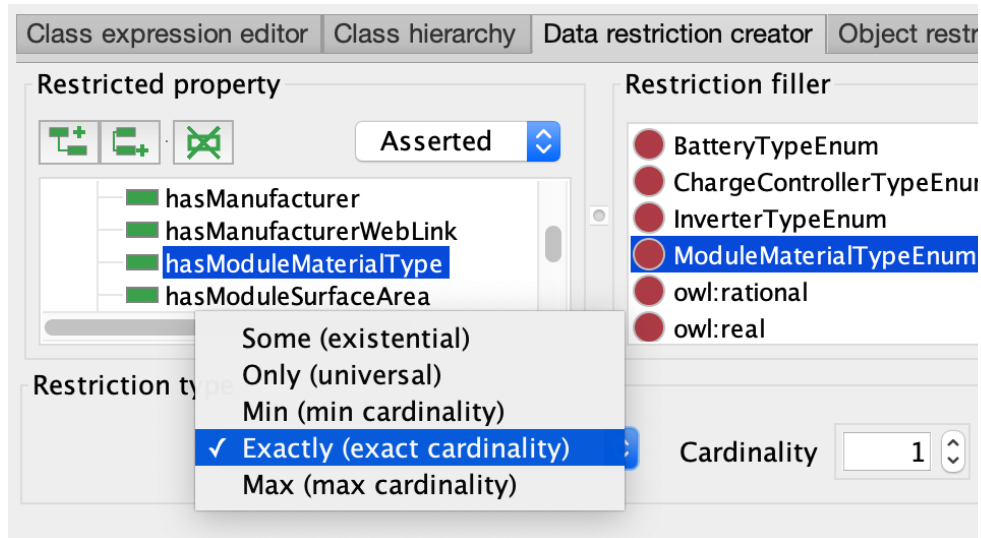


Figure 7.13 A: Defining cardinality restrictions

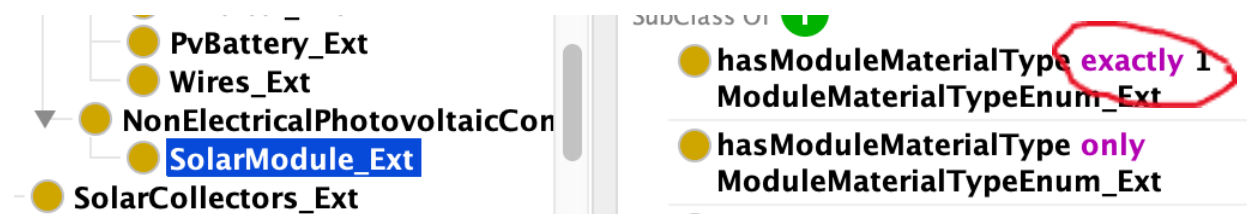


Figure 7.13B: Cardinality Restrictions for SolarModuleMaterialType.

Based on these, the most important stages on the development of ifcOWL-PvOntology have been accomplished. The ontology consists of OWL classes transformed from our established UML class diagram in chapter 6. Data, object, and annotations properties have been transformed correspondingly in accordance with the conventional transformation rules we provided. We confirm that the classes structure in the conceptual model (UML class diagram) have been correctly mapped to OWL equivalents in protégé-OWL. Validation of the resulting knowledge model is done using reasoners within the protégé ontology editor and is presented in the evaluation chapter.

7.6. Storage of Ontology

There are several methods of storing ontologies. These methods are categorised under two main models: the native storage approach and the database management storage approach. The first approach, ontology is stored using a straightforward flat file system. The developer has options to store the ontology either in RDF triple store files or hierarchical stores. This form of ontology storage is direct. It is highly suitable for applications that involve a sole developer. The principal drawback of this approach is that it does not support shareability and scalability (Abburu and Golla, 2016). It also does not allow a remote interface whereby a client of an application cannot have access to or alter the knowledge within ontology through a client server architecture. The second approach uses relational and object relational database repositories to store ontology. It is an approach that involves several steps because databases do not support hierarchical representation of knowledge directly. However, this approach enhances robustness and performance of applications. It also supports scalability and shareability thereby making it suitable for applications involving several developers. The choice of ontology storage method therefore depends on how the ontology has been developed and on the intended form by which the ontology will be utilised.

Nonetheless, Jena API, the API selected for the second part of the prototype implementation (with further details in chapter 8), manages ontologies stored in a simple RDF file system. It supports querying of RDF data through SPARQL. The Jena API supports updates, and it satisfies major application yardsticks such as consistency, and reliability. It reads in the stored ontology using a series of command line scripts. These are further elaborated and provided in chapter 8. Furthermore, since this prototype is developed by a single developer, the native file approach is suitable and thereby adopted. An ontology can be stored in several file formats. This includes RDF/XML syntax, turtle syntax, OWL/XML syntax, OWL functional syntax, Manchester OWL syntax, OBO format, LaTeX syntax, and JSON-LD. Describing the details behind each format is beyond the context of this study. In the context of this study, the ontology is stored in RDF/XML and in OWL. The reasons are provided below.

The ontology is stored in OWL because it re-uses and extends the standardised ontology for IFC, the ifcOWL ontology. This re-used ontology is available in OWL and imported into the ontology editing software in OWL format. Therefore, the IfcOWL_pvOntology will be stored in an OWL format primarily to allow further extensions of the IFC. Secondly, the IfcOWL_pvOntology is a modular ontology for the PV domain, it could be adopted and approved by the buildingSMART working group in the W3C as a standard ontology for photovoltaic system, inspired by the IFC standard. Therefore, it is important for the ontology to be made available in OWL.

Similarly, the IfcOWL_pvOntology is also stored in RDF/XML syntax to fulfil vital requirements and/or features essential for this research to proceed with further implementation. The requirements are presented as follows:

- Semantic information retrieval: the ontology is stored in a way that it can be retrieved, its content can be read and processed.
- Query language support: RDF queries can be run, and the semantics of the ontology can be mapped with non-semantic data.
- API support: knowledge stored within the ontology can be run through the application's user interface.
- Access control: ontology can only be accessed and altered by the developer, not the clients of the application.
- Inference support: new facts and desired deductions can be obtained from the ontology.
- Performance: a high-performance application is expected at the end of this implementation.

A fragment of the IfcOWL_pvOntology in RDF/XML syntax is shown in figure 7.14 below.

```

    http://www.semanticweb.org/zainabusman/ontologies/2021/5/IfcOWL-pvOntologyFinal#Standalone_Ext
-->
<owl:Class rdf:about="http://www.semanticweb.org/zainabusman/ontologies/2021/5/IfcOWL-pvOntologyFinal#Standalone_Ext">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/zainabusman/ontologies/2021/5/IfcOWL-
pvOntologyFinal#NonGridConnection_Ext"/>
</owl:Class>
<!--
    http://www.semanticweb.org/zainabusman/ontologies/2021/5/IfcOWL-pvOntologyFinal#TechnologyDesigner_Ext
-->
<owl:Class rdf:about="http://www.semanticweb.org/zainabusman/ontologies/2021/5/IfcOWL-pvOntologyFinal#TechnologyDesigner_Ext">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/zainabusman/ontologies/2021/5/IfcOWL-pvOntologyFinal#Research_Ext"/>
</owl:Class>
<!--
    http://www.semanticweb.org/zainabusman/ontologies/2021/5/IfcOWL-pvOntologyFinal#Wires_Ext
-->
<owl:Class rdf:about="http://www.semanticweb.org/zainabusman/ontologies/2021/5/IfcOWL-pvOntologyFinal#Wires_Ext">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/zainabusman/ontologies/2021/5/IfcOWL-
pvOntologyFinal#ElectricalPhotovoltaicComponent_Ext"/>
</owl:Class>
<!--
    https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL#IfcElectricFlowStorageDevice
-->
<owl:Class rdf:about="https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL#IfcElectricFlowStorageDevice"/>
<!--
    https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL#IfcPersonAndOrganization
-->
<owl:Class rdf:about="https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL#IfcPersonAndOrganization"/>
<!--
    https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL#IfcSolarDevice
-->
<owl:Class rdf:about="https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL#IfcSolarDevice"/>
<!--
    https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL#IfcSolarDeviceTypeEnum
-->
<owl:Class rdf:about="https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL#IfcSolarDeviceTypeEnum"/>
<!--

```

Figure 7.14: Fragment of IfcOWL_pvOntology file.

This ontology is available on the web for PV component manufacturers to populate the ontology with their products. The complete ontology and its codes is available at the Github site for this study - https://github.com/zainabusman/IfcOWL_pvOntology. The ensuing sections show addition of instances into protégé.

7.7. Addition of Instances by Product Manufacturers

Currently, manufacturers present their products on their websites for buyers to view, add to cart and purchase. These products and their details do not serve any additional purposes because they are represented in HTML. One of the targets of this research is to have PV technology products represented in semantic language compatible formats so they can be well structured, read and processed by the machines and semantic reasoning can be run on them. Similarly, it is the target of this research to encourage manufacturers to represent product information using standard terms similar from one manufacturer to another. For that reason, the IfcOWL_pvOntology is made available

online at github so different PV manufacturers can add their instances. Once they are added into the ontology, further implementations can be conducted to provide decision support and recommendations to the intended user based on the components and products presented by the manufacturer. At the end of this prototype implementation, the manufacturer is presented with a new web page where their products are represented in RDF.

The process of adding instances to an ontology is straightforward. It involves three simple steps. In the ontology editing software protégé.

- Select a class
- Add instances of that individual class
- Populate the instance with appropriate object and data property values.

A prototype system needs to be only a preliminary version; therefore, instances are added by only three manufacturers. These instances are manually abstracted and added into protégé. This is because most of the product information is represented using different formats and terms. Screen scraping and other automatic tools used to copy information on digital display present data only as raw text. Such text needs to be re-processed to be added into the ontology. This approach is time consuming but will be considered in future versions of the software prototype. The abstracted instances are edited into protégé. Three manufacturers have added instances into the ifcOWL-pvOntology. The resulting ontologies are IfcOWL_pvOntologyMaxPv.rdf, IfcOWL_pvOntologySimbaSolutions.rdf, and IfcOWL_pvOntologySolarShop.rdf. Figure 7.15 below shows instances added by manufacturer *MaxPV*. As observed, this manufacturer saves the ontology that contains its instances as IfcOWL_pvOntologyMaxPv.

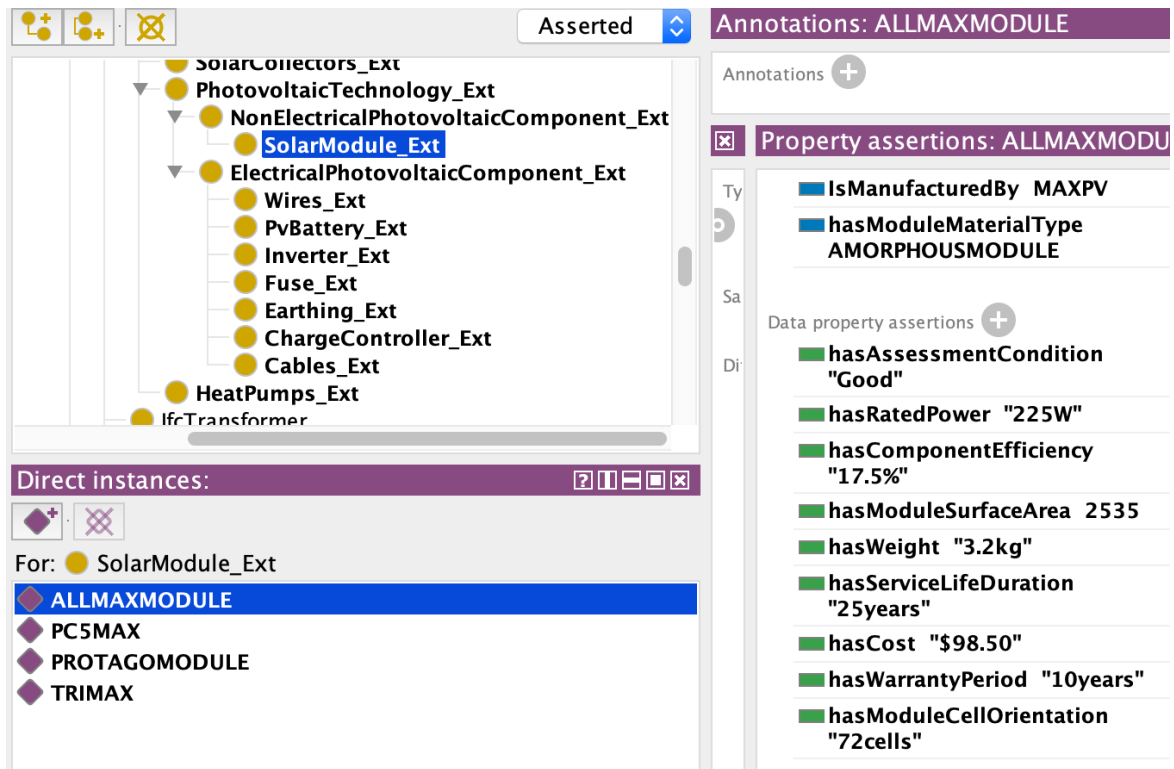


Figure 7.15: Instances added into the ontology by MaxPV

Naturally, when a knowledge model consists of classes, object, data, and annotation properties, restrictions, and instances, it suffices to be called an ontology. The IfcOWL_pvOntology obtained here is a comprehensive knowledge model of PV technology in the context of IFC. The OntoGraf protégé plugin is used for easy visualisation of the class hierarchies within the ontology as shown in figure 7.16.

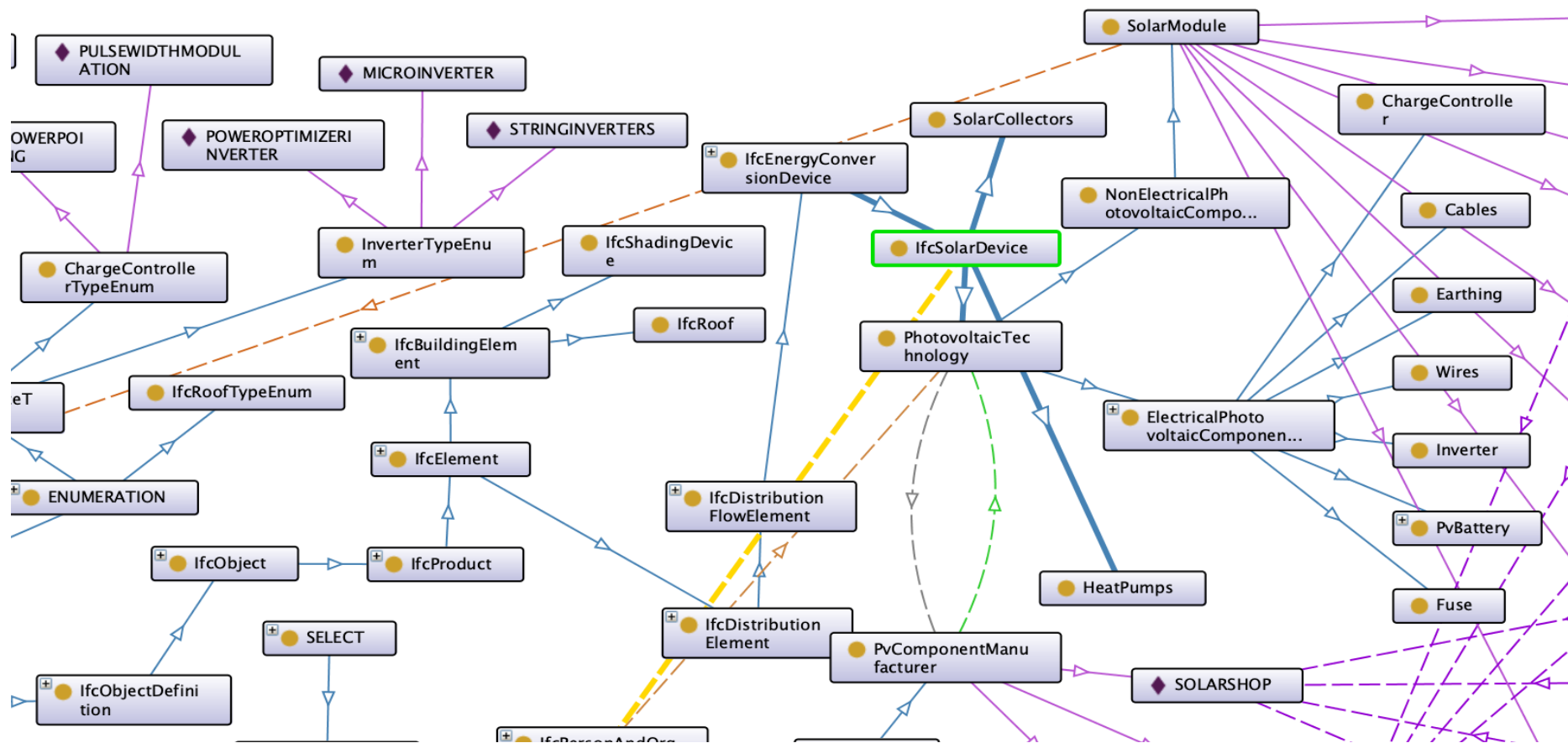


Figure 7.16: Visualisation of Class Hierarchies in Protégé for MaxPV.

It is important to note that the resulting ontology has further semantic requirements to fulfil. Automated reasoning, rules, queries, computations, and inferences on the data contained in the ontology are provided and explained in chapter 8.

7.8. Conclusion

This chapter discussed the design and the first stage of the implementation of the prototype system for this research. The approaches used were elaborated. Detailed definition of the system requirements and desired functionalities expected from the system was provided. An explanation was provided on how the software components and tools adopted for use were used within the presented system architecture. An illustration of the system architecture shows all the implementation stages.

The ontology was developed by adopting the steps outlined within the ontology development 101 methodology, using the ontology editing software protégé-OWL. Processes involved included techniques used to extend IfcOWL ontology. Entities from the conceptual knowledge model (in UML) developed in Chapter 6 were correctly mapped to OWL equivalents in protégé-OWL through adopting standard transformation rules. Corresponding OWL classes and properties were obtained. Other steps included adding facets of classes, and enhancing the knowledge model with restrictions and cardinalities, addition of instances and validation of the knowledge model using reasoners. The most important part of this chapter is that the ifcOWL ontology was successfully extended with PV technology entities, making sure that all classes, properties, and restrictions were modelled in the context of IFC's and are consistent with IFC structure.

The resulting ontology has further semantic requirements to fulfil. The second part of the implementation (Chapter 8) extends the OWL knowledge base to include queries, and computations that would enable the ontology to fulfil the objectives it has been developed for. Once the prototype system is fully implemented and recommends suitable PV technology products to users, it is evaluated to ensure completeness and correctness. Comprehensive validation and verification of the system is provided in chapter 9.

8. DEVELOPMENT OF SEMANTIC WEB IFC-BASED PHOTOVOLTAIC SYSTEM (SIPS)

8.1. Introduction

The target of the implementation in this study is 2-fold. In Chapter 7, we developed an ifcOWL-pvOntology for photovoltaic technology domain that is inspired by the IFC standard. Here, we propose a web-based application called the **Semantic web IFC-inspired Photovoltaic System (SIPS)** application, which reads in the developed ontology and executes computations to provide recommendations on photovoltaic technology products to the system users.

In this chapter, the system requirements and specifications for the development of SIPS are presented, which serve as a guide to ensure the application developed is functional and efficient. The chapter discusses software components and tools that are utilised. Furthermore, we presented a system architecture that clearly illustrates the back end and front-end of the application and the communication they exchange between them. Consequently, the chapter proceeds to show proof of concept for this research. Detailed discussions for development of the web-based application are presented within sections and subsections in the chapter. Firstly, the required system configuration is set up to establish the appropriate build paths for libraries of the software components and tools used. These libraries support development of semantic web applications from existing ontology. The chapter further analysed how these software components and tools facilitate an efficient extraction of the contents of the ontology including classes, instances, and properties. Extracted data are stored in a java file and used in the chapter to execute component sizing and computations. Furthermore, a back-end file is created to handle all algorithms for PV sizing and recommendations while a front-end interface is subsequently developed where the user puts in requirements into a form and submits. This action prompts the system to create a connection to the back end of the application through a servlet container in the tomcat server. The computations are executed within the servlet and the server presents the user with the dynamically obtained recommendations through HTTP response status codes to the front end of the homepage. The chapter provides details of these processes. Recommendation given to the user is displayed on the user interface. Screenshots are given to enhance understanding.

8.2. System Requirements and Specifications

In Chapter 7, detailed system requirements were outlined for the development of IfcOWL_pvOntology. This section presents the requirements and specifications needed to develop a web-based application. The proposed system (SIPS) requirements should elaborate how PV-technology can be recommended through a semantic web application designed using IFC standard terminology for building products. Using the system requirements format proposed by the agile software development methodology (Pocsova et al., 2020), the features, behaviour, and functionalities to be fulfilled by SIPS are identified and outlined in Table 8.1 below.

Table 8.1: System Requirements for SIPS

Purpose <ul style="list-style-type: none">• SIPS is proposed to demonstrate the extent to which semantic web technologies can be applied to provide recommendations on the use of photovoltaic systems technology.• Its purpose is to help its users make informed decisions on PV technology products.
Scope <ul style="list-style-type: none">• The domain is photovoltaic technology due to the vastness of the SBTP domains.
Intended Users <ul style="list-style-type: none">• Anyone looking to select and install PV technology• SBTP manufacturers looking to represent their products online using semantic web languages
General Requirements Knowledge base:

- The web-based application must have access to an ontology; in the case of this study - IfcOWL_pvOntology. Ontology should be read, and data should be easily retrieved.

Software product:

- System should be interoperable.
- System should be open source: It is hoped that SIPS will be re-used, extended and expanded in the future to cover the entire domain of sustainable building technology products.
- System should be agile: this implementation adopts the agile software development methodology and develops the prototype system incrementally.
- System must take in input parameters from users.
- System must exchange knowledge with the ifcOWL-PvOntology
- System must make necessary computations and recommend the most suitable PV technology and design to the user.

Non-Functional Requirements

- Look-and-feel requirements: system will have a user interface for global use.
Fit criteria: > 80% approval.
- Usability requirements: product will be made easy to use.
Fit criteria: error rate of <1%
- Performance requirement: recommendations will be administered to users in a timely manner. Fit criteria: Recommendation is made in < 5 mins. Quality assurance tests will be conducted.
- Optional requirements: system should simulate recommended PV-technology and provide a detailed report to the user.
- Portability requirements: system should be flexible. System will be web-based and can be accessed by anyone in the world.
- Security requirements: personal information of users will not be asked by the system.
- Legal requirements: SIPS is a prototype system and does not require a license yet.

Functional Requirements

Web-based Application

- Add user geographical location
- Based on locations' meteorological data, recommend a sustainable energy technology

Fit criteria: A sustainable energy technology is recommended.

- Add additional user requirements
- Read content of the ifcOWL-PvOntology and parse the values to relevant fields into the user interface.
- Undertake component and overall PV sizing and design
- Recommend PV technology design.

Fit criteria: PV technology design has been recommended.

Developing a web-based application as part of this study's implementation is ideal. It is imperative for information to be accessed and moved across different heterogeneous applications, platforms, or machines without constraints to ensure interoperability. To ensure that these requirements are met, the most suitable software development components and tools are selected. The ensuing section outlines these components and shows how they are adopted.

8.3. Software Components and Tools

Implementation of the proposed web-based application required the use of several software technologies and tools. A communication link is established between the IfcOWL_pvOntology developed in chapter 7 to the development environment adopted for the web-based application. Several software components and tools such as Jena API, java and JSP have been adopted. The ensuing section presents these tools and shows how they are applied throughout the implementation.

8.3.1. Platform and Development Framework

The result of this prototype system is web-based, interoperable and platform-independent, therefore we do not worry about the platform on which it is

developed. Furthermore, the development environments or editors used in this chapter are available in several platform-compatible formats. Therefore, they were installed based on the platform being used by the developer. The development environment used in this study is eclipse.

- Eclipse IDE - Implementation of SIPS required the use of several software developing applications/tools in a single graphical user interface (GUI). Typically, these multiple tools required are manually configured, set up and integrated within the software development editor. To avoid this tedious manual process, and to save time, an integrated development environment (IDE) called Eclipse is used. Eclipse IDE allows the developer to organise workflow to solve problems. It allows syntax highlighting and conducts automatic debugging. What makes eclipse most suitable to this study is its ability to support the numerous programming languages that are required to establish a communication link between IfcOWL-PvOntology and manufacturer websites. It executes actions using a single graphical user interface (GUI) without switching between applications.

8.3.2. Programming Languages

The programming languages used in this chapter are Java programming language (java), HyperText Markup Language (HTML) and Jakarta Server Pages (JSP).

- Java programming language is one of the most popular desktop and web-based applications development languages. It is suitable for this implementation because of its simple and efficient nature of asserting algorithms. Java is also highly fitting for this study because it is designed to run on multiple platforms. The principal algorithm used in this chapter to connect the developed ontology with the web-based application is contained within a java file. The content of this java file will be explained later in the chapter.
- HyperText Markup Language (HTML) – is adopted because it is the standard markup language for applications designed to be displayed in a web browser. In

this study, only the homepage is written in .html. It is assisted by cascading style sheets which help to define the refinement of the pages.

- Jakarta Server Pages (JSP) - is used to develop the graphical user interface layer of the prototype. This programming tool supports development of platform-independent web pages. JSP is adopted in this study mainly because it can read data in RDF and can insert java codes and RDF into HTML or XML pages or both (<% java code %>) depending on which format the developer wishes to use. This implementation contains several .jsp files can contain algorithms for selection of suitable photovoltaic technology components. Execution of JSP requires the use of a server. The server deployed for this study is highlighted in the subsequent subsection.

8.3.3. Web Server

The web server connects the developed application to the web and establishes a user/server model. A part of the server known as the servlet container executes defined operations. The server accepts requests from the user and forwards it into the servlet container to execute the operation. The server obtains results from the servlet and forwards it to the user at the front end of the webpage. The server and servlet used in the study is Apache Tomcat.

- Apache Tomcat – in the context of this study, tomcat is used as both the server and servlet container. It reads in the requirements the user inputs and sends it to the servlet container via a hypertext transfer protocol (HTTP) request code. The computations are executed within the servlet and the server presents the user with the dynamically obtained recommendations through HTTP response status codes.

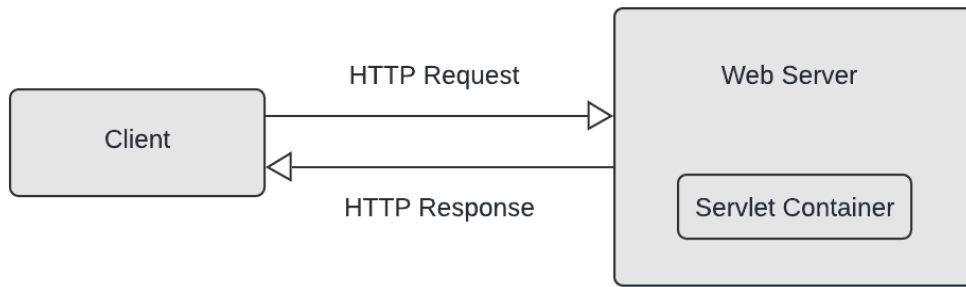


Figure 8.1: An example of a servlet container

8.3.4. Application Programming Interface (API)

APIs are programming interfaces that allow applications to access certain features or data from other software applications. It facilitates interactions between multiple software applications. The Jena API is adopted for use in this study.

- Jena - is a java API, available within JSP and eclipse. It is adopted for this study because its framework supports development of semantic web applications and interaction with ontology concepts. Jena has features that allow knowledge in RDF format to be read, created, and exploited. In this implementation, Jena reads and interacts with the IfcOWL-pvOntology within the JSP files, through simple lines of codes that will be elaborated on later in the chapter.

8.4. System Architecture

As explained in Chapter 7, this study uses an N-tier system architecture for its implementation because the prototype system involves connections and communication between multiple software components and layers.

This architecture contains two main components: the front-end and back-end. In Chapter 7, the back end of the system architecture was presented including ontology development in Protégé-OWL. This chapter presents the remaining components within the back end of the system. This includes the process of reading in the ifcOWL-pvOntologies and processing the content of the ontology to perform computations and recommend PV technology products. The front-end of the system is fundamentally the user interface layer. It includes a graphical interface through which

the client interacts with the system by inputting data and getting back desired recommendations. The front-end interface also includes several manufacturer websites (generated by the system) where PV technology components are displayed in RDF. Communication between the front-end and the back end is established through a communication servlet within the server. Figure 8.1 below shows the parts of the system architecture implemented in this chapter.

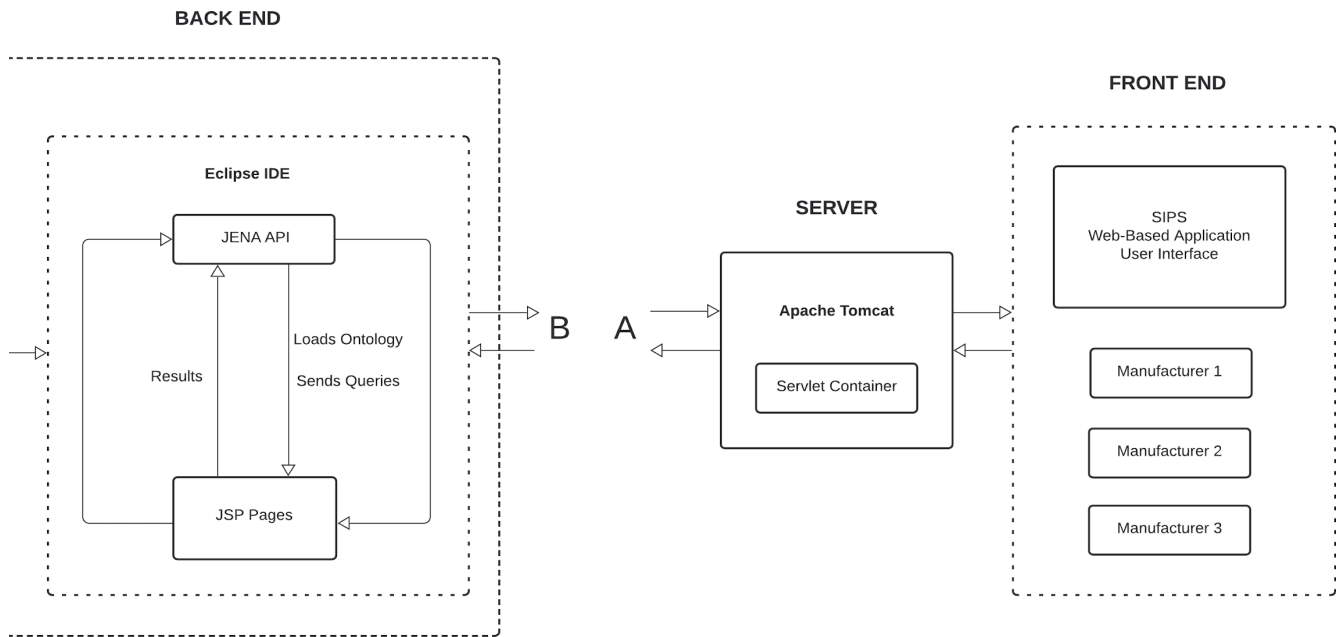


Figure 8.2: Fragment of System Architecture

The system architecture presented above shows how the software components and tools adopted for use in this study are applied.

8.5. Development of Semantic Web-Based Application

This section describes the processes involved in the development of the web application aspect of the prototype systems. The first step is to set up the back-end configuration. Recall in chapter 7 that the ifcOWL_pvOntology is saved in RDF to allow further implementation. Also recall that three product manufacturers have added instances into the ifcOWL-pvOntology. The resulting ontologies are IfcOWL_pvOntologyMaxPv.rdf, IfcOWL_pvOntologySimbaSolutions.rdf, and IfcOWL_pvOntologySolarShop.rdf. These three ontologies will be read into eclipse IDE, where component sizing and

computations will be performed. The process of developing the front end graphical user interface is then deliberated.

8.5.1. Back-End System Configuration

Eclipse comes in different formats depending on the purpose of implementation. Since the aim is to develop a web-based application, the study adopts the Eclipse IDE platform for enterprise java and web developers with incubating components. The latest version (2021-06) of this format is unstable and has continuous installation and run-time errors that need to be corrected and updated by the developers. Therefore, we used version 2020-06, with a build ID 20200615-1200 for this implementation.

Once eclipse is downloaded, certain configurations are required to make the development environment viable to run effectively. The first step is to download a compatible Jena (to load ontology), Apache tomcat (server) and Java Development Kit (JDK) for executing and compiling java codes. The second step is to configure the build path within eclipse for the project appropriately. Primarily, the Java Runtime Environment (JRE) (contained within the JDK) is configured using the java build path window from the eclipse properties. Subsequently, build paths for Jena libraries and Apache tomcat are configured so they can also be read within eclipse libraries. Figure 8.3 below shows how we configured the build path for Jena, Java SE 16.0.2, and apache tomcat v9.0 for the purpose of this study. The Java and JSP files should be able to have straightforward access to the jar files within the libraries. The slightest changes in these software configurations affects the application and may result in runtime errors. Step by step guide to set up and configure Apache tomcat, Jena and java JDK appropriately can be found in (Ameen, Khan and Rani, 2014; programcreek, 2023).

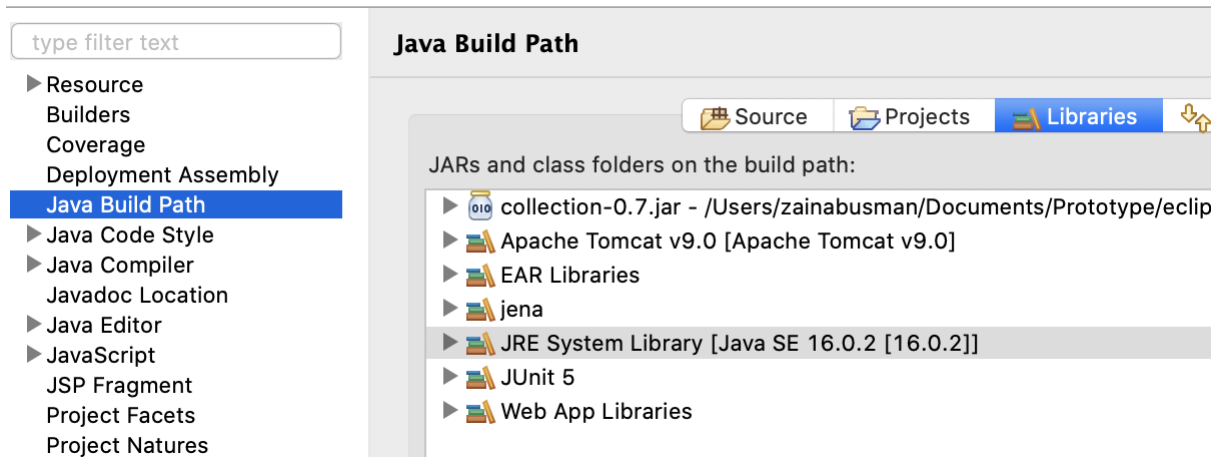


Figure 8.3: Java build path within Eclipse IDE

Once the software tools are successfully configured within eclipse, implementation can begin.

8.5.2. Extraction of Knowledge from Ontology into Eclipse Using Jena

As explained in section 8.3 above, Jena API is adopted to develop the semantic web IFC inspired application. Jena provides classes, interfaces and libraries containing .jar files that allow creation, manipulation, and interaction with semantic web data (either in RDF or OWL). The initial step in the implementation of this web-based application entails successfully loading the ontology developed in protégé into eclipse to read and display the classes, properties, and instances of the ontology. This consists of the following algorithmic steps:

- Create a new java project in Eclipse – a new java project called *Shams* is created. Because this is an enterprise java and web developers project, it automatically creates two main folders: A Java Resource and a WebContent folder. In the Java resource folder, we create a package *com.shams* that will hold all java files and libraries and the developed ontology. The web content folder is used in section 8.5.3 below.
- Load Jena libraries into Eclipse – the next step is to ensure that Jena libraries are imported correctly into the project through the ‘properties’ settings of the project. The Java build path is set up in a way it allows new libraries to be imported. Therefore, we import all the .jar files within the Jena library from the Jena folder

stored on the computer. The back-end configuration and build path of Jena is then cross checked to ensure it is accessible.

- Ontology – the next step is to ensure the different ontologies containing instances of manufacturers (IfcOWL_pvOntologyMaxPv.rdf, IfcOWL_pvOntologySimbaSolutions.rdf, and IfcOWL_pvOntologySolarShop.rdf) are saved in the appropriate location within the eclipse workspace. These ontologies are saved within the project *Shams* under the *src* and the created package *com.shams*.
- Create a java file – firstly, we create a *PvSemanticWeb.java* file within the java project. This java file is primarily used to import the developed ontology and to parse the extracted data to be used by the web-based application. Once the file is created, its package statement (*package com.shams.jsp;*) is automatically added.
- Import packages - The second step is to provide codes to import all packages required to support development of semantic web applications. The import statement is written after the package statement but before class definition. These imports include all required java packages (that provide access protection, namespace management and packages that support semantic application functionalities); Jena Apache packages (that support ontology, model, util, and iterator functionalities) and javax servlet packages (that contain classes that define the connection between the servlet and the runtime environment).
- Define a class - The third stage involves definition of class which carries the same name as the java file (*public class PvSemanticWeb {}*). Within this class, we declare a default namespace to hold semantic data as a *static String defaultNameSpace = "http://semanticweb.org/ontologies";*
- Define methods – the fourth and vital stage is the definition of several methods within the class that perform several roles in extracting the ontology. This begins with an automatic entry of the main class using command line arguments as an array of strings.

```
public static void main (String [] args) throws IOException {
```

```
// TODO Auto-generated method stub
PvSemanticWeb load = new PvSemanticWeb ();
```

The IO exception means this method could potentially “throw” an exception. Once this main method is successfully declared, other methods are subsequently asserted to perform different functions. The most important method defined is the *getOntology()* method. This model creates an ontology model that opens the *IfcOWL_pvOntologyMaxPv* into the java project and saves it within the created model created using the codes shown below:

```
public void getOntology() throws IOException{
    schema = ModelFactory.createOntologyModel();
    inschema =
    FileManager.get().open("./ifcOWL-pvOntologyMaxPv.rdf");
    schema.read(inschema,defaultNamespace);
```

Once the ontologies have been called, an iterative process of retrieving all classes, instances, and properties is followed. The subsequent method created returns an extended iterator over all the classes existing within the *IfcOWL-pvOntologyMaxPv*.

```
ExtendedIterator<OntClass> it = ((OntModel) schema).listClasses();
    while(it.hasNext()) {
    OntClass cls = (OntClass)it.next(); //Loops through the classes of the ontology
        System.out.println(cls.getLocalName());
```

The focus of this study is the *SolarModule* class that is a subclass of *PhotovoltaicTechnology* and *IfcSolarDevice* as defined in chapter 7. Therefore, within the loop that goes through the list of classes within the ontology, we instruct the iterator to find and retrieve the *SolarModule* class, its instances, and all properties. The following codes list out all the instances within the class *SolarModule*.

```
if(cls.getLocalName().equals("SolarModule")) {
```

```
System.out.println("classes of the ontology: " + cls.getLocalName() + "\n");
```

```
ExtendedIterator Instances = cls.listInstances();
```

```
while(Instances.hasNext()){
```

```
Individual currInstance = (Individual)Instances.next();
```

```
System.out.println("Individual of SolarModule Class:" +  
currInstance.getLocalName() + "\n");
```

As explained in chapter 7, numerous object and data properties are defined in the ontology. However, not all are given values when instances are defined. The local names of the object and data properties with given values for the instance of class *SolarModule* are retrieved by calling the *listProperties()* method as shown below. The loop begins by reading an instance then retrieves all its properties and their values before it moves to the next instance.

```
if(currInstance.getLocalName().equalsIgnoreCase("")){ // case ignored because  
individuals are named in caps
```

```
System.out.println("Individuals of SolarModule Class: " +  
currInstance.getLocalName());
```

```
for(StmtIterator ti= currInstance.listProperties(); ti.hasNext();){
```

```
Statement t = ti.next();
```

```
if(t.getPredicate().getLocalName().equals("type") //
```

```
t.getPredicate().getLocalName().equals("topDataProperty")) continue;
```

```
String propertyName = t.getPredicate().getLocalName();
```

```
String propertyValue = ""; }
```

At the end of this loop, all required instances on the class along their object and data properties and their values would have been retrieved from the ontology. Furthermore, note that this loop only calls instances from one manufacturer; IfcOWL_pvOntologyMaxPv.rdf. Therefore, additional loops are executed to call instances from the other two manufacturers' ontologies IfcOWL_pvOntologySimbaSolutions.rdf and IfcOWL_pvOntologySolarShops.rdf.

Since the instructions in a loop are continually repeated until a certain condition is reached, the *getOntology()* method is called within each loop to call the stated ontology and save it within the created java model. This is followed by an iterative process of traversing all the classes, instances and defined properties within that ontology as explained above. Once all ontologies are traversed, the execution terminates. This approach of using a loop to run through available manufacturer ontologies means the system can compare products from many manufacturers (value from 1...n ontologies). This makes this prototype system scalable.

- Define HashMaps – the fifth stage of extracting knowledge from the ontology involves the use of HashMaps to store the retrieved data. HashMap is a data structure in java programming. It allows objects to be stored and retrieved in constant time O(1) provided the object key is known. HashMaps are used in this study to store keys and values of properties of instances in the form of a *Map.Entry*. where an object entry is called by *get(key)* and it returns a key and a value *put(key, value)*. When we call a *put* method, the hash function of the map finds a bucket location to store the value of the object. When we wish to retrieve the object, we call the *get()* method and provide the key which generates the same hash codes to arrive at the same bucket location. If only one object is present the code returns the value of the object you had stored earlier. It is important to note that HashMap values are unalterable.

Within the class assertion, we place new HashMap statements to store all keys and values for properties of instances in a map. The codes below show a HashMap declaration *HashMap<String, String>* into a Map for an instance variable *trimax*.

```
public static Map<String,String> trimax = new HashMap<String,String>();  
//stores properties of the instance trimax in a Map
```

Within the main method of the java class, the following code *getInstances(trimax)*; calls the *getInstances()* method to present all the stored keys and values of properties. The *getInstances()* method is defined using the following codes.

```

public static void getInstances(Map<String,String> map) {
    for(String s: map.keySet()){
        System.out.println( s + " -> " + map.get(s)); }

```

A fragment of some of the codes used in this section are seen in figure 8.4 A, B, and C below.

```

eclipse-workspace - shams/src/com/shams/jsp/PvSemanticWeb.java - Eclipse IDE
PvSemanticWeb.java
82 public static void getOnto() throws IOException {
83     PvSemanticWeb load = new PvSemanticWeb();
84     load.getOntology();
85 }
86
87 public void getOntology() throws IOException{
88     schema = ModelFactory.createOntologyModel();
89     InputStream inschema = FileManager.get().open("./ifcOWL-pvOntologyMaxPv.rdf");
90     schema.read(inschema, defaultNameSpace);
91
92     ExtendedIterator<OntClass> it = ((OntModel) schema).listClasses();
93
94     while(it.hasNext()) {
95         OntClass cls = (OntClass)it.next(); //Loops through the classes of the ontology
96         //System.out.println(cls.getLocalName());
97         if(cls.getLocalName().equals("SolarModule")) {
98             System.out.println("A class of the IfcOWL_pvOntology: " + cls.getLocalName() + "\n");
99             System.out.println("URI: " + cls.getURI() + "\n");
100
101             ExtendedIterator Instances = cls.listInstances();
102             while(Instances.hasNext())
103             {
104                 Individual currInstance = (Individual)Instances.next();
105                 //System.out.println("Individual of SolarModule Class:" + currInstance.getLocalName
106                 if(currInstance.getLocalName().equalsIgnoreCase("tesla")) // case ignored because
107                 {
108                     //System.out.println("Individual of SolarModule Class: " + currInstance.getLoca
109
110                 for(StmtIterator ti= currInstance.listProperties(); ti.hasNext();) {
111                     Statement t = ti.next();
112
113                     if(t.getPredicate().getLocalName().equals("type") || t.getPredicate().getLo
114                     String propertyName = t.getPredicate().getLocalName();
115                     String propertyValue = "";
116                     if(propertyName.equals("IsInvolvedIn") || propertyName.equals("topObjectPro
117
118                     if(propertyName.equals("IsManufacturedBy")) {
119                         propertyValue= t.getResource().getLocalName();
120                         tesla.put(propertyName, propertyValue);
121                     }

```

Figure 8.4 (A): Codes for Extracting Ontology Knowledge into Eclipse from IfcOWL_pvOntologyMaxPV Manufacturer

```

/SemanticWeb.java
    }
    break;
}
//inschema.close();
}
//schema.close();
InputStream inschema1 = FileManager.get().open("./ifcOWL-pvOntologySimbaSolarSolutions.rdf");
schema.read(inschema1, defaultNamespace);

ExtendedIterator<OntClass> it1 = ((OntModel) schema).listClasses();

while(it1.hasNext()) {
    OntClass cls = (OntClass)it1.next(); //Loops through the classes within IfcOWL_pvOntologySimba
    //System.out.println(cls.getLocalName());
    if(cls.getLocalName().equals("SolarModule")) {

        ExtendedIterator Instances = cls.listInstances();
        while(Instances.hasNext())
        {
            Individual currInstance = (Individual)Instances.next();
            //System.out.println("Individual of SolarModule Class:" + currInstance.getLocalName()

            if(currInstance.getLocalName().equalsIgnoreCase("Neon"))
            {
                for(StmtIterator ti= currInstance.listProperties(); ti.hasNext();) {
                    Statement t = ti.next();
                }
            }
        }
    }
}

```

8.4 (B): Figure 8.4 (A): Codes for Extracting Ontology Knowledge into Eclipse from IfcOWL_pvOntologyMaxPV Manufacturer

```

SemanticWeb.java
    }
    break;
}
}

InputStream inschema11 = FileManager.get().open("./ifcOWL-pvOntologySolarShop.rdf");
schema.read(inschema11, defaultNamespace);

ExtendedIterator<OntClass> it11 = ((OntModel) schema).listClasses();

while(it11.hasNext()) {
    OntClass cls = (OntClass)it11.next(); //Loops through the classes IfcOWL_pvOntologySolarShop
    //System.out.println(cls.getLocalName());
    if(cls.getLocalName().equals("SolarModule")) {

        ExtendedIterator Instances = cls.listInstances();
        while(Instances.hasNext())
        {
            Individual currInstance = (Individual)Instances.next();
            //System.out.println("Individual of SolarModule Class:" + currInstance.getLocalName(

            if(currInstance.getLocalName().equalsIgnoreCase("jinko"))
            {
                for(StmtIterator ti= currInstance.listProperties(); ti.hasNext();) {
                    Statement t = ti.next();
                }
            }
        }
    }
}

```

Figure 8.4 (C): Codes for Extracting Ontology Knowledge into Eclipse from IfcOWL_pvOntologySolarShop Manufacturer

Once the java file is free of errors and saved, it can be run as a java application. The output after successfully running the codes is shown in figure 8.5 below.

```

591 public static void getInstances(Map<String,String> map) {
592     for(String s: map.keySet())
593         if(!map.isEmpty())
594             {
595                 System.out.println( s + " -> " + map.get(s));
596             }
597     }
598 }
599

```

<terminated> PvSemanticWeb [Java Application] /Library/Java/JavaVirtualMachines/jdk-16.0.2.jdk/Contents/Home/bin/java (20 No
 URI: <http://www.semanticweb.org/zainabusman/ontologies/2021/5/IfcOWL-pvOntologyMaxPV#SolarModul>
 ***** Trimax *****
 IsManufacturedBy -> MAXPV
 hasComponentEfficiency -> 21%
 hasServiceLifeDuration -> 25years
 hasWeight -> 2kg
 hasAssessmentCondition -> Excellent
 hasRatedPower -> 410W
 hasModuleSurfaceArea -> 1520
 hasCost -> \$205
 hasModuleCellOrientation -> 60cells
 hasModuleMaterialType -> MONOCRYSTALLINEMODULE
 hasWarrantyPeriod -> 10years
 ***** ProtagoModule *****
 IsManufacturedBy -> MAXPV
 hasComponentEfficiency -> 20.9%
 hasServiceLifeDuration -> 25years
 hasRatedPower -> 315W
 hasAssessmentCondition -> Excellent
 hasWeight -> 2.4kg
 hasModuleSurfaceArea -> 1650
 hasCost -> \$156.90
 hasModuleCellOrientation -> 60cells
 hasWarrantyPeriod -> 10years
 hasModuleMaterialType -> MONOCRYSTALLINEMODULE
 ***** PC5Max *****
 IsManufacturedBy -> MAXPV
 hasServiceLifeDuration -> 25years
 hasComponentEfficiency -> 15.9%
 hasWeight -> 3.8kg
 hasAssessmentCondition -> Good
 hasRatedPower -> 260W
 hasModuleSurfaceArea -> 3042
 hasCost -> \$60

Figure 8.5: IfcOWL_pvOntology Knowledge Obtained at Runtime in Eclipse

We have successfully completed the extraction of knowledge we require from the ontologies into eclipse using Jena API, classes, and libraries. The first line of the output shows the URI of the class retrieved from IfcOWL-pvOntologyMaxPV. Successively, the instances of the class are printed with a list of its properties and their values. It prints

out all instances of *SolarModule* requested from all 3 ontologies. These values can now be applied to computations that facilitate the design of Photovoltaic Systems.

8.5.3. Photovoltaic Component Sizing and Computations

Before going into the implementation of the component sizing and computations, we refer to discussions done earlier in previous chapters that have also been published by the author of this study.

- Decision model (Figure 6.3A) developed in Chapter 6 - Recall from section 6.3.2.2, a comprehensive process/decision model is proposed with detailed steps that guide towards the design and selection of PV-system. The decision model is backed by mathematical computations that support the design process we developed and presented. Detailed design decisions and proof of use and effectiveness is given in chapter 6. Some aspects of it have also been published in Usman *et al.*, (2020a).
- Assessment of poor performance of installed PV systems done in Chapter 2 shows systems do not generate the required electric output they have been designed to produce. Using performance assessment parameters, we critically appraised factors that lead to poor performance of PVS and concluded that different design methods lead to significant differences in performance. Recall the chapter also thoroughly investigates the different methods, their advantages and disparities and analyses the variations in designs. We provided recommendations for the ultimate design approach backed by detailed mathematical computations presented alongside the process model proposed in Chapter 6. The mathematical formulae proposed in this study form the basis for the algorithm used in this section to size the photovoltaic components. These formulae have gone through rigorous peer review in author's journal publication (Usman et al., 2020b).
- Trade-off between budget, efficiency, and roof space – In Chapter 2, we deliberated on the major differences of solar modules, its efficiency, cost, and

size. The user often considers these three options when trying to determine the type and size of their photovoltaic system. High efficiency modules generate more electric output than a similar sized module of lower efficiency. They also require smaller mounting space. However, the higher the efficiency of the technology, the more expensive it is. When the cost of installation is too high for the user, a less efficient and less expensive module type is considered if sufficient mounting surface area is available. Therefore price, efficiency and surface area are key trade-offs. Where surface area is involved, the size of the array should always be less than or equal to the mounting area: $\text{ArraySurfaceArea} \leq \text{MountingArea}$. Conversely, extensive review shows that the average residential building always has sufficient roof (or other mounting types) surface area to install the required size of photovoltaic components (Palmetto, 2021). Therefore, this implementation will overlook the mounting surface area of the user.

The decision model takes in four parameters as user requirements: building energy load, solar radiation of location, mounting surface area and budget. The decision to use these parameters have been discussed in chapter six. In terms of the energy load of the building, it is imperative to understand that some users prefer to optimise their photovoltaic system for longer duration of discharge thereby powering the household at an average capacity. Others may have high peak demands and will prefer to enhance energy generation which will result in a larger and more expensive system. Because there are huge disparities in terms of user needs, energy load profiles and storage requirements, a single system of measurement will not satisfy the varying system requirements. The proposed system in this study establishes the minimum components' sizes required to generate the average electrical energy output the user requires. Energy generated would be sufficient for eight months out of the year. Little excesses generated from the other four months could be stored in a battery bank to use during the period of peak energy demands (where the energy load of the building is higher than the electric energy generated by the PV-system).

Recall the project created in section 8.5.2 above is an enterprise java and web developers project, which upon creation generated a Java resource folder (used in

section 8.5.2 above to hold all java files and libraries) and a Web Content folder. In this section, we utilise the latter folder. In the WebContent folder, we create three sub folders for Images, META-INF, and WEB-INF. Under the WEB-INF, we create a .jsp file to handle Photovoltaic computations and sizing. This file is named estimate.jsp. The ensuing sub-sections explain seven steps we used to size photovoltaic systems to the user.

- In estimate.jsp file, the first step is to import the java file (PvSemanticWeb.java) created in section 8.5.2 above that extracted knowledge from ifcOWL-pvOntology into eclipse using Jena, read in the contents of the ontology, and saved them into HashMaps. `<%@page import="com.shams.jsp.PvSemanticWeb"%>`. This import makes all components and their values parsed into the HashMaps available for further applications.
- The second step is to create the html environment which includes the `<head>` `<title>` `<body>` for the jsp file. We then call the HashMaps and `getOnto()` methods from the PvSemanticWeb.java file to retrieve stored ontology.

```
PvSemanticWeb.getOnto();
```

```
java.util.Map<String,Double> values = new java.util.HashMap<String,Double>();
```

- Thirdly, we set up codes to accept the four input parameters that the user will give: the budget, surface area, solar radiation, and energy load. Since *String* is the key of the HashMaps, we parse the values into doubles for further calculations.

```
String budget = request.getParameter("budget");
```

```
double b = Double.parseDouble(budget);
```

```
String surfaceArea request.getParameter("size");
```

```
double surfa = Double.parseDouble(surfaceArea);
```

```
String energy request.getParameter("energyLoad");
```

```
double x = Double.parseDouble(energy);
```

```
String solar = request.getParameter("solar");
```

```
double y = Double.parseDouble(solar);
```

- The fourth step calculates the target watt peak rating expected from the solar modules to satisfy the energy load provided by the user through the input parameters.

From formulae if energy load is x , and solar radiation is y ;

$WattHours = x \times 10^3 \times 1.3$ where 1.3 accounts for system losses.

$WattPeakRating = wattHours \div PanelGeneratingFactor(PGF)$.

$PGF = y \times (Losses\ due\ to\ temperature,\ dirt,\ poor\ radiation,\ ageing)$.

Therefore,

$$WattPeakRating = (x \times 10^3 \times 1.3) \div (y \times (0.90 \times 0.86 \times 0.90 \times 0.95))$$

$$WattPeakRating = 2000x \div y$$

Therefore,

double moduleWattHrs = x*1000*1.3;

double result = 2000*x/y; //where result is the watt peak rating expected from the modules

- The fifth step uses the values of the result obtained above to run a loop over every single solar module in the ontology. It retrieves the rated power and cost of each module and computes to find the array size, total cost of the array, cost of inverter and battery for the system and subsequently the total cost of the components.

$$ArraySize = result \div ratedPower$$

$$CostOfArray = ArraySize \times costOfModule$$

In chapter 2. We established that the inverter and battery costs take up roughly 9% of the total component costs. Since the modules are the main components of interest, we determine the total component costs by multiplying the array costs by a ratio of 100:91. Therefore,

$$TotalCost = (CostOfArray \times 100) \div 91$$

Due to the volume of the implementation, we provide the algorithm used for only one of the modules; trimax.

```

String t = PvSemanticWeb.trimax.get("hasRatedPower"); //reads the rated power
of trimax as saved in the HashMap in PvSemanticWeb.java
t = t.replaceAll("\\D+", "");
double tOutput = Double.parseDouble(t);
String pt = PvSemanticWeb.trimax.get("hasCost"); //reads the cost of trimax as
saved in the HashMap in PvSemanticWeb.java
pt = pt.replaceAll("[\$]", "");
double ptt = Double.parseDouble(pt);
double tas = result/tOutput; //computes trimax array size
double nmn = tas * ptt; //total cost of trimax array size
double totalTrimax = nmn*1.099; //total component cost of installing Trimax
based on user requirements
double inverterBatteryTrimax = totalTrimax-nmn; //cost of inverter and battery

```

- The sixth step is cost comparison. Recall the input parameter for budget is read above and parsed into a double variable “b”. One of the main aims of this implementation is to determine which solar module whose array size is suitable to handle the target watt peak rating the user requires, comes closest to the budget of the user. Hence, once the system establishes the array size, cost of array, inverter and battery cost and the total component cost for the module type, we find the absolute difference of the budget and the total cost of the component if that module is selected.

```

double f = Math.abs(b-totalTrimax); //returns absolute difference between the
budget and trimax module, stores result in a variable f.
values.put("trimax",f);

```

- In the seventh and last step, we run a loop that reads in all absolute differences between the budget and the total component costs for each module type. This loop returns the key and value of the module whose total component costs comes closest to the budget of the user. The returned module is recommended to the user.

```

String best = "trimax";

```

```
for(String o:values.keySet()){  
    if(values.get(o) < values.get(best)){  
        best = o;  
    }  
}
```

Figure 8.6 below shows a screenshot of the implementation done on other components. The full code in estimate.jsp is available at the Github site for this study - <https://github.com/zainabusman>.

```

eclipse-workspace - shams/WebContent/estimate.jsp - Eclipse IDE
estimate.jsp
15
16 String jk = PvSemanticWeb.jinko.get("hasRatedPower");
17 jk = jk.replaceAll("\\D+", "");
18 double jkOutput = Double.parseDouble(jk);
19 String jkCost = PvSemanticWeb.jinko.get("hasCost");
20 jkCost = jkCost.replaceAll("[\$]", "");
21 Double jkS = Double.parseDouble(jkCost);
22 double jas = result/jkOutput; //array size if jinko is selected
23 double arrayJinkoRatingTotal = jkOutput*jas;
24 double nmnJk = result/jkOutput * jkS; //total cost of jinko array
25 double totalJinko = nmnJk*1.099; //total component cost
26 double inverterBatteryJinko = totalJinko-nmnJk; //inverter and battery
27 double j = Math.abs(b-totalJinko);
28 values.put("jinko", j);
29
30 String sm = PvSemanticWeb.sunModule.get("hasRatedPower");
31 sm = sm.replaceAll("\\D+", "");
32 double smOutput = Double.parseDouble(sm);
33 String smCost = PvSemanticWeb.sunModule.get("hasCost");
34 smCost = smCost.replaceAll("[\$]", "");
35 Double smS = Double.parseDouble(smCost);
36 double sunas = result/smOutput; //array size if sunModule is selected
37 double nmnSm = result/smOutput * smS; //total cost of array
38 double totalSunModule = nmnSm*1.099; //total component costs
39 double inverterBatterySunModule = totalSunModule-nmnSm; //costs of inverter and battery
40 double mi = Math.abs(b-totalSunModule);
41 values.put("sunModule", mi);
42
43 String xl = PvSemanticWeb.solarWorldXl.get("hasRatedPower");
44 xl = xl.replaceAll("\\D+", "");
45 double xlOutput = Double.parseDouble(xl);
46 String xlCost = PvSemanticWeb.solarWorldXl.get("hasCost");
47 xlCost = xlCost.replaceAll("[\$]", "");
48 Double xlS = Double.parseDouble(xlCost);
49 double soas = result/xlOutput; //array size if solarWorldXl is selected
50 double nmnXl = result/xlOutput * xlS; //total cost of array
51 double totalSolarWorldXl = nmnXl*100/91; //total component costs
52 double inverterbatterySolarWorld = totalSolarWorldXl-nmnXl; //costs of inverter and battery
53 double mxl = Math.abs(b-totalSolarWorldXl);
54 values.put("solarWorldXl", mxl);
55

```

Figure 8.6: Screenshot of estimate.jsp Showing Computations Done on Other Components

Another .jsp file was developed to size and select photovoltaic components based on the efficiency of the electric output the components would generate. Based on the review on solar modules in chapter 2, the most efficient modules produce more electricity from

every single square metre of its mounting space. For users who are not concerned about the high cost of components but the overall performance and efficiency of electric output, then the system computes which module can achieve that desired output. *Output.jsp* contains algorithms (as shown in figure 8.7 below), that read in ontology data from *PvSemanticWeb.java* and recommend the most efficient module to the user. The full code in *estimate.jsp* is available at the Github site for this study - <https://github.com/zainabusman/>. The structure of *output.jsp* and *estimate.jsp* are very similar. Both are part of the back-end implementation. For both to run, a front-end interface must be created which when run, automatically loads either *estimate.jsp* or *output.jsp* depending on the type of recommendation the user seeks. Furthermore, the interface requires a server to run the .jsp files and present results. The next section elaborates on the front-end implementation.

```

311     double eallmax = eall/10;
312     values.put("allMaxModule", eallmax);
313
314     String mNeon = PvSemanticWeb.neon.get("IsManufacturedBy");
315     String en = PvSemanticWeb.neon.get("hasComponentEfficiency");
316     en = en.replaceAll("\\D+", "");
317     double eNe = Double.parseDouble(en);
318     double eNeon = eNe/10;
319     values.put("neon", eNe);
320
321     String mPerc = PvSemanticWeb.perc.get("IsManufacturedBy");
322     String eper = PvSemanticWeb.perc.get("hasComponentEfficiency");
323     eper = eper.replaceAll("\\D+", "");
324     double ePe = Double.parseDouble(eper);
325     double ePerc = ePe/10;
326     values.put("perc", ePerc);
327
328     String mja = PvSemanticWeb.jasolar.get("IsManufacturedBy");
329     String eja = PvSemanticWeb.jasolar.get("hasComponentEfficiency");
330     eja = eja.replaceAll("\\D+", "");
331     double ejasol = Double.parseDouble(eja);
332     double ejasolar = ejasol/10;
333     values.put("jasolar", ejasolar);
334
335     String malpha = PvSemanticWeb.alpha.get("IsManufacturedBy");
336     String eal = PvSemanticWeb.alpha.get("hasComponentEfficiency");
337     eal = eal.replaceAll("\\D+", "");
338     double ealp = Double.parseDouble(eal);
339     double ealpha = ealp/10;
340     values.put("alpha", ealpha);
341
342     String ultman = PvSemanticWeb.ultron.get("IsManufacturedBy");
343     String ult = PvSemanticWeb.ultron.get("hasComponentEfficiency");
344     ult = ult.replaceAll("\\D+", "");
345     double eult = Double.parseDouble(ult);
346     double efultron = eult/10;
347     values.put("ultron", eult);
348
349     String best = "trimax";
350
351     for(String o:values.keySet()){
352         if(values.get(o) > values.get(best)){
353             best = o;
354         }
355     }
356

```

Figure 8.7: Codes from output.jsp

8.5.4. Front-End - Client User Interface

The front-end of the system is fundamentally the user interface layer. Here, we develop a friendly graphical user interface through which the client interacts with the system by inputting data and getting back desired recommendations. The user interface is developed under the WEB-INF folder within the java web enterprise project. We begin by creating an html file named calculate.html. This file becomes the main homepage of

this web-based application. It is developed to be friendly and simple for the user's benefits. The home page is set up in two ways: appearance and functionality.

For its appearance, we set up the structure of the html page to include the <head> and the <body> and give the file a title, Solar Calculator. `<div id="RDF-header-title"> Solar Calculator</div>`. We reference the file to a Cascading Style Sheet (CSS) that contains codes which determine the fonts, colours, spacing, and other parameters that determine the overall presentation of `calculate.html`. After the title bar is set-up, a left menu bar is created to hold references to manufacturer websites that will be represented in the next section.

The functionality of this page is straightforward. Its main purpose is to allow the user to input requirements and present obtained results back to the user. Codes used to design the form that allows the user to input in parameters is provided in Figure 8.8 below. Once `calculate.html` file is run, the front end provides two forms to the user, to select either to get recommendation based on budget (as shown in Figure 8.9 below) or based on electric output (as shown in figure 8.10 below), assuming the user inputs a budget of \$2,500, an energy load of 10kWh, and solar radiation of 4.9.

```
calculate.html
47 <h1>
48     Get Recommendation Based on Budget
49 </h1>
50 <form action="estimate.jsp">
51     Budget ($): <input type="text" name="budget" />
52     <br/><br/>
53     Energy Load of Building (kWh): <input type="text" name="energyLoad" />
54     <br/><br/>
55     Solar radiation of location : <input type="text" name="solar" />
56     <br/><br/>
57     <input type="submit" value="RecommendByBudget" />
58 </form>
59 <h1></h1>
60 <h1></h1>
61 <h1></h1>
62 <h1>
63     Get Recommendation Based on Output
64 </h1>
65 <form action="output.jsp">
66
67     Energy Load of Building (kWh): <input type="text" name="energyLoad" />
68     <br/><br/>
69     Solar radiation of location : <input type="text" name="solar" />
70     <br/><br/>
71     <input type="submit" value="RecommendByOutput" />
72 </form>
73
```

Figure 8.8: Codes that allow the user to input requirements

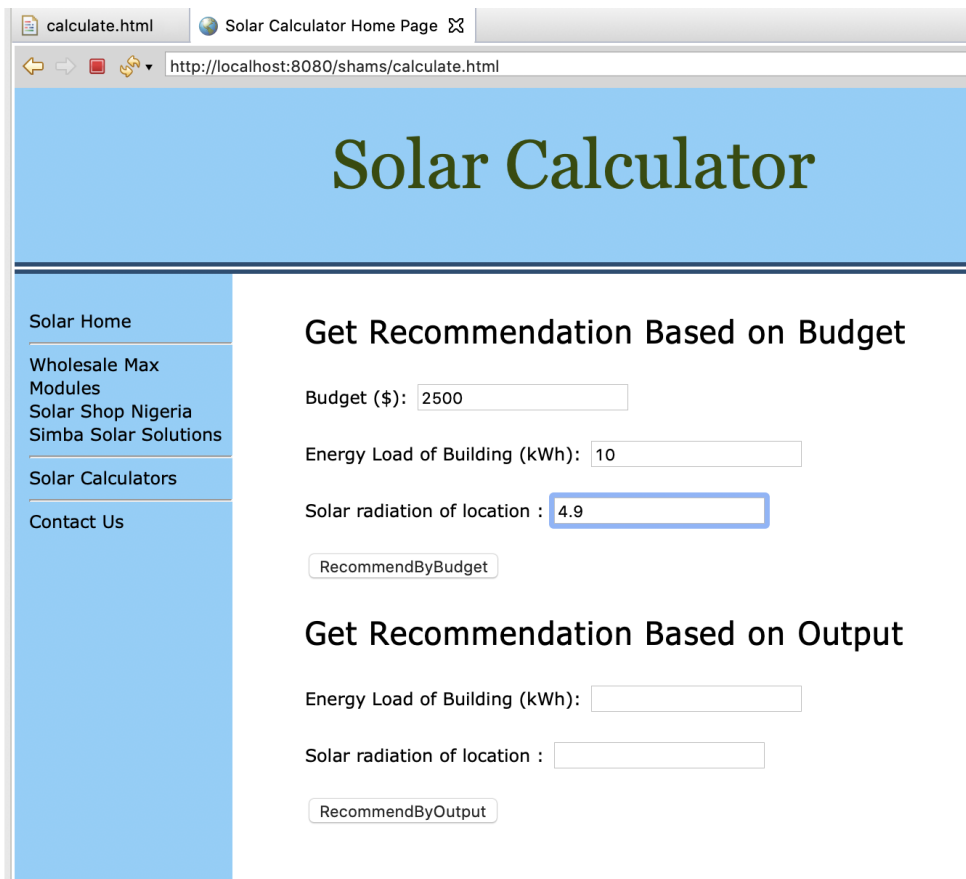


Figure 8.9: Front-End Look - User Input for Recommendation Based on Budget

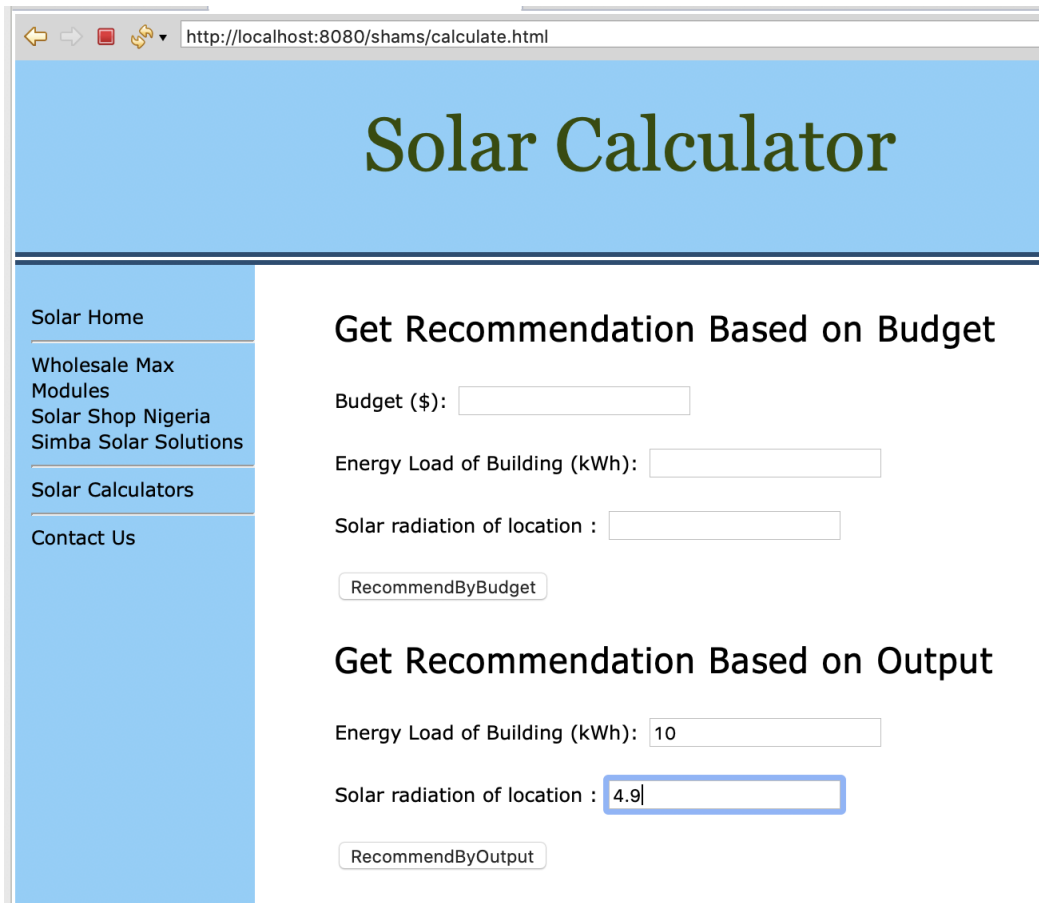


Figure 8.10: Front-End Look - User Input for Recommendation Based on Budget

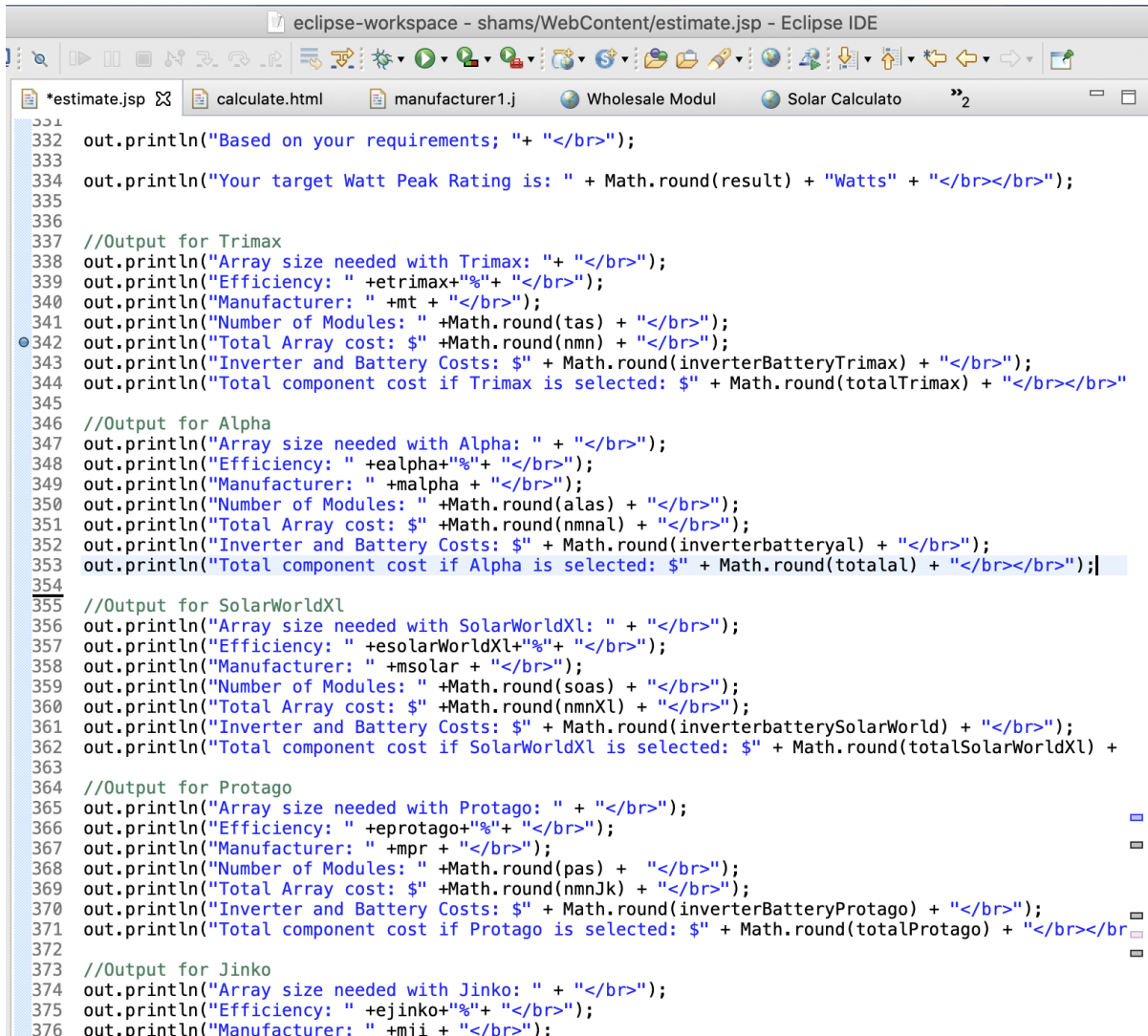
Once the user puts in their requirements, two submit buttons called *RecommendByBudget* and *RecommendByOutput* are available to click as shown in figures above. The submit button triggers a communication link to be set up (through a communication servlet within the server) between the front-end and the back end. The server accepts requests from the user and forwards it into the servlet container via a hypertext transfer protocol (HTTP) request code to execute the defined operations. If *RecommendByBudget* is selected, user requirements are submitted into *estimate.jsp*, however, if *RecommendByOutput*, then the user requirements are submitted into *output.jsp*. The computations are executed within the servlet and the server presents the user with the dynamically obtained recommendations through HTTP response status codes to the front end of the homepage. The results are then presented. The ensuing subsection shows how the web-based application displays recommended component to the user.

8.5.5. Display of Recommendations

Assuming the user inputs a budget of \$2,500, an energy load of 10kWh, and solar radiation of 4.9. Within *estimate.jsp*, the requirements the user inputs are displayed before the calculations are made. Figure 8.11 below shows the codes that prompt the system to display the user inputs. Figure 8.12 shows codes that prompt the system to display results for all components.

```
49  <h1>
50      Recommend Components Based on Budget
51  </h1>
52  Budget $: <%= request.getParameter("budget") %>
53  <br>
54  Solar radiation of location: <%= request.getParameter("solar") %>
55  <br>
56  Energy Load of Building in KiloWatts: <%= request.getParameter("energyLoad") %>
57  <br>
58  </hr>
```

Figure 8.11: Codes that Prompt User Requirements to be Displayed



```
eclipse-workspace - shams/WebContent/estimate.jsp - Eclipse IDE
*estimate.jsp calculate.html manufacturer1.j Wholesale Modul Solar Calculato
332 out.println("Based on your requirements; "+ "<br>");
333
334 out.println("Your target Watt Peak Rating is: " + Math.round(result) + "Watts" + "<br><br>");
335
336
337 //Output for Trimax
338 out.println("Array size needed with Trimax: " + "<br>");
339 out.println("Efficiency: " + etrimax+"%" + "<br>");
340 out.println("Manufacturer: " + mt + "<br>");
341 out.println("Number of Modules: " + Math.round(tas) + "<br>");
342 out.println("Total Array cost: $" + Math.round(nmn) + "<br>");
343 out.println("Inverter and Battery Costs: $" + Math.round(inverterBatteryTrimax) + "<br>");
344 out.println("Total component cost if Trimax is selected: $" + Math.round(totalTrimax) + "<br><br>");
345
346 //Output for Alpha
347 out.println("Array size needed with Alpha: " + "<br>");
348 out.println("Efficiency: " + ealpha+"%" + "<br>");
349 out.println("Manufacturer: " + malpha + "<br>");
350 out.println("Number of Modules: " + Math.round(alas) + "<br>");
351 out.println("Total Array cost: $" + Math.round(nmna) + "<br>");
352 out.println("Inverter and Battery Costs: $" + Math.round(inverterbatteryal) + "<br>");
353 out.println("Total component cost if Alpha is selected: $" + Math.round(totalal) + "<br><br>");
354
355 //Output for SolarWorldXl
356 out.println("Array size needed with SolarWorldXl: " + "<br>");
357 out.println("Efficiency: " + esolarWorldXl+"%" + "<br>");
358 out.println("Manufacturer: " + msolar + "<br>");
359 out.println("Number of Modules: " + Math.round(soas) + "<br>");
360 out.println("Total Array cost: $" + Math.round(nmnXl) + "<br>");
361 out.println("Inverter and Battery Costs: $" + Math.round(inverterbatterySolarWorld) + "<br>");
362 out.println("Total component cost if SolarWorldXl is selected: $" + Math.round(totalSolarWorldXl) +
363
364 //Output for Protago
365 out.println("Array size needed with Protago: " + "<br>");
366 out.println("Efficiency: " + eprotago+"%" + "<br>");
367 out.println("Manufacturer: " + mpr + "<br>");
368 out.println("Number of Modules: " + Math.round(pas) + "<br>");
369 out.println("Total Array cost: $" + Math.round(nmnJk) + "<br>");
370 out.println("Inverter and Battery Costs: $" + Math.round(inverterBatteryProtago) + "<br>");
371 out.println("Total component cost if Protago is selected: $" + Math.round(totalProtago) + "<br><br>");
372
373 //Output for Jinko
374 out.println("Array size needed with Jinko: " + "<br>");
375 out.println("Efficiency: " + ejinko+"%" + "<br>");
376 out.println("Manufacturer: " + mji + "<br>");
```

Figure 8.12: Codes that Prompt the System to Display Results for all Components

As earlier explained, once the user submits requirements, *estimate.jsp* is automatically run and computations within that file are executed. Once the file is executed over the server, it presents the results of the computations. First, the system displays the parameters or values earlier inputted by the user. Secondly, it states which component it recommends based on the computations. Lastly, it provides a detailed result of computations for each module type. Figures 8.13 (a) and (b) below show the display of results. Observe that the system provides the efficiency and manufacturer of each module for reference. Furthermore, array size is calculated for each. Based on the calculated costs of the array, and an estimated cost of inverters and battery, the total cost if that component is selected is determined. It is this total cost that is compared to the budget given by the user. Based on the comparison, the application recommends the most suitable component.

Solar Calculator

Recommend Components Based on Budget

Budget \$: 2500
Solar radiation of location: 4.9
Energy Load of Building in KiloWatts: 10

Based on your requirements;
Module recommended is pc5max --> Difference from Budget: \$668

Find details of all modules bellow;

Array size needed with Trimax:
Efficiency: 21.0%
Manufacturer: MAXPV
Number of Modules: 18
Total Array cost: \$3612
Inverter and Battery Costs: \$358
Total component cost if Trimax is selected: \$3970

Array size needed with Ultron:
Efficiency: 20.0%
Manufacturer: SOLARSHOP
Number of Modules: 21
Total Array cost: \$4037
Inverter and Battery Costs: \$399
Total component cost if Ultron is selected: \$4436

Figure 8.13 (a): Module Recommended Based on Budget

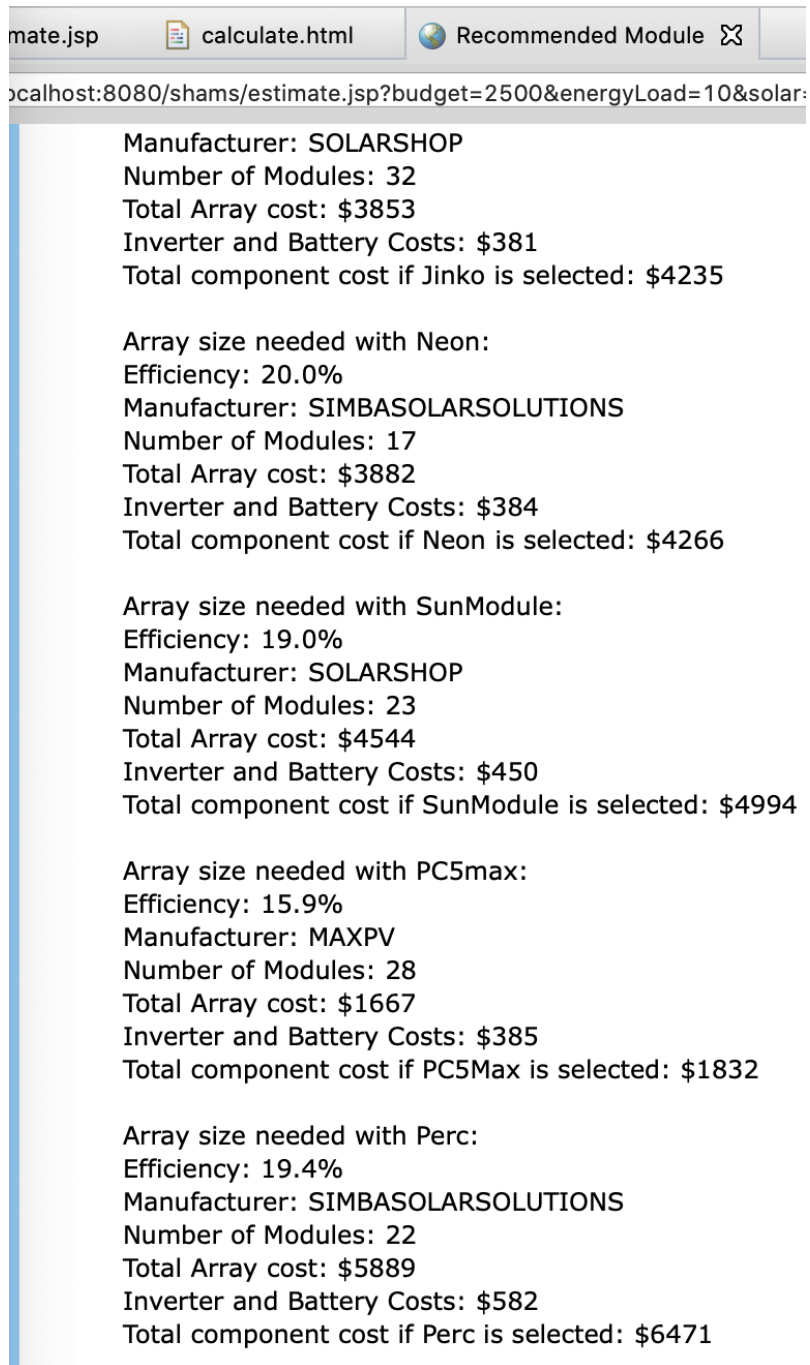


Figure 8.13 (b): Detailed Display of Results for other Modules

It can be observed from figures above that the details of calculations show that the PC5max module is closest to the budget of the user and therefore recommended for use. It is important to know that calculations are done for absolute best, which means that the recommended component could also be slightly above budget but recommended if it's closest to the budget compared to the other components. If more than one of the components are suitable for the user, or the user is not concerned about cost of

components but rather worried about the efficiency of the output of the photovoltaic system, then the users could select to get recommendations based on output from the system. On the front-end of the application, the user inputs in their building energy load and solar radiation of their location and clicks *RecommendByOutput*. This button automatically executes *output.jsp* as explained above. Using the same assumptions for an energy load of 10kWh, and solar radiation of 4.9, the result of this computation is provided in Figure 8.14 below.

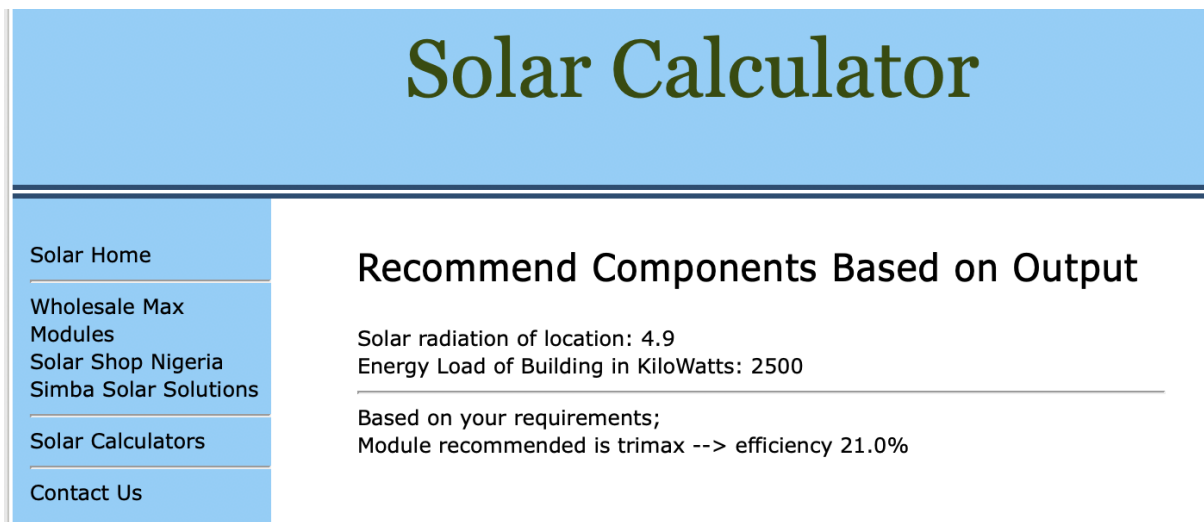


Figure 8.14: Display of Results for Recommendation Based on Output

8.5.6. Manufacturer Webpages

The front-end interface also includes manufacturer webpages for the manufacturers of Photovoltaic components used as instances in the ontology. These webpages call the components directly from *PvSemanticWeb.java* where they have been read from the ontology. These components have a structured representation in RDF whereby they cannot be modified like it is done in HTML representation. The only way the components can be modified is if they are amended within the *ifcOWL-pvOntology* by the manufacturers. This system allows manufacturers to represent their products in semantic language.

The three manufacturers are Wholesale Max Module, Solar Shop and Simba solutions represented in manufacturer1.jsp, manufacturer2.jsp, and manufacturer3.jsp respectively. Figures 8.13, 8.14 and 8.15 show the components manufacturer webpages.

The screenshot shows a web browser window displaying the 'Wholesale Max Modules' website. The page has a blue header with the title 'Wholesale Max Modules' and the tagline 'Sustainable, Affordable, Profitable...'. A navigation menu on the left includes links for 'Solar Home', 'Solar Modules', 'Solar Inverters', 'Solar Batteries', 'Charge Controllers', 'Solar Calculators', and 'Contact Us'. The main content area is titled 'Solar Modules' and contains a table with four columns representing different solar module types: AllmaxModule, Trimax, Protago, and PC5max. Each column includes an image of the module and a table of specifications.


	AllmaxModule	Trimax	Protago	PC5max
Module Type	AMORPHOUSMODULE	MONOCRYSTALLINEMODULE	MONOCRYSTALLINEMODULE	POLYCRYSTALLINEMODULE
Rated Power	225W	410W	315W	260W
Cell Orientation	72cells	60cells	60cells	72cells
Component Area	2535	1520	1650	3042
Weight	3.2kg	2kg	2.4kg	3.8kg
Efficiency	17.5%	25.3%	20.9%	15.9%

8.13: Wholesale Max Modules (manufacturer1.jsp)

eclipse-workspace - http://localhost:8080/shams/manufacture2.jsp - Eclipse IDE

estimate.jsp calculate.html Solar Calculato manufacturer1.j Wholesale Modul manufacture2.j SolarShopNigeri

http://localhost:8080/shams/manufacture2.jsp



Home PV Modules PV Inverters Solar Calculators Contact Us

Solar Modules





				
	Ultron	Jinko	Sunmodule	SolarWorldXL
Module Type	MONOCRYSTALLINEMODULE	POLYCRYSTALLINEMODULE	AMORPHOUSMODULE	MONOCRYSTALLINEMODULE
Rated Power	340W	225W	310W	380W
Cell Orientation	60cells	72cells	72cells	60cells
Component Area	2050	3200	2650	1750
Weight	2kg	2.9kg	2.4kg	2.2kg
Efficiency	20%	17%	19%	26.7%

Figure 8.14: Solar Shop (manufacturer2.jsp)

eclipse-workspace - http://localhost:8080/shams/manufacture3.jsp - Eclipse IDE

calculate.html Solar Calculato manufacturer3.j Wholesale Modul manufacture2.j SolarShopNigeri Simba Solar Sol

http://localhost:8080/shams/manufacture3.jsp

Simba Solar Solutions

Home PV Modules PV Inverters Solar Calculators Contact Us Simba Locations Promotions

				
	Neon	Perc	JaSolar	Alpha
Module Type	MONOCRYSTALLINEMODULE	MONOCRYSTALLINEMODULE	POLYCRYSTALLINEMODULE	MONOCRYSTALLINEMODULE
Rated Power	415W	330W	290W	310W
Cell Orientation	60cells	60cells	60cells	60cells
Component Area	2000	1950	2800	2050
Weight	1.9kg	2.4kg	2.6kg	18.9%
Efficiency	27%	19.4%	17.4%	18.9%
Costs	\$223	\$269	\$130	\$172
Warranty Period	10years	10years	12years	12years

Figure 8.15: Simba Solar Solutions (manufacturer3.jsp)

This web-based application runs through the products available and supplied by these manufacturers and returns a recommendation. The codes are available at the Github site for this study - <https://github.com/zainabusman/>.

8.6. Conclusion

This chapter discussed the development of a web-based application that recommends suitable photovoltaic technology components to the user based on the knowledge obtained from ifcOWL-pvOntology. Details of required software requirements and specifications and software components and tools were given. The system architecture presented showed how the software tools were applied throughout the chapter.

SIPS was developed in the eclipse development environment using java and JSP languages, Jena API and tomcat server. The IfcOWL_pvOntologyMaxPv, IfcOWL_pvOntologySimbaSolutions, IfcOWL_pvOntologySolarShop were successfully read into eclipse and knowledge extracted were stored in HashMaps for further processing. The .jsp files contained algorithms used to size photovoltaic technology components. A user interface was developed that allows the user to input requirements. This input is then submitted to the backend file and executed within the servlet container. The server returns the recommended component to the user and presents the values on the front-end interface. Manufacturer webpages were dynamically developed, and their products are represented in RDF as extracted from the ontology.

This chapter successfully reads in knowledge from the IfcOWL_pvOntology, creates a connection with a user interface and uses knowledge extracted from the ontology to recommend suitable photovoltaic technology components to the user. However, it is important to prove whether the prototype system can be fully utilised to demonstrate the purpose for its development. In the next chapter, we use semantic and syntactic evaluation to ensure ontology developed is efficient and free of anomalies. Furthermore, a case study will be conducted to validate the prototype system.

9. PROTOTYPE EVALUATION

9.1. Introduction

Ontology development is hard and time consuming despite available and useful modelling methodologies. Obtaining errors and inconsistencies are often inevitable, therefore it is imperative to conduct thorough evaluation on the quality and correctness of the developed ontology. Relatedly, an ontology is evaluated to ensure it is built in compliance with its modelling requirements. Several approaches to ontology evaluation that exist in literature have been reviewed in chapter 3. After a thorough appraisal, it has been observed that the most common studies in ontology evaluation concentrate on only two main evaluation metrics; to check for ontology quality and ontology correctness through semantic and syntactic verification and validation. Some key discussions on semantic verification and validation can be found in Noy and McGuinness (2001), Gomez-Perez (2001), Noy and Musen (2002), Baumeister and Seipel, (2005), Antoniou et al., (2012), Hlomani & Stacey, (2014), Bandeira, et al., (2017), Sun et al., (2020), Fernández-Izquierdo and García-Castro, (2021).

Therefore, in this chapter, an evaluation is done on the IfcOWL_pvOntology and the web-based application to check whether the overall prototype system meets the objectives set out in the requirement analysis stage. The evaluation approaches adopted include semantic and syntactic verification where ontology reasoners will be used for verification of the knowledge base. Inconsistencies and anomalies within the ontology will be checked. It is equally important to evaluate the quality of an ontology, therefore syntactic quality, semantic quality, pragmatic quality, and social quality checks are done. A statistical framework based on criteria-based and goal-oriented evaluation was used to test the quality of the developed ontology. It is also pragmatic to ask whether a correct ontology means it has been developed in a qualitative approach? Or if having a good quality ontology means it is correct? Since the ontology is an approximate specification of a domain, its evaluation should investigate how close to the real-world the knowledge model is (Fernández-Izquierdo and García-Castro, 2021), to ensure that the correct model of the domain is created. This chapter also includes an instance

evaluation which is done to ensure all instances within the knowledge model are syntactically and semantically valid.

Furthermore, the use of case studies is adopted to validate the prototype system. New knowledge will be added to check efficiency of the system. It is important to involve domain experts and users in the evaluation of *IfcOWL_pvOntology*, to retrieve valuable feedback. Responses and criticisms can help researchers to identify misunderstandings in requirements specifications, and improve the correctness and quality of the knowledge captured (Fernández-Izquierdo and García-Castro, 2021). The main objective of this chapter is to ensure that the *ifcOWL-pvOntology*, efficiently represents domain knowledge on photovoltaic systems technology using IFC structure, it is syntactically correct and qualitative, and to ensure that knowledge about PV-systems can be easily accessed from *ifcOWL-pvOntology* and used effectively within the web-based application for component sizing and recommendation. Screenshots were used where applicable for easy comprehension. At the end of evaluation, the prototype system should be both syntactically and semantically fit for use in a practice.

9.2. Semantic and Syntactic Verification of Ontology

In chapter 6, the PROMPT methodology was adopted for alignment of established concepts. Using PROMPT ruled out inconsistencies and inaccuracies. The knowledge refinement done guaranteed the semantic verification of the knowledge model. However, in chapter 7, the implementation of the ontology was done in a software editing environment therefore syntactic and semantic errors are very likely to occur. Examples of syntactic error may be incompatibility of a datatype and its values while semantic errors are often related to inaccuracies with relationships of classes and their hierarchies. Therefore, it is important to conduct further semantic and syntactic verification of the *ifcOWL-pvOntology* knowledge model implemented in chapter 7. Eliminating anomalies are done using ontology-reasoning tools such as FaCT++, Pellet incorporated in protégé 4.0.2. Verification is an iterative process until all errors are removed.

9.2.1. Semantic and Syntactic Anomalies

Following the CommonKADS methodology adopted in this study, the last stage in ontology knowledge development is refining the knowledge to ensure it is free of anomalies. Different stages in ontology development such as merging or extending an already existing ontology (as is the case here with ifcOWL ontology) lead to anomalies (Baumeister and Seipel, 2005). The most common types of design anomalies and taxonomic errors that must be checked for are inconsistencies, incompleteness, and redundancies (Gomez-Perez, 1999).

Inconsistency errors occur when a contradictory definition or axiom is found in the knowledge base. There are three types of anomalies that are deliberated as inconsistency errors. These are circularity, partition, and semantic errors. A *circularity* error occurs when a class is defined in any level of hierarchy within an ontology as a specialisation or generalisation of itself (Baumeister and Seipel, 2005). For instance, from the conceptual model presented in Figure 6.7; if a *SolarDeviceInstallation* is defined as a subclass of *GridConnection*, a circularity error will occur since *GridConnection* is already a subclass of *SolarDeviceInstallation*. Generally, subconcepts of a class can be defined as a disjoint partition of that class. A *partition* error occurs when a class is defined as a common subclass of several other classes with specified disjoint axioms (Gomez-Perez, 1999). This means that a single instance belonging to two disjoint classes will inevitably result in a partition error. For instance, while modelling the photovoltaic system in Figure 6.7, an instance of a *Battery* cannot be modelled as an instance of a *SolarModule* because their super classes (*ElectricalPhotovoltaicComponent* and *NonElectricalPhotovoltaicComponent*) are disjoint classes where disjoint axioms are appropriately specified. Any scenario where a common instance is forced to be an instance of two disjoint classes, a partition error will occur. *Semantic* inconsistency occurs when the ontology developer asserts incorrect semantic classification such that a class is defined as a subclass of another class that it does not belong to, thereby there is no semantic relation between the concepts (Baumeister and Seipel, 2005). For instance, defining the class *IfcComponentManufacturer* as a subclass of *Location*.

Ontological knowledge base can be *incomplete* if relationships between concepts are inaccurately defined. *Incompleteness* errors also occur when there are missing definitions and important concepts of the domain are omitted from the taxonomy. Similarly, imprecise definition of partitions also causes incompleteness (Baumeister and Seipel, 2005). Incompleteness errors make the ontology an unambiguous representation of domain knowledge and hinders ontology reasoners from reasoning effectively. *Redundancy* errors are detected when classes, subclasses, instances, or properties are inferred more than once in the ontology. Any concept or knowledge in an ontology that can be removed from the knowledge base, and such action will not affect the semantics of the ontology, that concept is a redundant concept.

Aside from the taxonomy anomalies described above, some design anomalies exist that are subtle. An example of such an anomaly is lazy concepts that are defined in the ontology but never used. Such concepts are often found in large ontologies as is the case with IfcOWL_pvOntology. For instance, *IfcRoof* is a lazy concept in this implementation. However, it is preserved in the ontology for reusability and standardisation purposes. Such anomalies do not necessarily lead to errors in reasoning. The ontology developer simply forfeits simplicity of the taxonomy for reusability.

9.2.2. Checking and Clearing Inconsistency, Incompleteness and Redundancy

Errors

Internal consistency checks were carried out on IfcOWL_pvOntology to check for inconsistencies, incompleteness, and redundancies to ensure the content of the ontology are represented in logical sequence, are consistent and satisfactory. To get accurate results, we adopted protégé-OWL plugins such as FaCT++, hermit and pellet to check for all anomalies. These reasoners were executed interchangeably through every single stage in the development process of the ontology so that errors that occurred were resolved instantly. During the development of ifcOWL-pvOntology in protégé-OWL 5.5.0, each time FaCT++ 1.6.5 hermit 1.4.3.456 and pellet 2.3.1 were run, the results were displayed in a pop-up window as seen in figure 9.1 below. The pop-up window shows options log file, preferences, time stamp and to clear the log. In cases where errors are detected, inference is shown in red. In most cases, the ontology developer must traverse

the ontology to rectify the aspects of the ontology resulting in errors, however, in some cases, the protégé-OWL is able to correct some errors automatically. FaCT++, hermit and Pellet were used incrementally and consistently during the development of IfcOWL_pvOntology until the ontology was free of errors. Reasoning results show no semantic errors and that means the ontology is complete and semantically and syntactically verified. The verification of IfcOWL_pvOntology is presented in Figure 9.1 below.

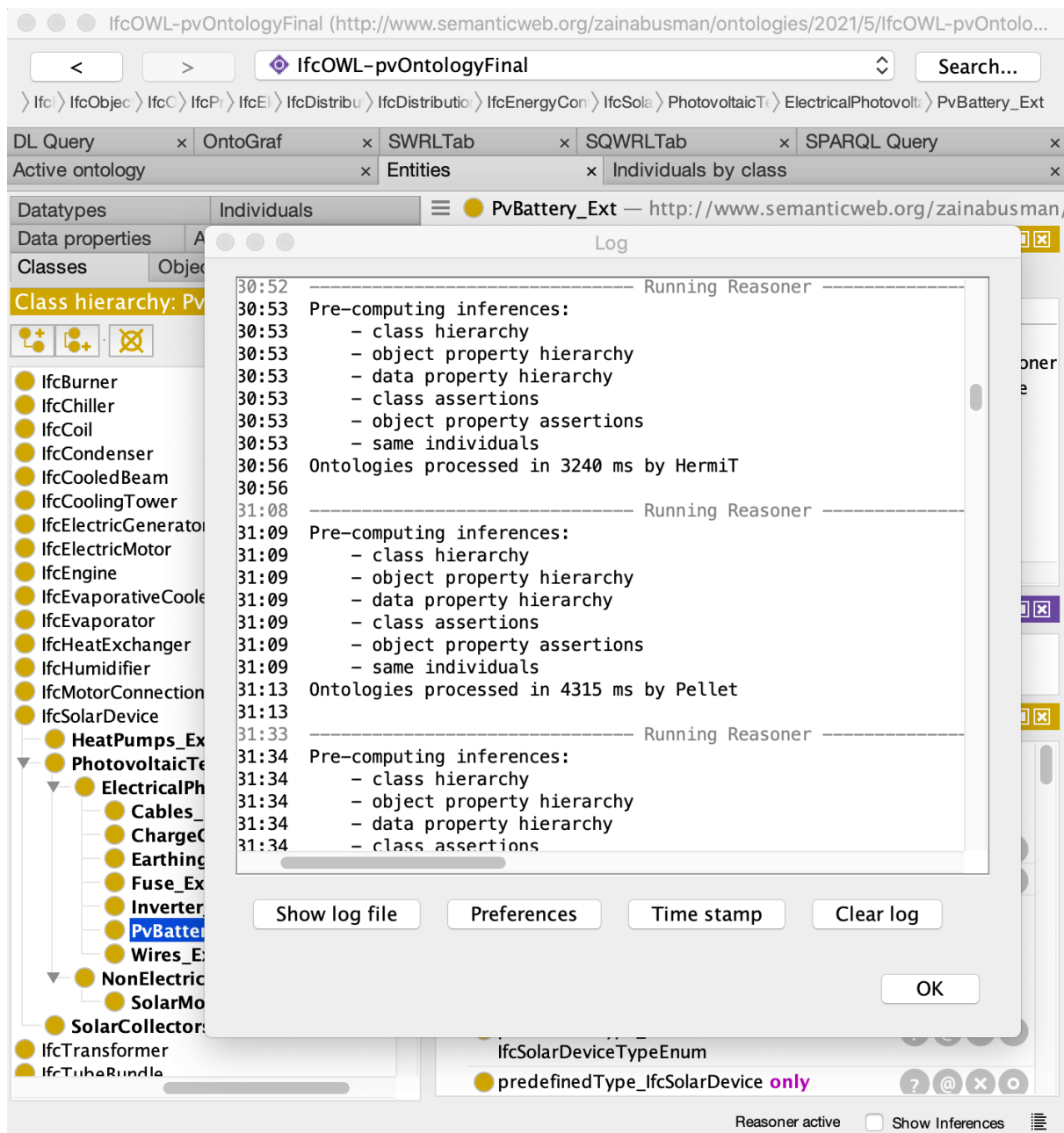


Figure 9.1: Checking of Anomalies of IfcOWL_pvOntology in Protégé-OWL

It is usual practice that once an ontology is fully developed and is semantically and syntactically verified, new Knowledge is added to test the correctness of the ontology. This is known as the probe test (Horridge et al. 2007). The probe test changes the ontology by introducing new classes (all called probe classes), then test the assertions and eventually revert the changes after the test is complete (Warrender & Lord, 2015). In the context of the *IfcOWL_pvOntology*, we conduct probe tests by intentionally adding faults to the ontology and observe how it affects the knowledge base. This is specifically important for the domain of photovoltaic system technology. Recall in Chapter 6, knowledge acquisition for the photovoltaic systems domain showed that some established concepts were disjoint to each other while others were not. For example, the subclasses of *PhotovoltaicTechnology* which are *ElectricalPhotovoltaicComponent* and *NonElectricalPhotovoltaicComponent* are disjoint classes because they cannot share instances. An instance of a battery cannot be an instance of a solar module. Furthermore, naturally, classes within an ontology overlap each other until they have been clearly specified as disjoint. Therefore, it is imperative that developed ontology has appropriate definition of disjoint axioms where they are required. If certain classes are not defined as disjoint from each other, errors will occur. Similarly, if some classes are inappropriately specified as disjoint from each other, then errors will occur until specifications have been corrected.

To test the correctness of disjoint axiom specifications in the ontology for the superclass *PhotovoltaicTechnology*, we introduce a probe class named *ProbeTestClass* as subclass of both *ElectricalPhotovoltaicComponent* and *NonElectricalPhotovoltaicComponent*. It can be observed in Figure 9.2 below that an error message occurred preventing the introduction of a subclass that is similar to the subclass *ProbeTestClass* that has already been created under *ElectricalPhotovoltaicComponent*. Therefore, we take a different approach to test the system.

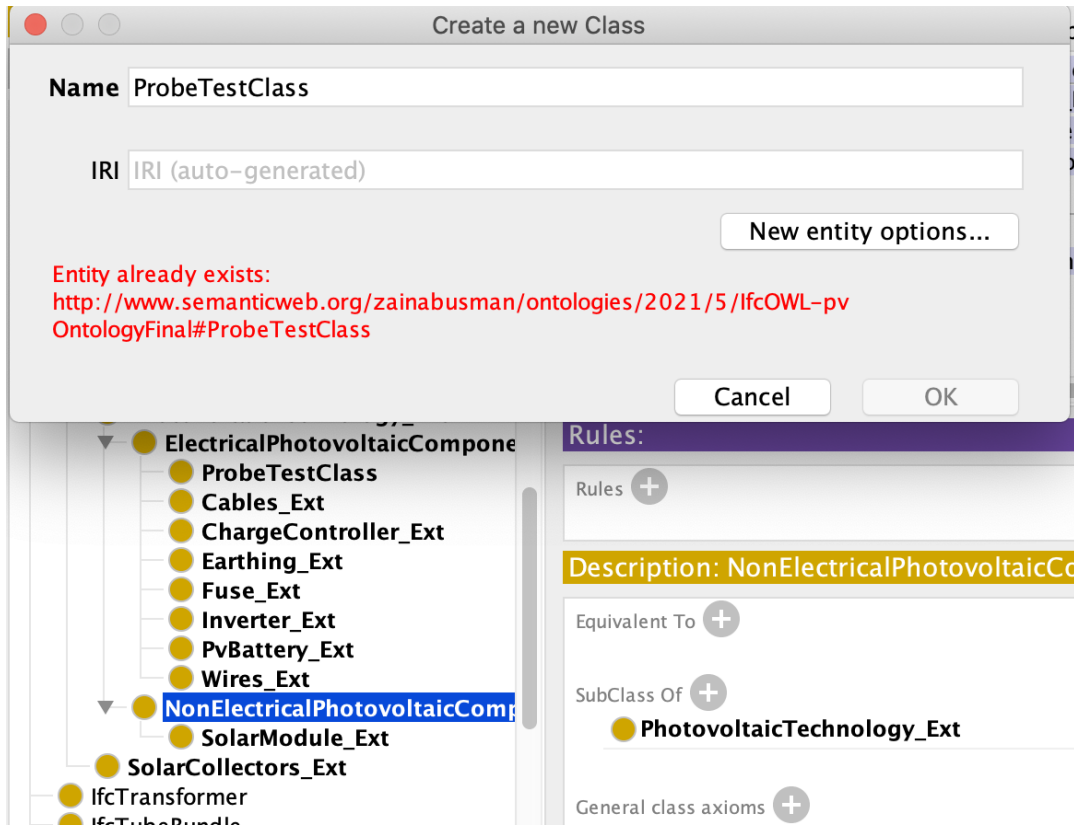


Figure 9.2: ProbeTestClass Error Message

To further test the system, an individual named *probeTestInstance* is created and forced to be an instance of both *ElectricalPhotovoltaicComponent* and *NonElectricalPhotovoltaicComponent*. Then, we run the Pellet reasoner as described above to observe how these changes affect the knowledge base. An inconsistency partition error message came up as seen in Figure 9.3 below. Figure 9.4 below shows inconsistency errors captured by the reasoner. This error message confirms that the appropriate specification for disjoint axioms was made for *ElectricalPhotovoltaicComponent* and *NonElectricalPhotovoltaicComponent*. If the error message had not occurred, the developer would have had to go back to the ontology to ensure the appropriate disjoint axiom specifications were made. Similarly, the probe test would have to be re-run until the test was free of anomalies. The probe test instance can then be removed.

Hybrid systems share some instances. Similarly, *NonGridConnection* and *GridConnection* also share certain instances such as instances of *SolarModule*, *PvBattery*, *Inverters* and *ChargeControllers*. This means that these classes should not be modelled as disjoint axioms. To test whether the appropriate disjoint specifications are made, a new individual named *ProbeTestInstance2* is introduced and forced to be an instance of both *StandAlone* and *Hybrid*. The Pellet reasoner is then executed. No error messages occurred as seen in Figure 9.5 below, and this confirms that disjoint axioms were not specified between the classes. If an error message has occurred, it means disjoint axioms must have been specified and thereby must be removed. The *probeTestInstance2* is then removed. The reasoner is executed until there are no errors detected. After the semantic and syntactic verification of the ontology is complete, further checks are conducted (in the next subsection) to ensure developed ontology is correct.

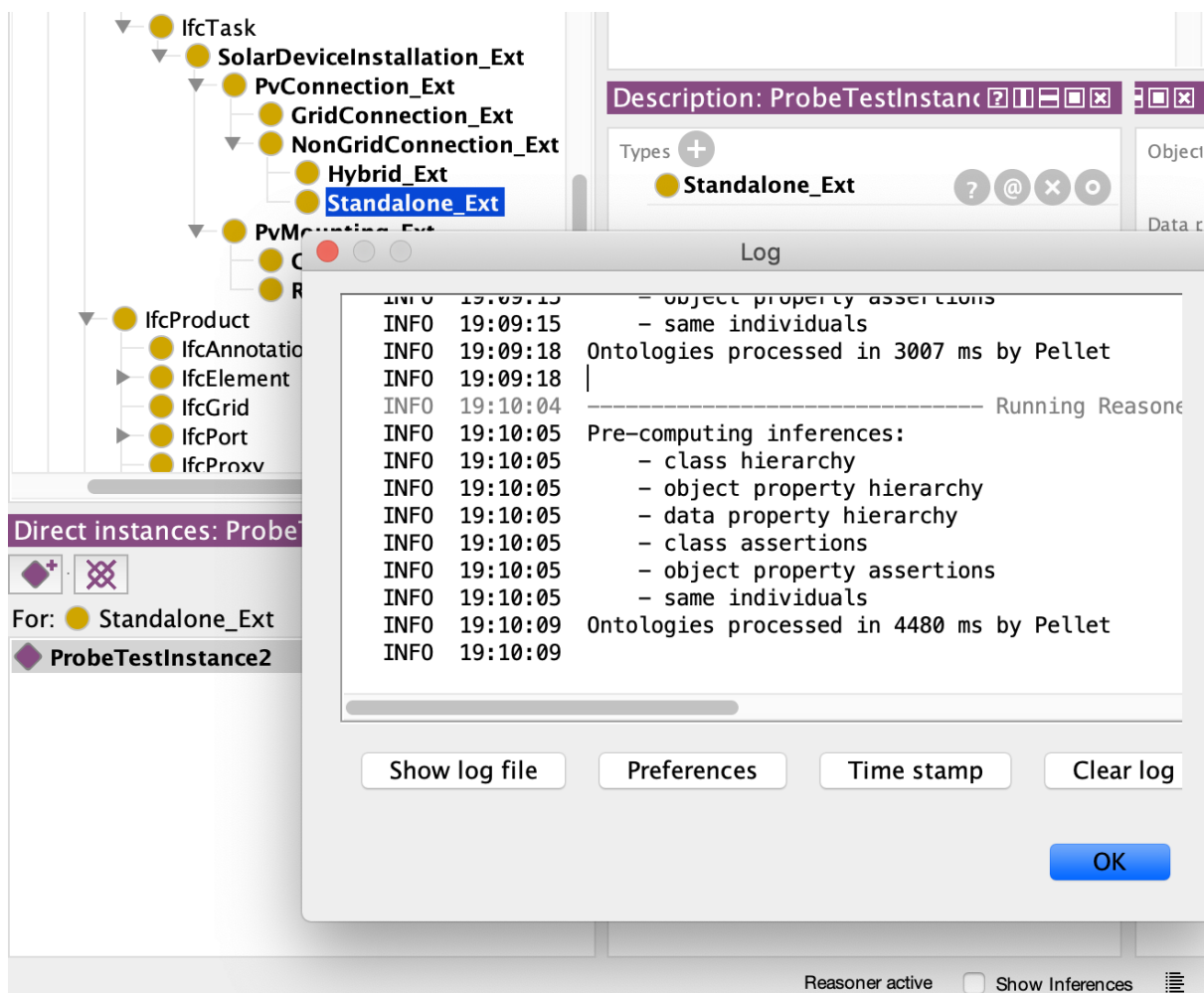


Figure 9.5: Inference After Executing Pellet During Probe Test 2

9.3. Quality Check Evaluation

Most research done on ontology evaluation prioritises evaluation done based on quality of the developed knowledge model (Sun et al., 2020). Key research questions require appraised answers. Elimination of inconsistencies, incompleteness and redundancy anomalies ensures developed ontology is correct. However, does a correct ontology mean it has been developed in a qualitative approach? To answer this question, a statistical framework based on criteria-based evaluation by Bandeira, et al., (2017) and Sun, et al., (2020) is adopted and modified to evaluate the quality of IfcOWL_pvOntology. Table 9.1 below presents a set of restructured questions that follow the GQM (goal, question, metric), a goal-oriented approach to software quality evaluation. Each question serves a goal, and each answer fulfils an ontology quality criterion. Observe that metrics evaluated here (completeness, adaptability, consistency, conciseness, computational efficiency, and clarity) are consistent with evaluation matrices thoroughly appraised in popular ontology evaluation studies (Noy and McGuinness (2001); Gomez-Perez (2001); Noy and Musen (2002); Bandeira, et al., (2017) and Sun et al., (2020). As seen in the table below, a metric and grade ranging 0-100 is given for each question answered in the context of the IfcOWL_pvOntology so that a mean value can be calculated for each goal.

Table 9.1: Applying GQM approach on IfcOWL_pvOntology

Goal	Questions	Metric	Grade	Average
1. Ontology should comply with substitute	Q1. Were competency questions and objectives of the ontology defined and answered?	Completeness	100	100
	Q2. Were existing ontologies reused?	Adaptability	100	
2. Does ontology comply with ontological commitments?	Q3. Is a maximum ontological commitment imposed?	Conciseness	75	87.5
	Q4. Are the properties of the ontology consistent with the domain?	Consistency	100	
3. Does the ontology comply with intelligent reasoning	Q5. Are there incomplete axioms?	Completeness	100	100
	Q6. Are there contradictory axioms?	Consistency	100	
	Q7. Are there redundant axioms?	Conciseness	100	
4. Does the ontology allow efficient computations	Q8. Do ontology reasoners run successfully and identify errors?	Computational Efficiency	100	100
	Q9. Do ontology reasoners run speedily?			

5. Does ontology comply with human communication?	Q10. Were ontology questions well defined?	Clarity	80	60
	Q11. Does ontology contain thorough annotations that describe concepts?	Clarity	40	

The equation adopted (is criteria-based and a beta regression model) for checking ontology quality is as follows:

$$\mu_i = \frac{\exp\{-0.44+0.03(\alpha_{g1} \times G_1)+0.02(\alpha_{g2} \times G_2)+0.01(\alpha_{g3} \times G_3)+0.02(\alpha_{g4} \times G_4)-0.66Exp^{i-25(0.1 \times NE)^i}\}}{1+\exp\{-0.44+0.03(\alpha_{g1} \times G_1)+0.02(\alpha_{g2} \times G_2)+0.01(\alpha_{g3} \times G_3)+0.02(\alpha_{g4} \times G_4)-0.66Exp^{i-25(0.1 \times NE)^i}\}} \dots (1)$$

To calculate the quality of the ontology, the average value is obtained for the questions within each goal. $\alpha_{g1}, \alpha_{g2}, \alpha_{g3}, \alpha_{g4}, \alpha_{g5}$ denotes the average values for goals 1, 2, 3, 4, and 5 respectively. Exp denotes the experience of the evaluator. This value is 1 if the evaluator has a vast knowledge and experience in ontology modelling, otherwise the value is 0. NE is a variable whose value is 1 if the evaluator can answer all questions. Variables G_1, G_2, G_3, G_4 , denote constant values for each goal used depending on roles required. These roles are explained in further details below. Once the statistical evaluation is done for each question, the quality of the ontology is calculated and checked against a scale of 0, 1. The quality of the ontology can be evaluated either partially or completely. Partial evaluation considers one goal at a time. For instance, to evaluate for completeness and adaptability (goal 1), α_{g1} would be 100 (average value for goal1); G_1 would be 1 because only goal one is being considered by the evaluator; $\alpha_{g2} = 0, \alpha_{g3} = 0, \alpha_{g4} = 0, \alpha_{g5} = 0, G_2 = 0, G_3 = 0, G_4 = 0$, because goals 2, 3, 4, and 5 are not being considered.

To evaluate the quality of IfcOWL_pvOntology, we perform a complete quality evaluation. That means values are given for all variables for all goals. The values are substituted into equation (1) as shown below in equation (2) where $\alpha_{g1} = 100, \alpha_{g2} = 87.5, \alpha_{g3} = 100, \alpha_{g4} = 100, \alpha_{g5} = 60, G_1 = 1, G_2 = 1, G_3 = 1, G_4 = 1, Exp = 1, and NE = 1$.

$$\mu_{IfcOWL_pvOntology} = \frac{\exp\{-0.44+0.03(100 \times 1)+0.02(87.5 \times 1)+0.01(100 \times 1)+0.02(100 \times 1)-0.66(1)^{i-25(0.1 \times 1)^i}\}}{1+\exp\{-0.44+0.03(100 \times 1)+0.02(87.5 \times 1)+0.01(100 \times 1)+0.02(100 \times 1)-0.66(1)^{i-25(0.1 \times 1)^i}\}} \dots (2)$$

$$\mu_{IfcOWL_pvOntology} = 0.9891$$

The result of the total quality of the ontology is 0.9891. Since this value is nearest to 1, it shows the quality of ifcOWL-pvOntology is high. Thus, the developed ontology is successfully evaluated in the context of its quality.

9.4. Instance Data Evaluation

Several observations on online publications of ontologies show that instance data are typically published separately in RDF documents by users and these documents are published after their corresponding ontologies have been published (Tao, Ding and McGuinness, 2009). Similarly, it is observed that a large percentage of published RDF are only instance data. The argument from these observations is whether representation of structure of classes and properties are less needed online than the instance statements. This argument signifies instance data is significant in ontology modelling (Qin and Atluri, 2009). Most ontologies developed are meticulously evaluated to eliminate errors, therefore, since the semantic and syntactic structures of ontology concepts and properties are like their instances, this indicates that instance data are also prone to contain semantic errors. Therefore, intuitively, it is imperative that an instance data evaluation is conducted to check the quality of instances defined within the ifcOWL-pvOntology.

Instance evaluation is very common to general ontology evaluation because both methods inspect the syntax and semantic problems related to the knowledge being evaluated. However, instance data evaluation focuses more on anomalies caused by the instances thereby adopting approaches that are more prominent in instance data evaluation such as syntax errors, logical inconsistencies and potential issues (Tao, Ding and McGuinness, 2009). While ontology evaluation focuses on consistency and completeness of data, instance evaluation focuses on gauging whether the instance data explicitly and implicitly meets the expectations of the corresponding ontology. For instance, how well does the trimax module fit as an instance of SolarModule? Therefore, we define and evaluate the structure and semantics of the instances within one of the manufacturer ontologies; OWL-pvOntologyMaxPv.

A data instance denoted as i is a valid instance of an ontology o , if it meets certain conditions. $i \in I(c)$, is valid if the concept it belongs to (c) is a well-defined and valid concept in the ontology o ; if each property instantiated by i has a formal specification belonging to c ; and if each property instantiated by i provides a value within its range and restrictions. If these conditions are met, then an instance i can be said to be structurally valid ($structValid(i,o) = true$). The structural validity of every instance in IfcOWL_pvOntology can be checked against these conditions. It is important to note that structural validity does not guarantee semantic validity of an instance. However, for an instance to be valid semantically, it must be structurally valid. Therefore, an instance $i \in I(c)$, is semantically valid if it is structurally valid ($structValid(i,o) = true$) and it belongs to a class c that is semantically effective or contains no semantic errors ($c \in C(o)$). Based on this algorithm, the instances within this ontology are traversed manually to ensure each is syntactically and semantically valid.

At this stage, the semantic and syntactic verification of the IfcOWL_pvOntology is complete. It is imperative to validate the knowledge model with real data from a domain case study to determine whether the ontology executes the functions it has been developed for. The details are provided in the ensuing section.

9.5. Validation Using a Case Study

Validation ensures that developed ontology models precisely what it is created for. In chapter 7, it was established that the IfcOWL_pvOntology was designed to represent domain knowledge on photovoltaic systems technology. This ontology becomes the foundation for development of a web-based application that designs and recommends photovoltaic systems components. Validation of the web-based application checks if the prototype developed meets the system requirements. Two main criteria are considered during the process of validation through case study.

Firstly, an *application or task-based evaluation* is considered. This measures how effective the ifcOWL-pvOntology is in the context of the application developed. To test this, all components of the web-based application will be kept constant, while slight modifications are done to the ontology-dependent parts. This will confirm the

importance and effect of the ontology on the performance of the application. Secondly, a *data-driven evaluation* approach is applied by comparing the ifcOWL-pvOntology and SIPS against existing data on photovoltaic systems. This also checks the coverage or how wide the ontology and system cover the domain of discourse. It measures precision. Successful evaluation provides conformity on how well the system fits the domain, coverage of concepts and coverage of the relationships between them. It also checks whether the system follows the natural workflow or process model of decisions made in the domain. A common mistake is to implicitly consider domain knowledge as constant. This is inconsistent with the natural characteristic of domain knowledge and certainly inconsistent with reality. The reality is that domain knowledge is dynamic and subject to continuous change. However, since the ontology is IFC-based and uses standardised structure for representation, new knowledge simply fits into the established structure. Any change in the ontology will be due to changes in domain knowledge. Therefore, data-driven evaluation should be directed towards data modification or addition, considering domain knowledge will most likely change or be updated in the future, the application should be able to handle that modification. For this reason, this validation will include adding new knowledge (that has been abstracted from a case study) into the ontology. The ensuing section provides a description of the case study used to validate this system.

9.5.1. Description of Case Study

To validate this ontology, a hub of high efficiency at Oxford Brookes University has been selected. Oxford Brookes University campus buildings are a classic paradigm that use photovoltaic systems technology for their electricity needs. The university has reduced over 35% of its energy usage and carbon emissions, making it one of the UK's top tier universities leading the way to support sustainability (Oxford Brookes University, 2020; Jojusolar, 2021). This case study is selected because the project contains detailed documentation suitable for knowledge abstraction into the IfcOWL_pvOntology.

The Oxford Brookes University solar journey started as early as 2005 when a target of 34% reduction in carbon emissions was set to be met by 2025. The university is already ahead of its target aimed to create a highly sustainable student campus. Their

photovoltaic system was designed by Joju Solar, a highly accredited UK solar installation company founded in 2006 by Oxford University lecturer on renewable energy Dr Chris Jardine and by entrepreneur Joe Michaels (Jojusolar, 2021). Joju solar fits well in the study because it is brand agnostic. It designs photovoltaic systems without being tied to specific manufacturers or products, thereby providing a suitable solution to the user based on the user requirement and budget (just as the prototype system we have developed works). High efficiency maxeon modules were selected for this project and installed on campus buildings such as the Buckley Building, John Payne Building, Lloyd Building, Sinclair Building. Maxeon modules are manufactured by SunPower, a popular UK photovoltaic technology manufacturing company known to produce modules with the highest efficiency in the UK and generate more electricity per square metre of roof space (SunPower, 2021; Ingrams, 2021). However, they are more expensive than other modules, costing an average of \$354.



Figure 9.6: SunPower Modules Installed on an Oxford Brookes University Building
[source: (Oxford Brookes University, 2020)]

The solar irradiance value recorded for Oxford is 3.8 kW/m^2 . Oxford Brookes' latest solar project (installed in November 2020) added about 300kWp doubling the solar capacity of the school. The selection of the modules were done by experts at JojuSolar. They used 700 maxeon modules, assembled and padded together as seen in figure 9.6

above. This resulted in an array of panels that generates 224,912kWh of electricity and reduces 57 tonnes of CO₂ (Oxford Brookes University, 2020; Jojusolar, 2021). Design considerations followed in this project is that the system is installed as part of the building; technically and architecturally. Additionally, the system does not generate the entire electric energy output the buildings need but covers desired energy offsets. For that reason, the buildings use up the entire electric energy generated by the solar panels during both summer and winter periods. Furthermore, it works like a typical grid-connected model where the grid provides the remaining energy needs of the buildings.

The design and selection of PV components entails having requirements of the building to be specified. The design of the overall PV system means sizing of PV components, most importantly the modules so that they can generate the required electric output or offset. Each component can be sized detachedly, however it is vital to size the module, then subsequently the other components. Given that PV technology components are manufactured and supplied by different companies, most often, PV system designs result in component sizes that are different from sizes available by manufacturers. For instance, a design of a PV system may lead to a selection of 10 modules of 349W. However, there may not be modules of 349W available in the market but 350W, 355W e.t.c. This complicates the process of PV design and selections.

Recall that the conceptual knowledge model (developed in UML) presented in chapter 6 was transformed into ontological concepts and implemented into the ontology chapter 7. Similarly, it is important to evaluate how the knowledge or concepts within the case study are abstracted and input into the ifcOWL-pvOntology and subsequently validated in the web-based application. Three main sources were used. Firstly, a case study description given by the installation company used by Oxford Brookes University (Jojusolar, 2021). Secondly, the PV technology manufacturer website (SunPower, 2021) was explored and thirdly, Oxford Brookes University (Oxford Brookes University, 2020) webpage detailing their sustainable energy projects was also explored.

During abstraction of knowledge from these sources, a key challenge was confronted. There were differences in use of terms where PV panel is often used to mean a module. In this study, a choice was made to focus on modules instead of panels for two main reasons. Firstly, literature sources and manufacturer websites provide details (including efficiencies) of single PV modules, rather than efficiency of panels. The properties of every single module are unique. Secondly, recall that a PV panel is made up of a group of modules assembled using external frames. The choice of these frames is entirely up to the manufacturers, and this often affects the overall efficiency of the PV-system. Furthermore, the property of the panel and subsequently the array depends on the manner by which the modules are assembled. All discussions henceforth in this chapter will use modules except where an array is explicitly defined.

The ensuing section provides details of knowledge abstraction about the Oxford Brookes Joju Solar case study from these literature sources provided above.

9.5.2. Abstraction of Knowledge into Ontology

Based on the analysis of the case study documents, four main concepts emerged. These are the Oxford Brookes University Buildings, the installation company (Joju Solar), the manufacturers of the components used (SunPower) and the components used. The Oxford Brookes Building is considered as an instance of *IfcBuilding*, the installation and manufacturing companies are instances of *IfcPersonAndOrganization*, while the PV module component used as an instance of *SolarModule*. For easy comprehension, Table 9.2 below provides details of the abstracted concepts and how they have been inputted into ifcOWL-pvOntology in protégé to create ifcOWL-pvOntologySunPower.rdf

Table 9.2: Details of Concepts Abstracted from the Case Study into IfcOWL_pvOntology

Instances			
Name	Annotation	Value	Context into IfcOWL-pvOntology
Oxford Brookes	This is the name and location of the buildings where PV-systems were installed	None	This is classified as an instance of the class <i>IfcBuilding</i> .
Joju Solar	This is the company that handled installation for the project.	None	This is classified as an instance of the class <i>Installer</i> with superclass <i>IfcPersonAndOrganization</i>

Sun Power	This is the manufacturer of the components used in this case study	None	This is classified as an instance of the class <i>PvComponentManufacturer</i> with super class <i>IfcPersonAndOrganization</i> .
Maxeon Modules	This is the brand of modules used in the case study	None	This is classified as an instance of the class <i>SolarModule</i> with superclasses <i>PhotovoltaicTechnology</i> & <i>IfcSolarDevice</i> .
Enphase Inverter	This is the microinverter paired with maxeon modules	None	This is classified as an instance of the class <i>Inverter</i> with superclasses <i>ElectricalPhotovoltaicComponent</i> , <i>PhotovoltaicTechnology</i> & <i>IfcSolarDevice</i> .
Data-type Properties for instance Maxeon			
Cost	This is the cost for a single maxeon module used in the case study	\$354	This property is captured under the data property assertion <i>hasCost</i>
Rated Power	This is the nominal power rating of the module	400W	This property is captured under the data property assertion <i>hasRatedPower</i>
Efficiency	This is the conversion efficiency of the module	23%	This property is captured under the data property assertion <i>hasComponentEfficiency</i>
Surface Area	This is the surface area of a single module in cm	1774	This property is captured under the data property assertion <i>hasModuleSurfaceArea</i>
Warranty	This is the warranty coverage period for the module	25 years	This property is captured under the data property assertion <i>hasWarrantyPeriod</i>
Weight	This is the weight of a single maxeon module	1.9 Kg	This property is captured under the data property assertion <i>hasWeight</i>
Cell Orientation	This refers to the number of solar cells contained within a module	104 cells	This property is captured under the data property assertion <i>hasModuleCellOrientation</i>
Assessment Condition	This is the condition of the component at installation time.	Excellent	This property is captured under the data property assertion <i>hasAssessmentCondition</i>
Service Life	The expected service life duration for maxeon is very high.	40years	This property is captured under the data property assertion <i>hasServiceLifeDuration</i>
Object Property Assertions			
Material Type	Here, the material type the maxeon module is made up of is provided	MONOCRYSTALLINE MODULE	This property is captured under the object property assertion <i>hasModuleMaterialType</i>
Installer	This refers to the Photovoltaic systems installer used	JOJUSOLAR	This property is captured under the object property assertion <i>IsInstalledBy</i>
Manufacturer	This refers to the manufacturer of the module used	SUNPOWER	This property is captured under the object property assertion <i>IsManufacturedBy</i>
Annotation Properties			
Annotations provided varied depending on the different properties.			

Note that the *IfcOWL_pvOntology* already contains instances. These concepts from the case study and their properties have now been populated as well into *IfcOWL_pvOntologySunPower.rdf* as shown in figures 9.7 below.

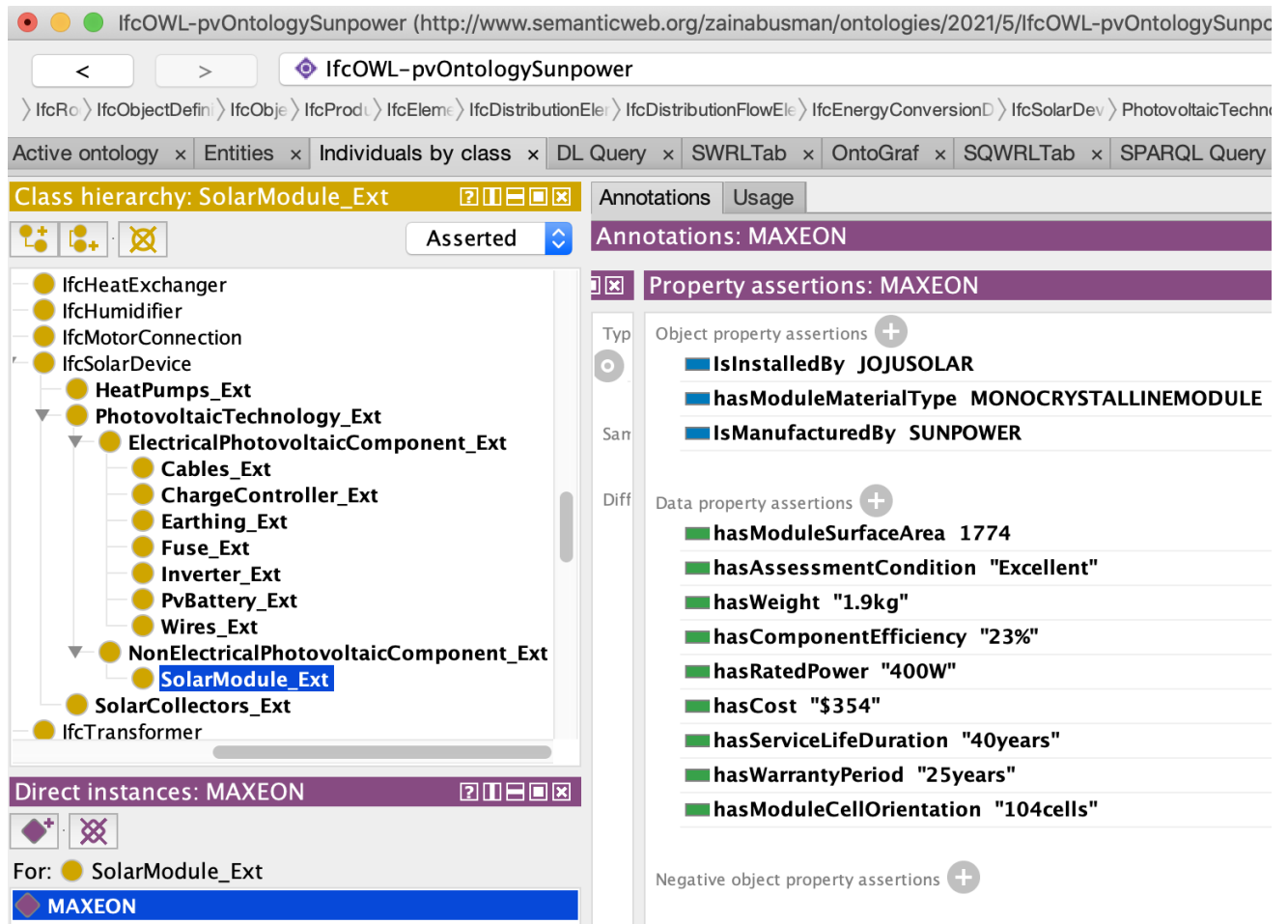


Figure 9.7: Maxeon Module and its Properties in *IfcOWL_pvOntologySunPower.rdf*

From figure 9.7, it can be observed that SunPower's solar module named *Maxeon* is an instance of the class *SolarModule* which is a subclass of *PhotovoltaicTechnology* & *IfcSolarDevice*. Notice the object and data type property values have been added. These values can be manually edited. Although only a single case study is abstracted into the ontology, many other instances have already been populated in *IfcOWL-pvOntologySunPower.rdf*. This adds to the complexity of the system, its reasoning efficiency, and its process of recommending products. After the *IfcOWL_pvOntology* has been populated with the concepts from the case study, it is imperative to test the system

to see if the knowledge added into the model can be used in the web-based application. The ensuing sections,

9.5.3. Design and Selection of PV-system Components for Oxford Brookes

University

One of the main goals of this study is to use the concepts stored in a structured manner within the IFC inspired ontology, to recommend photovoltaic systems components to users. For that reason, the web-based application was developed in chapter 8 to retrieve (and assess) the content of the *IfcOWL_pvOntology* and apply algorithmic computations to the content to recommend photovoltaic system technology. Therefore, it is imperative to check whether the added knowledge model is read correctly and successfully by Jena into the web-based application. The *pvOntologySunPower.rdf* is now stored in the right directory in eclipse. Furthermore, the project package *shams* is refreshed by right clicking on it and selecting the refresh button. This action automatically updates the RDF files within the eclipse directory. In chapter 8, a java file *PvSemanticWeb.java* was created to read the contents of the ontology through defined java methods. Its *getOntology()* method calls the ontology and begins an iterative process of retrieving classes, instances and properties. Therefore, to check if the file successfully reads the updated ontology, the *PvSemanticWeb.java* file is run as a java file. The outcome of this action displays the contents of the updated ontology as seen in figure 9.8 below. Observe the *SolarModule* instance *Maxeon* presented with all its declared data and object property assertions. Notice properties such as *IsManufacturedBy* - *SUNPOWER*, *IsInstalledBy* - *JOJUSOLAR* are presented in capitals because they are also instances within the ontology, and *hasModuleMaterialTypes* - *MONOCRYSTALLINEMODULE* is also an instance of enumerated module material type.

```

425     }
426 }
427 else if(currInstance.getLocalName().equalsIgnoreCase("Maxeon"))
428 {
429     for(StmtIterator ti= currInstance.listProperties(); ti.hasNext();
430         Statement t = ti.next();
431
432         if(t.getPredicate().getLocalName().equals("type") || t.getLocalName().equals("type"))
433             String propertyName = t.getPredicate().getLocalName();
434             String propertyValue = "";
435             if(propertyName.equals("IsInvolvedIn") || propertyName.equals("IsManufacturedBy"))
436
437                 if(propertyName.equals("IsManufacturedBy")) {
438                     propertyValue= t.getResource().getLocalName();
439                     maxeon.put(propertyName, propertyValue);
440                 }
441
442                 else if(propertyName.equals("hasModuleMaterialType")) {
443                     propertyValue= t.getResource().getLocalName();
444                     maxeon.put(propertyName, propertyValue);
445                 }
446                 else if(propertyName.equals("IsInstalledBy")) {
447                     propertyValue= t.getResource().getLocalName();
448                     maxeon.put(propertyName, propertyValue);
449                 }
450                 else {
451                     propertyValue = t.getLiteral().getLexicalForm();
452                 }
453             maxeon.put(propertyName, propertyValue);
454     }

```

Markers Properties Servers Data Source Explorer Snippets Console

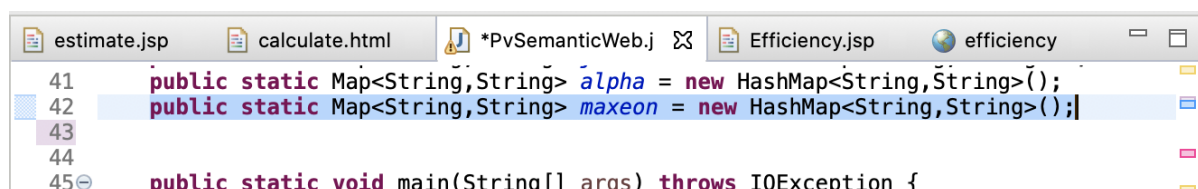
```

<terminated> PvSemanticWeb [Java Application] /Library/Java/JavaVirtualMachines/jdk-16.0.2.jdk
hasModuleMaterialType -> MONOCRYSTALLINEMODULE
hasWarrantyPeriod -> 12years
***** Maxeon *****
IsManufacturedBy -> SUNPOWER
hasComponentEfficiency -> 23%
hasServiceLifeDuration -> 40years
hasRatedPower -> 400W
hasAssessmentCondition -> Excellent
hasWeight -> 1.9kg
hasModuleSurfaceArea -> 1774
hasCost -> $354
IsInstalledBy -> JOJUSOLAR
hasModuleCellOrientation -> 104cells
hasWarrantyPeriod -> 25years
hasModuleMaterialType -> MONOCRYSTALLINEMODULE

```

Figure 9.8: New Knowledge Added into pvOntologySunPower.rdf Successfully Read into Eclipse

Furthermore, as explained in chapter 8, HashMaps are created within the *PvSemanticWeb.java* file to hold values of extracted ontology data so that they can be parsed in subsequent .jsp files to make calculations for photovoltaic sizing. Figure 9.9 below shows new HashMaps defined for maxeon module.

A screenshot of an IDE window titled '*PvSemanticWeb.j'. The code editor shows the following lines:

```
41 public static Map<String,String> alpha = new HashMap<String,String>();  
42 public static Map<String,String> maxeon = new HashMap<String,String>();  
43  
44  
45 public static void main(String[] args) throws IOException {
```

Figure 9.9: HashMaps for Maxeon Module

Once the contents of the updated ontology are read and stored in HashMaps, the system can now be evaluated to check whether the result from its photovoltaic component sizing and recommendation will be consistent with the case study. This is discussed in the ensuing section.

9.5.3.1. Module Sizing and Components Selections

Recall that the algorithms defined in chapter 8 for photovoltaic component sizing and selections were created from the comprehensive process model developed in chapter 6 and based on design methods appraised and recommended in journal publication (Usman et al., 2020). In the *estimate.jsp* file, code to import *PvSemanticWeb.java* is already available (`<%@page import="com.shams.jsp.PvSemanticWeb"%>/`) and is refreshed once the file is run. This import makes all values read into HashMaps available for further use.

Recall a graphical user interface was developed as part of the front-end of the web-based application through which the client interacts with the system by inputting data and getting back desired recommendations. Within that file (*calculate.html*), the user is presented with a form to input their energy requirements. To test the system, the requirement for oxford is inputted into the form as seen in figure 9.10 below. Recall from the case study, an energy offset of 300kW was targeted and solar irradiance value

recorded for Oxford is 3.8 kW/m². Furthermore, an estimated budget of 250,000 pounds (approximately 180,000 dollars) was provided for this project. Observe in figure 9.10 below, the requirements are inputted.

http://localhost:8080/shams/calculate.html

Solar Calculator

Solar Home

Wholesale Max Modules

Solar Shop Nigeria

Simba Solar Solutions

Solar Calculators

Contact Us

Get Recommendation Based on Budget

Budget (\$):

Energy Load of Building (kWh):

Solar radiation of location :

Get Recommendation Based on Output

Energy Load of Building (kWh):

Solar radiation of location :

Figure 9.10: Input of Requirements for Oxford Brookes University – Based on Budget

Observe a submit button *RecommendByBudget* in Figure 9.10 above. The submit button triggers a communication link to be set up between the front-end and back-end of the application through a server. Once the user clicks on this button, the file *calculate.html* runs and automatically executes *estimate.jsp* where all computations for component sizing and recommendation immediately take place based on the inputted values from user requirements. The results of this runtime execution are presented in the ensuing section.

9.5.3.2. Display of Results

Chapter 2 presented a discussion regarding trade-offs that the user often considers in design, sizing, and selection of solar modules. The user could consider the cost of components (by using their budget as a parameter), the size or surface area of array (considering their roof space) and the conversion efficiency (by prioritising efficiency of energy output over cost and surface area). In Chapter 8, it has been outlined that this implementation will overlook the surface area of the user because extensive review shows that the average residential building always has sufficient roof (or other mounting types) surface area to install the required array size (Palmetto, 2021). A similar assumption is valid for Oxford Brookes University since it is a school campus and does not run high-energy powered industrial machines. Therefore, for the context of this study, we provide results based on budget and efficiency of energy output.

Recommendation based on budget - Once the file is executed over the server, it presents all results for all module instances as seen in Figure 9.11 (a) and 9.11 (b) below. Figure 9.11 (a) shows the recommended module. Observe that for each module, the system displays its *efficiency* and *manufacturer*. Similarly, the system computes the array size (*number of modules*) required to generate 300kW of energy based on the rated power of each module provided from *hasRatedPower* property within the ifcOWL-pvOntology. Furthermore, based on the array size and the cost per module provided from the *hasCost* property assertion in the ontology, the *total array cost* for each module type is determined and presented. Based on these values, an estimated cost of inverters and battery and the *total cost of components* (for each type of module) is provided. The total component cost if each of the modules is selected is compared to the budget given by the user. The system takes an absolute difference and recommends the module whose total component cost comes closest to the budget.

It can be observed in Figure 9.11(a) that the system recommends *Ultron* modules which if selected, the total component cost is closest to the user's budget. The PV-system will require 822 pieces of *Ultron* assembled together to form an array that would generate 300kW of energy (the electric output required by the user). The difference from the provided budget is \$8,378. It is important to note that because an absolute difference is

obtained, it is possible for the recommended module to be slightly above the budget but must be the closest amount to the budget. In Figure 9.11 (b), notice that *SunModule* and *Perc* are both above the budget by \$13,200 and \$70,344 respectively. Both are way off the budget and therefore not recommended. It can be seen that the results provided for the Maxeon module shows 699 of maxeon modules are required to generate the required energy output of 300kWp. Although maxeon isn't the recommended module based on budget, its array size of 699 (compared to the array sizes computed for the other modules), comes closest to the array size provided in the case study (700). This proves the effectiveness of the algorithms developed in this study and confirms validation for the design approach used. Furthermore, the total component cost if maxeon is selected is \$132,059 and below the budget by \$47,941 making it equally suitable for the project.

Figure 9.11 (a): Display of Results – Recommended Module Based on Budget

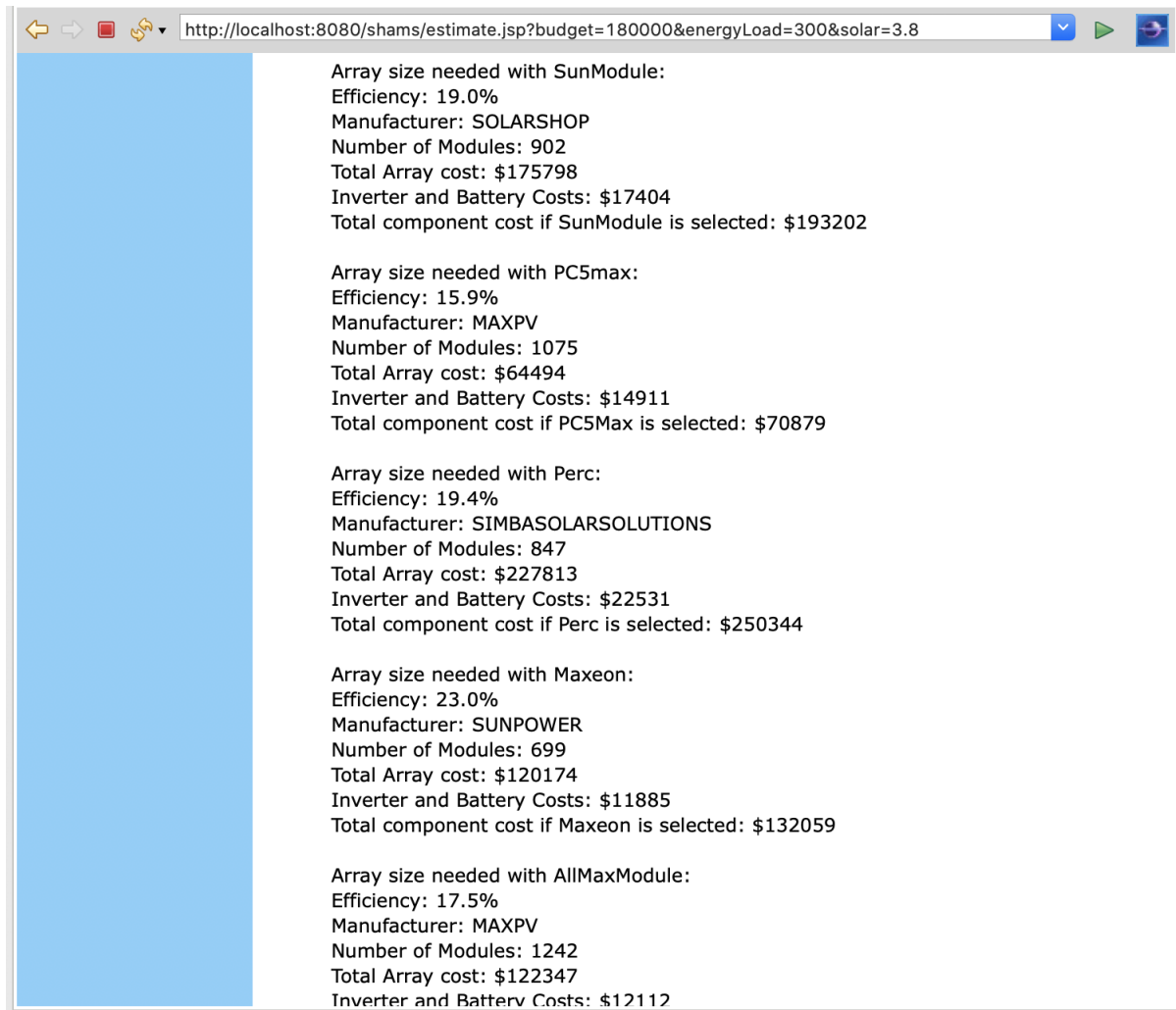


Figure 9.11 (b): Display of Results – Other Module Types and Their Details

If more than one of the components falls within the budget of the user, the user could request for the system to recommend the component that would generate the required energy output most efficiently. Results for recommendations based on output are provided below.

Recommendation based on best electric output – the review on solar modules in chapter 2 provided that the most efficient modules produce more electricity from square metres of its mounting space. Therefore, the most efficient modules have the smallest surface area, but much higher costs compared to other conventional modules. If the user is not concerned about the cost of components and wants to generate as much electricity as possible, then the module capable of achieving that should be selected. To

recommend a module based on best electric output, the system prompts the user to input only its energy needs and solar irradiance at the front end of the web application. Figure 9.12 below shows the *RecommendByOutput* button. This button automatically executes *output.jsp* over the server. The result of this execution is a component recommendation presented in figure 9.13 below.

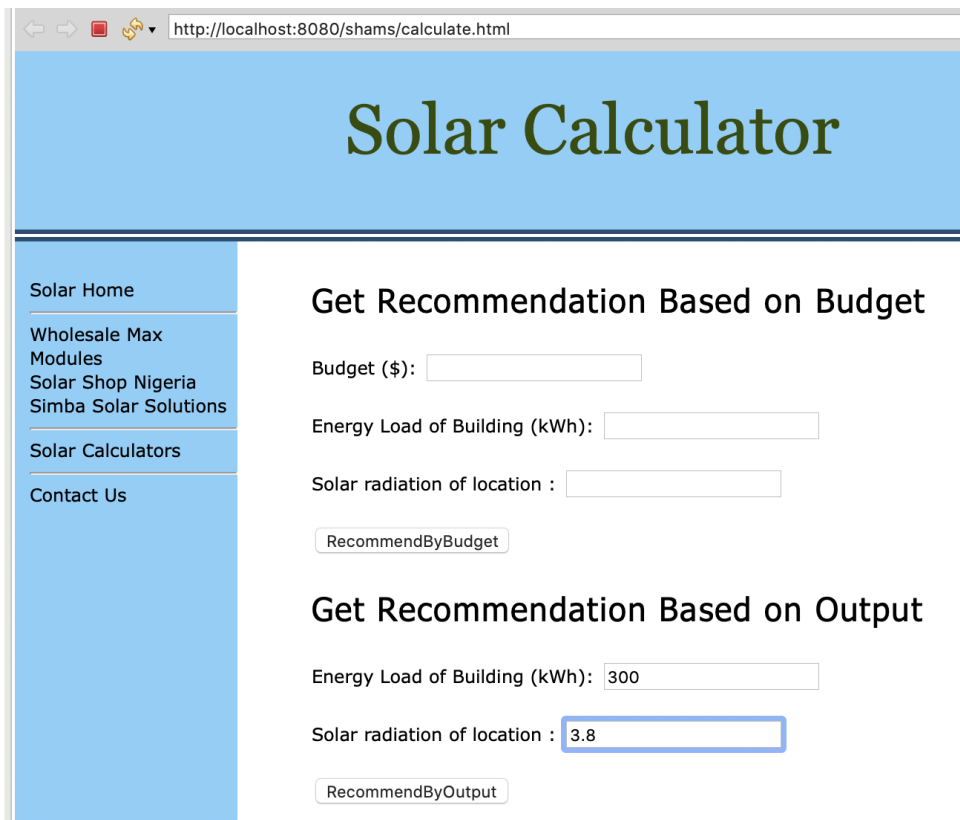


Figure 9.12: Input Requirements Based on Electric Output

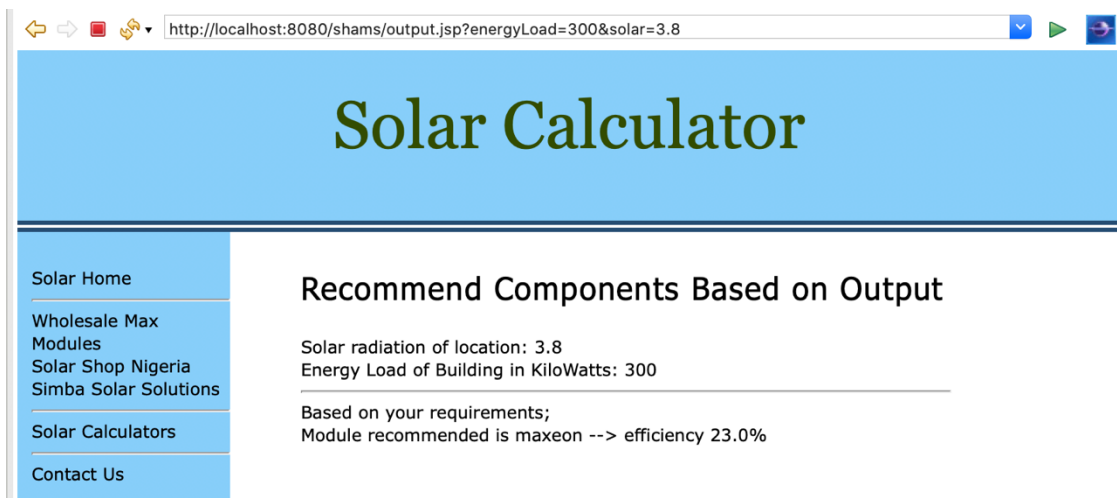


Figure 9.13: Recommendation Based on Electric Output

The system recommends maxeon module and presents its efficiency. This means that of all the available solar modules, maxeon module would generate the highest electric output per square metre of roof space. To users who are concerned with quality of the electric energy generated, the modules with the highest efficiency should always be obvious choice. There are several other advantages to selecting the module with the highest efficiency. Firstly, they have relatively much smaller surface area compared to modules of low efficiency. Secondly, high efficiency modules are backed by long warranties. Maxeon module has a warranty period of 25 years thereby minimising risks associated with investment. Thirdly, to some users, aesthetics matter. The high efficiency modules are designed in such a way that they look cleaner and sleeker than convention modules. Its only disadvantage is its high cost. Regardless, the user decides which aspect of photovoltaic design to trade off, either costs or performance.

The use of case study has confirmed the validation of the prototype system and established the developed system meets the system requirements. It loads in the contents of the IfcOWL_pvOntology and runs java and .jsp files successfully to perform sizing computations and recommend photovoltaic components based on user requirements. It is important to note that prototype evaluation is a repetitive process. It is done continuously until satisfied, and the desired application is attained. Evaluation conducted proves the pragmatic quality of the prototype system and its communication aspect. The developed prototype system fits well into the repository of existing reusable IFC components and is recommended for use in practice.

9.6. Conclusion

This chapter discussed the evaluation techniques and criteria used to verify and validate the reliability and correctness of the ifcOWL-PvOntology and the web-based application. This includes syntactic and semantic verification, quality evaluation, instance evaluation, and validation through a case study.

Semantic verification was an iterative process until no anomalies or errors were obtained. This was done using semantic reasoners available as plugins within protégé-owl. Similar, syntactic verification was conducted at different stages of the ontology modelling where new faults were added to test the ontology reasoning capabilities. Furthermore, a statistical framework based on criteria-based evaluation and goal question metric was used to test the quality of the developed ontology. After, an instance evaluation is done to ensure all instances within the knowledge model are syntactically and semantically valid. After a complete verification of the ontology was conducted, the prototype system was validated using a case study with real concepts and values. These values were added into the ontology to carry the IFC-based semantic web structure of representation. Algorithms developed were applied to demonstrate how the web-based application sizes and recommends desirable photovoltaic components to the user based on the values of the components retrieved by the system from the ontology. Screenshots were used where applicable for easy comprehension. The prototype system effectively reads in the content of the ontology and sizes photovoltaic components based on user requirements. It provides recommendations to the user based on either components that best fit the user's budget or components that would most efficiently generate the highest electric output per square metre of mounting space to the user. The prototype system was generally evaluated based on its reliability, modularity, usability, effectiveness, correctness and on how best tasks are completed.

10. CONCLUSION

10.1. General

This chapter summarises the achievements of this research and presents its contributions to knowledge. Limitations of the research and capacity for future studies are also presented.

The rationale of this study stems from the need to mitigate global environmental impacts caused by carbon emissions, especially from activities in the construction industry. One of the established ways of reducing climate change effects is to adopt sustainable building technology products into construction projects. However, the limitations of the most popular form of finding information on any domain i.e., the world wide web does not facilitate effective elicitation of relevant knowledge on sustainability products to interested users, thereby hindering the uptake of the technologies. This led to the need to develop effective knowledge modelling techniques such as the semantic web which overcomes the limitations of the current web, represents knowledge in machine-processable formats and processes the obtained knowledge to provide solutions to the domain. Due to the vastness of the domain, the study focussed on photovoltaic system domain and investigated the capabilities of the semantic web to provide decision support. The domain can expand to cover other sustainable building technologies in future studies.

This study aims to investigate the extent to which semantic web technologies can be used for knowledge modelling to support informed decision making and recommendations on the selection of sustainable building technology products. The objectives are to:

- a. to identify challenges associated with managing information and selection of sustainable building technology products on the web.
- b. develop a conceptual knowledge model of a representative sustainable building technology product problem domain.

- c. design and implement a prototype system that demonstrates the potential of semantic web technologies to provide decision support in the selected problem domain based on (b) above.
- d. Evaluate the prototype system and make recommendations on the implementation of such systems in practice.

10.2. Attainment of Research Objectives

Objective a was achieved through a critical review of sustainable building technologies and emerging semantic web technologies. The review of SBTPs undertaken focused on the PV systems domain. The review identified gaps in the literature, established the state-of-the-art PV technology components, their distinctive characteristics, challenges faced in design and selection due to multiple criteria and the requisite for developing efficient ways of knowledge modelling for the domain. Since the design of PV is centred on its diverse components, a detailed review was done on the unique attributes, contributions, and relevance of each PV component to the overall system. After several multiplicities in the PV system design have been explored, a unique and exhaustive sizing and selection model was proposed. This process model is supported with mathematical computations and therefore effective for the selection of PV. After the SBTP domain has been explored, state-of-the-art semantic web technologies were critically reviewed to identify their capabilities to provide decision support and recommendation on the design and selection of SBTPs. The review further includes identification of existing domain knowledge bases for reuse to ensure the development of rich domain knowledge. The outcome of the review on semantic web domain identified the most suitable technologies to be used in the study to overcome the limitations of the current web and efficiently model knowledge on the problem domain. The review on sustainable building technology and semantic web technology established the theoretical underpinning for the research.

Once the PV system design was developed, the theoretical knowledge for this research was extended to develop the conceptual knowledge (objective b) using the classification task template of the CommonKADS research framework. An initial knowledge acquisition was undertaken. Domain concepts were extracted from knowledge identified in **objective a** and modelled into Unified Modelling Language (UML) diagrams. UML diagrams also described the set of actions involved with the PV selection process including roles played by external users. Furthermore, a complete knowledge specification was undertaken to define all class hierarchies, properties, relations, and constraints of PV concepts. Lastly, a knowledge refinement was done to add any class instances left out into the UML. Thus, the complete conceptual knowledge model was developed.

Objective c was achieved by exploiting semantic web and its technologies to provide decision support for the PV technology domain based on the conceptual knowledge model developed in object b above. To avoid an isolated conceptualisation of the domain, this research further identified, studied, and reviewed domain-specific online product repositories and taxonomies in order to enrich the domain ontology. Subsequently, this research integrated the proposed ontology with Industry Foundation Classes (IFC) to develop an enhanced knowledge conceptualization of the PV domain - ifcOWL-pvOntology. Ontology reasoners were used on the ontology consistently to check for anomalies and inconsistencies.

Objective d involved thorough evaluation of the prototype application; an IFC inspired semantic web-based navigable interface. The prototype system serves as a proof-by-demonstration to verify and provide proof of the concepts established in the research. Evaluation was conducted through semantic and syntactic verification using ontology reasoners. Validation of the prototype was conducted using real-life data through a case study. Challenges, limitations, and findings during research lead to the establishment of recommendations on the future application of semantic web technologies in practice. The successfully verified and validated prototype system that demonstrates the capabilities of semantic web technology on providing decision support to the problem domain led to the achievement of the aim of this study.

10.3. Key Findings from Literature

The literature review conducted to examine sustainable building technology products and photovoltaic systems technology led to key findings. These are outlined below:

The primary finding was the abundance of emerging sustainable building technology products and PV technology components. Manufacturers have taken up the challenge to develop components with higher efficiency than those available in the market. Policies and incentives from government agencies have increased the uptake of the technologies thereby increasing the demand. Construction professionals and researchers have also continued to publish enormous amounts of knowledge over the web on the design and adoption of SBTPs and PV technologies. This exponential growth of large volumes of information about PV technology was documented as one of the overwhelming challenges in adequate knowledge acquisition faced by professionals and end users. This has hindered the uptake of the technology.

A similar challenge faced by professionals and end-users is lack of standard terms in describing the different PV technologies and their components. Construction experts involved in different disciplines within the construction domain use unique terminologies to describe products and these terms vary from one sub-field to another. Similarly, terms often differ among product manufacturers. For instance, some manufacturers define a PV panel as a module, while others define a PV panel as a selection of modules assembled together. Fortunately, the emergence of domain taxonomies such as IFC, offers a unified and consistent taxonomy for knowledge exchange in the construction domain. However, IFC itself is incomplete and evolving, thereby needing to be extended as proposed in this study.

The review of literature disclosed that there is often a great difference between the energy output projected in the PV design stage and the actual output obtained after its installation. This shows the presence of errors in design approaches and methodologies. There is a paucity in literature about the best methods of designing PV-systems and their disparities. Similarly, there are often significant differences in PV electric output

when these methods are compared to each other due to differences in approaches. This led to the need to propose an effective PV sizing approach that considers the complexity in the permutations of PV components and their unique features such as efficiencies, performance indicators and sizes. The use of an effective design approach ensures the resulting PV system generates the required and optimised electric output.

Another important finding from the literature is the availability of different properties for characterising PV technology components. Some components share common properties such as CO₂ emissions offsets, efficiency, etc. However, they have unique features that distinguish one component type to another. The differences in these properties are vital in making informed decisions about which technology to uptake and how to size them. These unique properties were applied in the study as object and data properties whose values were used in sizing and recommending PV components to the user. Formats of information representation about PV technology increased the difficulty in its elicitation process. A primary finding shows the limitations of the current web may have hindered the uptake of PV technologies. Firstly, is the inability of the web search engines to retrieve information that is relevant to the user. Secondly, information is represented in a non-machine processable format and cannot be reasoned on. This finding is the main reason behind exploring semantic web technologies to represent knowledge on PV technology in machine processable formats.

The review of semantic web technologies disclosed the generic nature of previous applications of semantic web in the construction domain. Firstly, most previous studies developed top level concepts without detailed sub concepts. Secondly, previous studies have developed isolated conceptualization of the PV domain based on their terms without using standard domain taxonomies in representing construction knowledge. Such systems have low chances of being used in practice or standardisation. It is for this reason that this study re-used and extended the IFC in its application. Equally, previous studies have developed desktop-based solutions. While desktop applications are confined, web-based applications are accessible to users worldwide.

Another important key finding in literature is the vastness of huge repositories of applications of semantic web in different fields including medicine, engineering, and teaching. Similarly, there are numerous frameworks that exist for developing semantic web applications. These include several semantic web languages, development environments, and application programming interfaces that support reasoning and semantic evaluation. Furthermore, review of literature shows the progressive nature of these technologies. Since its emergence, the semantic web languages have been evolving and growing in terms of power of expressiveness and efficiency in reasoning. For instance, OWL language used in this study overcomes the challenges of the preceding languages and provides sufficient expressiveness to support the semantic web. Although the semantic web as a whole is evolving, its continuous adoption by researchers confirms the strength of the technologies in providing solutions to domain challenges.

10.4. Contribution to Knowledge

This study developed a comprehensive conceptual model of the sustainable building technology domain with focus on photovoltaic systems to demonstrate the capabilities of the semantic web. Detailed sub-level concepts of the PV domain were defined to facilitate elicitation of rich semantics of the domain. The proposed exhaustive PV selection model and the recommended algorithmic approach for sizing and selection of photovoltaic technology components involving multicriterial design parameters, informed the conceptual knowledge model. While the above achievement is based on exhaustive research and meticulous identification of important knowledge gaps in the domain, it led to three main contributions to knowledge that are exclusive and distinctive to this research. The three key contributions to knowledge are presented below.

The first key contribution of this research is the proposed Semantic Web-based methodological research framework for knowledge representation and decision support for making informed decisions about the use of sustainable building technology products. This methodological framework is unique because it encompasses several research methodologies in order to achieve the objectives and aim of this research. The

application of knowledge engineering methodologies, ontology engineering approaches, software engineering approach with exploratory qualitative research methods was strategic and distinctive.

The second contribution is the integration of the proposed PV ontology with IFC for efficient knowledge discovery about building domain knowledge. The ontology developed in this study re-uses and extends IFC to ensure the resulting ontology is rich in domain standard terms and has a possibility of being standardised. The *IfcOWL_pvOntology* is a PV specific domain implementation based on the buildingSMART standard *ifcOWL* ontology. This ontology is available on the web for PV component manufacturers to populate the ontology with their products. This makes the prototype scalable. The *IfcOWL_pvOntology* could be used as an official extension of IFC ontology for the photovoltaic technology domain. It could be adopted and approved by the buildingSMART working group in the W3C as a standard ontology for photovoltaic system, inspired by the IFC standard.

The third contribution is a Semantic Web navigable interface for exploring the integrated IFC and PV ontology decision support system. This user interface called the Semantic IFC-inspired Photovoltaic System (SIPS) executes computations to provide recommendations on photovoltaic technology products to the system users based on the PV knowledge captured in the *IfcOWL_pvOntology*. This prototype will be treated as part of a future large expert system where the knowledge and inference models will be used in practice to provide decision support for not only photovoltaic systems technology but other SBTPs at large.

There are PV calculators available online that import BIM models and provide PV projections based on the specifications of the building. The prototype developed in this research is different from available calculators because it focuses on exact specifications of existing PV products in the market. While BIM models focus on the specifications of a building, SIPS focusses on modelling photovoltaic technology components and recommends to the user a product represented in machine processable formats. This approach encourages product manufacturers to represent their products in RDF rather

than the conventional HTML methods. Continuous uptake of this approach would eliminate the limitations of the current web and increase the uptake of PV technology and sustainable building technology products.

10.5. Implications for Practice

The proposed prototype system is recommended to be used in practice. One of the main drawbacks for PV adoption is lack of understanding of the technology and lack of in-depth knowledge of the design methods. It has been established in this study that there is not a single standard method of selecting and sizing of PV components. Different methods for the design of PV-systems require different assumptions, leading to different inaccuracies in the design of the different components and systems. Furthermore, poor design methods create a great difference between the projected energy output and the actual output obtained after installation. This discourages adoption of photovoltaic system technologies. Therefore, since this research proposes an evaluated decision model, one of the potential benefits is an increase in the adoption of PV is expected.

Another challenge in the uptake of PV-systems is also the difficulty users face when trying to obtain relevant data on products over the web. The implication of this study is that knowledge modelling of information on PV systems and products is obtained and PV products can be represented in machine processable formats. Likewise, a decision model to help users select and size their PV system using specific attributes of the product they explicitly require is feasible. Additionally, previous studies have developed desktop-based solutions using legacy database languages such as SQL, which do not handle unstructured data well. This study proposes a web-based application that can be accessible to users worldwide giving users access to a rich domain knowledge model. The developed application is functional, and the graphical user interface is friendly and applicable.

One of the main drawbacks of the current web established in this study is its representation of data in html formats for instance, PV components represented on

manufacturer websites. The prototype system in this study reads in product manufacturer ontologies and retrieves its instances (components and their values) in the form of RDF to provide decision support to the user. An important implication of this study for practice in the construction industry is that it will encourage product manufacturers to use semantic web and its technologies in representing their products so that they can be machine-processable.

Most previous studies in the domain of sustainable building technology products and photovoltaic systems developed concepts based on their terms and from human experts without using standard domain taxonomies in representing construction knowledge. It is important to mention they are an isolated conceptualization of the domain. They cannot serve their purpose for long since domain terms used in literature differ from one expert to the other. This study contributes to practise through the ontology-based knowledge model it has developed that is an extension of the standard ifcOWL ontology. Firstly, the ifcOWL-pvOntology is stored in an OWL format primarily to allow further extensions of the ontology. This is because only continuous use can make standards advance. Secondly, the IfcOWL_pvOntology is a modular ontology for the photovoltaic technology domain, it could be adopted and approved by the buildingSMART working group in the W3C as a standard ontology for photovoltaic systems, inspired by the IFC standard.

10.6. Challenges and Limitations of the Research

The challenges faced in this study were difficulties encountered during research and prototype development while limitations are concerns that can affect future studies and results.

10.6.1. Challenges

One of the main challenges encountered during the course of this research is exploring the most effective way of integrating semantic web with standard domain taxonomies in order to obtain a rich domain knowledge model. Similarly, learning resources were very limited with scarce online resources for learning. Extending the IFC was challenging. Firstly, there are new concepts within the updated version of the model view definition

(MVD) database (IFC4 ADD2 TC1) used in this study that are not available within the standard ifcOWL ontology. Therefore, too much time and energy went into thoroughly traversing the bulky IFC schema documentation to identify the missing concepts in the ontology. Secondly, the current IFC specifications do not fully capture all entities of the PV technology domain, therefore, this research extended it with knowledge obtained during the research. Key challenges were faced in determining the proper position within ifcOWL ontology to add new concepts from the conceptual knowledge model developed in this research. The position for some classes such as the *IfcSolarDevice* and its subsequent subclass were obvious, however, it was challenging to determine the appropriate position to place other concepts on the existing ifcOWL to establish the appropriate class hierarchy. For instance, a major challenge was faced when deciding the appropriate hierarchy for an *organisation* which could be suitable as a subclass of either *IfcActor* or *IfcActorSelect*. Determination of the most appropriate IFC parent class required comprehensive understanding of the domain concepts involved and ample understanding of the IFC schema specifications. The following subsection outlines the limitations of the research.

10.6.2. Limitations

The review of literature presented vast volumes of information on sustainable building technology products. Although there are a few similarities in terms of properties of some of the technology products, the differences are more. There are major differences in terms of design and selection methods. Design and selection of heat pump technologies is completely different from PV systems technology. Therefore, due to the restricted time frame of this research, it was not possible to investigate all sustainable building technologies. The research focussed on exploring the decision support system for PV technology. However, the study can be extended to cover other technologies within the sustainable building products domain.

This research focused on providing decision support for PV technology based on detailed sizing and selection of modules, inverters, and batteries because they are the most important and expensive components. In practice, charge controllers, cables, wires, fuses, maximum power point tracking, and earthing are designed as well even

though their design depends on the outcome of the sizes for modules, inverters, and batteries.

Recall the study established the trade-off between installation budget, efficiency of electric output and surface area of the mounting space. The prototype system recommends PV components based on the budget of the user and the efficiency of the electric output without considering the surface area. This is a limitation because the assumption that a typical building should always have sufficient roof space to accommodate the array size it requires is no longer true. As observed in the case study, Oxford Brookes University requires innovative ways to create mounting surface space for future PV installations. Nonetheless, there is a scope to provide recommendations to users based on their mounting surface area.

10.7. Recommendation for Further Studies

There are some areas that require further investigation in future studies within the research domain. This study has identified a few areas to be studied to enhance the application of semantic web technologies in modelling knowledge on sustainable building technology products and recommending providing support to decision making in the domain. These areas include an expansion on the scope of the domain, automatic extension of re-used ontologies and an enhancement of the prototype system.

Firstly, this study reviewed the domain of sustainable building technology products. These include emerging smart building technologies, waste and water minimization technologies, wind energy technologies, hydro energy technologies, biomass generators, concentrated solar power (CSP), thermal collectors and heat pumps, solar heating and cooling (SHC) technologies, and photovoltaic systems technologies. During this study, the benefits of these technologies have been established. Furthermore, their similarities such as being sustainable energy technologies from pure and clean sources have also been discussed. However, each of these technologies have unique and great deal of differences in terms of their designs and adoptions. Since this study has proven that the semantic web technologies can be adopted to aid decision making in the domain of

photovoltaic systems, the scope can be increased to cover all other sustainable building technology products.

Another aspect recommended for future research is enhancement of the prototype system. Firstly, the prototype system reads a single amount as the energy load of the user's building and uses that value for computations. This study could be extended to include automatic generation of user requirements. Further studies could use APIs to retrieve solar radiation of the user's location directly from meteorological stations. Furthermore, APIs that import BIM models of buildings could also be used to automatically generate the mounting surface area of the building. However, because the research presented is a prototype, it did not focus on automation, rather it focused on proof of concept to verify that research findings and the proposed models are correct and can be implemented successfully to provide decision support.

Secondly, further studies can perform automatic verification of manufacturers. Manufacturers used within the system should be certified by the leading sustainability certification standards such as BRE, energy savings trust. Thirdly, the prototype developed focused on providing decision support for selection of PV technology based on detailed sizing and selection of modules. This is because modules are the most important and expensive components. It is also due to the large volume of work and time constraints that prototype development as part of a research entails. However, the sizes and costs of inverters, batteries, charge controllers, cables, wires, fuses, also have real effects in practice. Therefore there is a scope to investigate how these components can be incorporated into the decision support tool.

The proposed prototype system is recommended to be used in practice. It proposes an ontology that is a structured representation of a knowledge base for the photovoltaic systems technology. Standard taxonomies were reused and extended during development. The developed prototype aids the user in sizing and selection of PV components represented in RDF. The graphical user interface is friendly and applicable. Challenges and limitations faced during the research and prototype development are

recorded for future research studies and implementations. This prototype could be extended to recommend more sustainable building technology products.

REFERENCE

- Abanda, H. *et al.* (2015) 'A critical analysis of Building Information Modelling systems used in construction projects', *Advances in Engineering Software*, 90, pp. 183–201. doi: 10.1016/j.advengsoft.2015.08.009.
- Abanda, F. H., Kamsu-Foguem, B., & Tah, J. (2017). 'BIM – New Rules of Measurement Ontology for Construction Cost Estimation'. *Journal of Engineering Science and Technology*, 20 (2), 443-459.
- Abanda, F. H., Tah, J. H. M. and Duce, D. (2013) 'PV-TONS: A photovoltaic technology ontology system for the design of PV-systems', *Engineering Applications of Artificial Intelligence*, 26(4), pp. 1399–1412. doi: 10.1016/j.engappai.2012.10.010.
- Abanda, F. H., Tah, J. H. M. and Keivani, R. (2013) 'Trends in built environment Semantic Web applications: Where are we today?', *Expert Systems with Applications*, 40(14), pp. 5563–5577. doi: 10.1016/j.eswa.2013.04.027.
- Aberer, K., Chau, V. T., Gal, A., Nguyen, Q. H., Nguyen, T., Miklos, Z., Wijaya, T. K., Weidlich, M. (2015), 'SMART: A tool for analysing and reconciling schema matching networks', In Proceedings of ICDE
- Abburu, S. and Golla, S. B. (2016) 'Ontology Storage Models and Tools: An Authentic Survey', *Journal of Intelligent Systems*, 25(4), pp. 539–553. doi: doi:10.1515/jisys-2014-0167.
- Adan, A., Quintana, B., Prieto, S. A., & Bosche, F. (2020). 'An autonomous robotic platform for automatic extraction of detailed semantic models of buildings', *Automation in Construction*, 109(September 2019), p. 102963. doi: 10.1016/j.autcon.2019.102963.
- Adobe, 2021. *Adobe*. [Online] Available at: <https://www.adobe.com> [Accessed 9 March 2021].
- Ahmad, S., Wasim, S., Irfan, S., Gogoi, S., Srivastava, A., & Farheen, Z. (2019). 'Qualitative v/s Quantitative Research'. 6, 2828–2832. <https://doi.org/10.18410/jebmh/2019/587>
- Ahmed, B.; Necaibia, A.; Slimani, A. A (2020), 'Demonstrative Overview of Photovoltaic Systems Faults'. Proceedings of the 1st Global Power, Energy and Communication Conference (IEEE GPECOM2019), Turkey, Nevsehir, Turkey, 12–15 June 2019. Buildings 2020, 10, 192 21 of 22
- Airswift. (2022, July 27). 'Everything you need to know about Concentrated Solar Power technology'. Retrieved February 2023, from Airswift: <https://www.airswift.com/blog/concentrated-solar-power>
- Alavi, H., Bortolini, R. and Forcada, N. (2022) 'BIM-based decision support for building condition assessment', *Automation in Construction*, 135, p. 104117. doi: <https://doi.org/10.1016/j.autcon.2021.104117>.
- Al-Ghussain, L. *et al.* (2023) 'Optimising the orientation of solar photovoltaic systems considering the effects of irradiation and cell temperature models with dust accumulation', *Solar Energy*, 249, pp. 67–80. doi: <https://doi.org/10.1016/j.solener.2022.11.029>.
- Ali N, Lai R. (2021), Global software development: a review of its practices. *Malaysian J Comput Sci* 2021;34(1):82–129. Retrieved from, <https://ejournal.um.edu.my/index.php/MJCS/article/view/28581>.
- Allu-Kangkum, E. (2023) 'Global Sustainable Architectural Trends: Top Three Trends In Nigeria And Their Implications', 6, pp. 25–32.
- Alsadi, S.; Khatib, T. (2018) Photovoltaic Power Systems Optimization Research Status: A Review of Criteria, Constraints, Models, Techniques, and Software Tools. (2018) *Appl. Sci.* 2018, 8, 1761.
- Al-Soud, M. and Alsafesfah, Q. H. (2015), 'Economical Evaluation for Various Renewable Energy Products in Jordan,' 2015 6th International Renewable Energy Congress (IREC)'.
- Alternative Energy (2019). Alternative Energy Battery Bank Sizing. Available online: <http://www.altenergy.org/renewables/solar/DIY/battery-bank-sizing.html> (accessed on 12 January 2019).
- Ameen, A., Khan, K. R., and Rani, B. P. (2014), 'Extracting Knowledge from Ontology using Jena for Semantic Web', IEEE international conferences for convergence of technology, Pune, DOI:[10.1109/I2CT.2014.7092025](https://doi.org/10.1109/I2CT.2014.7092025)
- Antonios, P., Konstantinos, K. and Christos, G. (2023) 'A Systematic Review on Semantic Interoperability in the IoE-enabled Smart Cities', *Internet of Things*, p. 100754. doi: <https://doi.org/10.1016/j.iot.2023.100754>.
- Antoniou, G., Groth, P., van Harmelen, F., and Hoekstra, R. (2012). A Semantic Web Primer, Third Edition. The MIT Press, Massachusetts.
- Antunes, A. L., Cardoso, E. and Barateiro, J. (2022) 'Incorporation of Ontologies in Data Warehouse/Business Intelligence Systems - A Systematic Literature Review', *International Journal*

- of *Information Management Data Insights*, 2(2), p. 100131. doi: <https://doi.org/10.1016/j.jiime.2022.100131>.
- Antwi-Afari, M. F. *et al.* (2018) 'Critical success factors for implementing building information modelling (BIM): A longitudinal review', *Automation in Construction*, 91, pp. 100–110. doi: <https://doi.org/10.1016/j.autcon.2018.03.010>.
- Apostoleris, H.; Sgouridis, S.; Stefancich, M.; Chiesa, M. (2019) Utility solar prices will continue to drop all over the world even without subsidies. *Nat. Energy* 2019, 4, 833–834.
- Agarwal, R., Fernandez, D.G., Elsaleh, T., and Gyrard, A. (2018), *FIESTA-IoT Ontology*. doi: 10.5281/zenodo.1193299.
- ARCHLine, (2021). *Software Communication: How To Import And Export IFC Files In ARCHLine.XP?*. [Online] Available at: <https://www.archlinexp.com/blog/professional-articles/software-communication-how-to-import-and-export-ifc-files-in-archline-xp>
- Aresolar. (2023), 'The Benefits of Going Solar: Helping the Environment While Saving You Money. *Aresolar*. Date accessed 2 March 2023. Available online: <https://aresolar.com/the-benefits-of-going-solar-helping-the-environment-while-saving>
- Van Assche, D. *et al.* (2023) 'Declarative RDF graph generation from heterogeneous (semi-)structured data: A systematic literature review', *Journal of Web Semantics*, 75, p. 100753. doi: <https://doi.org/10.1016/j.websem.2022.100753>.
- Augenbroe, G., Pearce, A.R. and Kibert, C.J. (2010) "Sustainable Construction in the United States of America: A perspective to the year 2010", CIB-W82
- Aurora. (2023). *How to Estimate a Homeowner's Energy Load Profile*. Retrieved 2023, from Aurora: <https://aurorasolar.com/blog/how-to-estimate-a-homeowners-energy-load-profile/> (Accessed 2023, February 12)
- Autodesk, (2018). *Autodesk Revit IFC Manual*. [Online] Available at: https://damassets.autodesk.net/content/dam/autodesk/drafrtr/2528/180213_IFCHandbuch.pdf [Accessed March 2021].
- Autodesk, (2021). *Revit*. [Online] Available at: <https://www.autodesk.com/products/revit/overview? ga=2.111400132.15735322701614850810-662938897.1614700741>
- Autodesk, (2022). 'Architecture Engineering and Construction'. <https://www.autodesk.co.uk>
- Ayman, H. M., Mahfouz, S. Y. and Alhady, A. (2022) 'Integrated EDM and 4D BIM-Based Decision Support System for Construction Projects Control', *Buildings*, 12(3). doi: 10.3390/buildings12030315.
- Ayompe, L.M.; Duffy, A.; McCormack, S.J.; Conlon, M. (2010), 'Validated real-time energy models for small-scale grid-connected PV-systems'. *Energy* 2010, 35, 4086–4091.
- Bandeira, J., Bittencourt, I., Espinheira, P. & Isotani, S., (2017). FOCA: A Methodology for Ontology Evaluation. 2 September.
- Bansal, M., Khatod, D.K., Saini, R.P. (2014), 'Modelling and Optimization of Integrated Renewable Energy System for a Rural Site', 14 International Conference on Reliability, Optimization and Information Technology - ICROIT 2014, India, Feb 6-8
- Baonza, M. (2010), 'NeOn Methodology for Building Ontology Networks: Specification, Scheduling and Reuse', Doctoral Thesis.
- Barbose, G., Darghouth, N. & Wiser, R. (2013), 'Tracking the Sun VI. An Historical Summary of the Installed Price of Photovoltaics in the United States from 1998 to 2012', LBNL report 6350E.
- Baumeister, J. and Seipel, D. (2005) *Smelly Owls - Design Anomalies in Ontologies*.
- Bazeley, P. (2015), 'Adoption of Mixed Approaches to Research by Management Researchers,' Proceedings of the 14th European Conference on Research Methods 2015: ECRM 2015.
- Bazilian, M., Onyeji, I., Liebreich, M., MacGill, I., Chase, J., Shah, J., Gielen, D., Arent, D., Landfear, D., Zhengrong, S. (2013), 'Reconsidering the economics of photovoltaic power', *Renewable Energy* 52, 329-338.
- Bechhofer, S., Horridge, M. (2011), 'OWL API: A Java API for OWL Ontologies', *Semantic Web* 2 (1) pp. 11–21.
- Beck, A. and Rai, V. (2019) 'Solar soft cost ontology: A review of solar soft costs', *Progress in Energy*, 2. doi: 10.1088/2516-1083/ab59be.
- Beckmann MJ. (1995), *Economic Models of Knowledge Networks. Networks in Action*. Berlin, Heidelberg: Springer; 1995.

- Bell, R. (2022). 'Construction Industry looks to BIM to improve decision-making, collaboration: Survey', Oracle Construction and Engineering Blog. <https://blogs.oracle.com/construction-engineering/post/construction-bim-collaboration-survey>
- Belozarov, A. A. and Klimov, V. V (2022) 'Semantic Web Technologies: Issues and Possible Ways of Development', *Procedia Computer Science*, 213, pp. 617–622. doi: <https://doi.org/10.1016/j.procs.2022.11.112>.
- Belsky, M., Sacks, R., & Brilakis, I. (2015). Semantic Enrichment for Building Information Modeling,
- Berners-Lee, T, Hendler, J. and Lassila, O. (2001) 'The Semantic Web: A New Form of Web Content That is Meaningful to Computers Will Unleash a Revolution of New Possibilities', *ScientificAmerican.com*. Scientific American Magazine. pp. 34–43.
- Beroe Inc., (2021). *Sustainable Building Materials Is a New Trend in Global Construction Industry, says Beroe Inc.* [Online] Available at: <https://www.prnewswire.com/news-releases/sustainable-building-materials-is-a-new-trend-in-global-construction-industry-says-beroe-inc-301352931.html> [Accessed August 2021].
- BCA. (2023). *Green Mark Certification Scheme*. Retrieved from Building and Construction Authority: <https://www1.bca.gov.sg/buildsg/sustainability/green-mark-certification-scheme> (Accessed January 1, 2023)
- Bhandari, B., Sunyong L., Chul-Ki, S., Maskey, R. and Ahn, H. (2014), 'A novel off-grid hybrid power system comprised of solar photovoltaic, wind, and hydro energy sources', *Applied Energy*, vol. 133, pp. 236–242,
- Bharadwaj, P.; John, V. Subcell (2019), 'Modeling of Partially Shaded Photovoltaic Modules'. *Trans. Ind. Appl.* 2019, 55, 3046–3054.
- Bilal, O. B., Sambou, V., Ndiaye, P. A., Kebe, CM F. and Ndong, M. (2010), 'Optimal design of a hybrid solar-wind-battery system using the minimization of the annualized cost system and the minimization of the loss of power supply probability (LPSP)', *Renewable Energy*, vol. 35, issue 10, 2010, pp. 2388-2390.
- Bing, L., Carr, L., Chan, K. (2014) 'Using Aligned Ontology Model to Convert Cultural Heritage Resources into Semantic Web', *IEEE International Conference on Semantic Computing*, pp. 120 – 123.
- Bloch, T., & Sacks, R. (2018). Comparing Machine Learning and Rule-Based Inferencing for Semantic Enrichment of BIM Models. *Automation in Construction*, 91, 256-272.
- Bloomfield, J., & Fisher, M. (2019). Quantitative research design. *Journal of the Australasian Rehabilitation Nurses' Association*, 22, 27–30. <https://doi.org/10.33235/jarna.22.2.27-30>
- Boje, C., Guerriero, A., Kubicki, S., & Rezgui, Y. (2020). Towards a semantic Construction Digital Twin: Directions for future research. *Automation in Construction*, 114.
- Bonduel, M.; Oraskari, J., Pauwels, P., Vergauwen, M., and Klein, R. (2018) 'The IFC to linked building data converter - Current status', *CEUR Workshop Proceedings*, 2159, pp. 34–43.
- Borgstede, M. and Scholz, M. (2021) 'Quantitative and Qualitative Approaches to Generalization and Replication—A Representationalist View', *Frontiers in Psychology*, 12. doi: 10.3389/fpsyg.2021.605191.
- British Standards Institution, (2022). 'What is ISO 19650?'. <https://www.bsigroup.com/en-GB/iso-19650-BIM/>
- BRE, (2022a). 'Background to the Green Guide to Specification', <https://tools.bregroup.com/greenguide/page.jsp?id=2069>
- BRE, (2022b). 'Is it Really Green? Many Green Claims for Building Materials are Made, but How Can You Be Sure of Their Accuracy', <https://www.greenbooklive.com/atoz.jsp>
- BuildingSMART, (2022a). 'IFC Formats'. [Online] Available at: <https://technical.buildingsmart.org/standards/ifc/ifc-formats/> [Accessed 20 June 2022].
- BuildingSMART, (2022b). 'IFC Schema Specification Database'. [Online] <https://standards.buildingsmart.org/IFC/RELEASE/IFC2x3/TC1/HTML/ifcmeasureresource/ifcmeasureresource.htm> [Accessed 20 June 2022].
- BuildingSMART, (2022c). 'Model View Definition (MVD)' - An Introduction. [Online] Available at: <https://technical.buildingsmart.org/standards/ifc/mvd/> [Accessed 20 June 2022].
- BuildingSMART, (2022d). 'IFC Solar Device'. [Online] Available at: <https://standards.buildingsmart.org/IFC/RELEASE/IFC4/ADD1/HTML/link/ifcrelationship.htm> [Accessed 20 June 2022].
- BuildingSMART, (2022e). 'IFC Certified Software', <https://www.buildingsmart.org/compliance/redevelopment/certified-software/> [Accessed 20 June 2022]

- BuildingSMART, (2022f), 'IfcObjectDefintion', [Online] Available at: <https://standards.buildingsmart.org/IFC/RELEASE/IFC4/ADD1/HTML/link/ifcrelationship.htm> [Accessed 20 June 2022]
- BuildingSMART, (2022g), 'IfcRelationship', [Online] Available at: <https://standards.buildingsmart.org/IFC/RELEASE/IFC4/ADD1/HTML/link/ifcrelationship.htm> [Accessed 20 June 2022]
- BuildingSMART, (2022h), 'IfcPropertyDefintion', [Online] Available at: <https://standards.buildingsmart.org/IFC/RELEASE/IFC4/ADD1/HTML/link/ifcrelationship.htm> [Accessed 20 June 2022]
- BuildingSmart, (2022i), 'IfcActor', [Online] Available at: https://standards.buildingsmart.org/IFC/DEV/IFC4_3/RC2/HTML/schema/ifckernel/lexical/ifcactor.htm [Accessed 20 June 2022]
- Burch, P. and Heinrich, C. (2015), 'Mixed Methods for Policy Research and Program Evaluation, Sage, Thousand Oaks CA.
- Buoncompagni, L., Kareem, S. Y. and Mastrogiovanni, F. (2022) 'OWLOOP: A modular API to describe OWL axioms in OOP objects hierarchies', *SoftwareX*, 17, p. 100952. doi: <https://doi.org/10.1016/j.softx.2021.100952>.
- Campo-Avila, J.D.; Piliouguine, M.; Morales-Bueno, R.; Mora-Lopez, L. (2019) A Data Mining System for Predicting Solar Global Spectral Irradiance. Performance Assessment in the Spectral Response Ranges of Thin-Film Photovoltaic Modules. *Renew. Energy* 2019, 133, 828–839.
- Cao, Y., Kamaruzzaman, S. N. and Aziz, N. (2022) 'Green Building Construction: A Systematic Review of BIM Utilisation', *Buildings*, 12. doi: 10.3390/buildings12081205.
- CarbonTrust. (2022). *Carbon Trust*. Retrieved from Carbon Trust: <https://www.carbontrust.com> (Accessed January, 2022)
- Cha, H. S. and Jiang, S. (2020) 'Special issue on BIM in the construction industry', *Applied Sciences (Switzerland)*, 10(12), pp. 3–5. doi: 10.3390/app10124306.
- Chibuko, U. *et al.* (2023) 'Module-level direct coupling in PV-battery power unit under realistic irradiance and load', *Solar Energy*, 249, pp. 233–241. doi: <https://doi.org/10.1016/j.solener.2022.11.040>.
- Chipman, T., Liebich, T. & Weise, M., (2016). *Model Support Group mvdXML*. [Online] Available at: https://standards.buildingsmart.org/MVD/RELEASE/mvdXML/v11/mvdXML_V1-1_Final.pdf [Accessed October 2020].
- Clark and Parsia, (2015). <http://clarkparsia.com/pellet>.
- Clemente, J. *et al.* (2022) 'A proposal for an adaptive Recommender System based on competences and ontologies', *Expert Systems with Applications*, 208, p. 118171. doi: <https://doi.org/10.1016/j.eswa.2022.118171>.
- Climate Change Committee, (2020). *The Sixth Carbon Budget Buildings*, s.l.: Climate Change Committee.
- Commission, E. (2019), Photovoltaic Geographical Information System. 2019. Available online: https://re.jrc.ec.europa.eu/pvg_tools/en/tools.html (accessed on 25 December 2019).
- CommonKADS. (2020). *Knowledge Model Templates*. Retrieved from CommonKADS: <https://commonkads.org/knowledge-model-templates/> (24 Accessed December 2020)
- CommonKADS. (2021). Retrieved January 2021, from <https://commonkads.org> (Accessed January 11 2021)
- Conceicao, R.; Silva, H.G.; Fialho, L.; Lopes, F.M.; Collares-Pereira, M. (2019), 'PV System Design with the Effect of Soiling on the Optimum Tilt Angle. *Renew. Energy* 2019, 133, 787–796.
- Construction, M.H., 2021. [Online] Available at: http://mddb.apec.org/Documents/2013/SCSC/WKSP5/13_scsc_wksp5_007.pdf
- Construction Industry Council, (2018). 'Building Information Modelling (BIM) Protocol Second Edition'. <https://www.cic.org.uk/uploads/files/old/bim-protocol2nd-edition-1.pdf>
- Cooke, N. J. (1994), 'Varieties of knowledge elicitation techniques', *International Journal of Human-Computer Studies*, 41(6), 801-849.
- Cooke, N. J. (1999), 'Knowledge Elicitation', In F. T. Durso, R. S. Nickerson, R. W. Schvaneveldt, S. T. Dumais, D. S. Lindsay & M. T. H. Chi (Eds.), *Handbook of Applied Cognition* (pp. 479-510). John Wiley & Sons, Ltd, Chichester, England, UK.
- Coombs, H. (2022) *Case Study Research Defined [White Paper]*. doi: 10.5281/zenodo.7604301.
- Corbin, J. and Strauss, A. L. (2015), 'Basics of Qualitative Research, Sage, Thousand Oaks CA.
- Corcho, O., Fernandez-Lopez, M., Gomes-Perez, A. (2003), 'Methodologies, tools and languages for building ontologies. Where is their meeting point?', *Data & Knowledge Engineering* 46 (2003) 41–64, Elsevier.

- Corcho, O., Fernández-lópez, M., Gómez-pérez, A., and López, A. (2005), 'Building legal ontologies with METHONTOLOGY and WebODE', V. R. Benjamins, P. Casanovas, J. Breuker, A. Gangemi (Eds.): Law and the Semantic Web, LNCS volume 3369, Springer-Verlag, Heidelberg, pp. 142-157.
- Costa, G. *et al.* (2020) 'A catalogue of energy conservation measures (ECM) and a tool for their application in energy simulation models', *Journal of Building Engineering*, 29(December 2019). doi: 10.1016/j.jobe.2019.101102.
- Courtot, M., Malone, J., & Mungall, C. (2016). Ten Simple Rules for Biomedical Ontology Development. In *ICBO/BioCreative*.
- Crewit. (2022). *Construction Industry Employment Statistics 2022*. Retrieved February 2023, from Crewit Resourcing: <https://www.crewitresourcing.com/2022/11/03/construction-industry-employment-statistics-2022> (Accessed November 3, 2022)
- Culha, A. (2023) 'Inclusive Leadership Practices in Schools: A Mixed Methods Study', 10, pp. 70–84.
- Dag, H.I.; Buker, M.S. (2020), Performance Evaluation and Degradation Assessment of Crystalline Silicon based Photovoltaic Rooftop Technologies under Outdoor Conditions. *Renew. Energy* 2020, 156, 1292–1300.
- David, M.; Lauret, P.; Boland, J. (2013) Evaluating tilted plane models for solar radiation using comprehensive testing procedures, at a southern hemisphere location. *Renew. Energy* 2013, 51, 124–131.
- Day-Richter, J. Harris, M. A. Haendel, M. (2007), Gene Ontology OBO-Edit Working Group, and Suzanna Lewis. Obo-edit—an ontology editor for biologists *Bioinformatics*, 23 (16) pp. 2198-2200
- Dhimish, M.; Holmes, V.; Mather, P.; Sibley, M. (2018), Preliminary assessment of the solar resource in the United Kingdom. *Clean Energy*, 2, 112–125.
- Ding, L. Y., Zhong, B. T., Wu, S., & Luo, H. B. (2016). Construction Risk Knowledge Management in BIM Using Ontology and Semantic Web Technology. *Saf Science*, 87, 202-213.
- Donev, J.; Yyelland, B.; Stenhouse, K.; Hanania, J.; Afework, B.; (2018) Energy Education. Types of Photovoltaic Cells. 2018. Available online:<http://www.nef.org.uk/knowledge-hub/solar-energy/types-of-photovoltaic-pv-cells> (accessed on 17 November 2019).
- Dutta, S. (2019), Plane of Array Vs Global Horizontal Irradiance. Solar PV Analytics. Available online: <https://solarpvanalytics.com/plane-of-array-vs-global-horizontal-irradiance/#respond> (accessed on 19 November 2019).
- Ecobin, (2021). *Worst GreenHouse Gases*. [Online] Available at: <https://www.ecobin.com.au/worst-greenhouse-gases/> [Accessed October 2021].
- EIA. (2023). *Environment*. Retrieved from U.S. Energy Information Administration: https://www.eia.gov/totalenergy/data/monthly/pdf/sec11_n.pdf (Accessed February 9th 2023)
- EnergyInformative (2019a). Amorphous Silicon Solar Panels. Energy Informative. Available online: <https://energyinformative.org/amorphous-silicon-solar-panels/> (accessed on 27 December 2019).
- EnergyInformative (2019b). Which Solar Panel Type is Best? Mono- vs. Polycrystalline vs. Thin Film. Energy Informative. Available online: <https://energyinformative.org/best-solar-panel-monocrystalline-polycrystalline-thin-film/> (accessed on 27 December 2019).
- Energy Saving Trust (2015). 'Government Grants and Supports', www.energysavingtrust.org.uk/grants-and-support
- Energy Savings Trust. (2021). *Impact Report*. Retrieved from Energy Savings Trust: <https://energysavingtrust.org.uk/wp-content/uploads/2021/08/Energy-Saving-Trust-Impact-Report-2020-21.pdf>
- EnergySavingsTrust (2022), 'Generating Renewable Electricity', <https://energysavingtrust.org.uk/energy-at-home/generating-renewable-electricity/>
- Energy Savings Trust (2023), 'Solar Panels', <https://energysavingtrust.org.uk/advice/solar-panels/> Retrieved February 2023.
- Engener, N. Solcast. (2019). Available online: <https://articles.solcast.com.au/en/articles/2404457-free-access-to-solar-irradiance-data-for-non-commercial-research-use> (accessed on 25 December 2019).
- Erdmann M., Waterfeld W. (2012), Overview of the neon toolkit Ontology Engineering in a Networked World, Springer (2012), pp. 281-301

- Fadaee, M. and Radzi, M. A. M (2012), 'Multi-objective optimization of a stand-alone hybrid renewable energy system by using evolutionary algorithms: A review,' *Renewable and Sustainable Energy Reviews*, vol. 16,2012, pp. 3364- 3369.
- Fargnoli, M. *et al.* (2019) 'A BIM-based PSS Approach for the Management of Maintenance Operations of Building Equipment', *Buildings*, 9(6). doi: 10.3390/buildings9060139.
- Fargnoli, M. and Lombardi, M. (2020) 'Building information modelling (BIM) to enhance occupational safety in construction activities: Research trends emerging from one decade of studies', *Buildings*, 10(6). doi: 10.3390/BUILDINGS10060098.
- Ferdosi, H. *et al.* (2022) 'BIM applications in sustainable construction: scientometric and state-of-the-art review', *International Journal of Construction Management*, 0(0), pp. 1–13. doi: 10.1080/15623599.2022.2029679.
- Fernandez, L. M. (2000), 'Overview of Methodologies for Building Ontologies',
- Fernández-Izquierdo, A. and García-Castro, R. (2021) 'Ontology verification testing using lexico-syntactic patterns', *Information Sciences*, 582, pp. 89–113. doi: <https://doi.org/10.1016/j.ins.2021.09.011>.
- Daniele, L. den Hartog, F. Roes. J. (2015), 'Created in Close Interaction with the Industry: the Smart Appliances REFERENCE SAREF Ontology (2015)
- Fedkin, M. (2020). *Sustainability Definitions*. Retrieved February 2023, from Technologies for Sustainability Systems.
- de Fine Licht, K. and Folland, A. (2019) 'Defining "Social Sustainability": Towards a Sustainable Solution to the Conceptual Confusion', *Etikk i praksis - Nordic Journal of Applied Ethics*, 13, pp. 21–39. doi: 10.5324/eip.v13i2.2913.
- Füssl FF, Streitferdt D, Triebel A. Modeling (2015), Knowledge bases for automated decision-making systems—a literature review. *Int J Adv Comput Sci Appl*. 2015;6(9):185-189.
- Gasparin, F. P., Kipper, F. D. de Oliveira, F. S. and Krenzinger, A. (2022) 'Assessment on the variation of temperature coefficients of photovoltaic modules with solar irradiance', *Solar Energy*, 244, pp. 126–133. doi: <https://doi.org/10.1016/j.solener.2022.08.052>.
- Gao, X. *et al.* (2023) 'Divide and Conquer Q-Learning (DCQL) algorithm based Photovoltaic (PV) array reconfiguration scheme for alleviating the partial shading influence', *Solar Energy*, 249, pp. 21–39. doi: <https://doi.org/10.1016/j.solener.2022.09.005>.
- Gayathri R, Uma V. (2018), Ontology based knowledge representation technique, domain modelling languages and planners for robotic path planning: a survey. *ICT Express*. 2018;4(2):69-74.
- Geng, Q. *et al.* (2020) 'Cross-domain ontology construction and alignment from online customer product reviews', *Information Sciences*, 531, pp. 47–67. doi: 10.1016/j.ins.2020.03.058.
- Gehry, F., (2021). *Advantages & Disadvantages of Green Building*. [Online] Available at: <https://environmental-conscience.com/green-building-pros-cons/> [Accessed October 2021].
- Gilani, S., Quinn, C. and McArthur, J. J. (2020) 'A review of ontologies within the domain of smart and ongoing commissioning', *Building and Environment*, 182, p. 107099. doi: <https://doi.org/10.1016/j.buildenv.2020.107099>.
- GOV.UK, (2021a). *Guidance Waste: environmental permits*. [Online] Available at: <https://www.gov.uk/guidance/waste-environmental-permits#history> [Accessed October 2021].
- GOV.UK, (2021b). *UK enshrines new target in law to slash emissions by 78% by 2035*. [Online] Available at: <https://www.gov.uk/government/news/uk-enshrines-new-target-in-law-to-slash-emissions-by-78-by-2035> [Accessed August 2021].
- Gomez-Perez, A., (1999). *Evaluation of Taxonomic Knowledge on Ontologies and Knowledge- Based Systems*. s.l., KAW.
- González, E. *et al.* (2020) 'An approach based on the ifcOWL ontology to support indoor navigation', *Egyptian Informatics Journal*, (xxxx). doi: 10.1016/j.eij.2020.02.008.
- Gougui, A.; Djafour, A.; Danoune, M.B.; Khelifaoui, N. (2019), Experimental Evaluation of a Photovoltaic System to Feed an AC Load. In Proceedings of the International Conference on Communications and Electrical Engineering (ICCEE), EL Oued, Algeria, 17–18 December 2019.
- Govil, N. and Sharma, A. (2022) 'Validation of agile methodology as ideal software development process using Fuzzy-TOPSIS method', *Advances in Engineering Software*, 168, p. 103125. doi: <https://doi.org/10.1016/j.advengsoft.2022.103125>.
- Graphisoft, (2021). *Graphisoft*. [Online] Available at: <https://graphisoft.com/case-studies/solve-collaboration-challenge-with-revu-and-archicad> [Accessed 9 March 2021].

- GreenBookLive. (2020). *Green Book Live*. Retrieved December 2020, from GreenBookLive: <https://www.greenbooklive.com> (Accessed December 28 2020)
- GreenGlobes. (2021). *A Practical Approach to Green Building*. Retrieved from Green Globes: <http://www.greenglobes.com/home.asp> (Accessed January 12 2021)
- Gruber, T. R. (1993) 'A translation approach to portable ontology specifications', *Knowledge Acquisition*, 5(2), pp. 199–220. doi: <https://doi.org/10.1006/knac.1993.1008>.
- Grüniger, M. and Fox, M. (1995), 'Methodology for the Design and Evaluation of Ontologies', in IJCAI Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada.
- Grzybek, H., Xu, S., Gulliver, S. and Fillingham, V. (2014), 'Considering the Feasibility of Semantic Model Design in the Built-Environment', *Buildings* 2014, 4, 849-879; doi:10.3390/buildings4040849. www.mdpi.com/journal/buildings/
- Guillén DS, Maceda JG. (2014), Ontology based inferences engine for veterinary diagnosis. In: Joint International Semantic Technology Conference. Cham: Springer; 2014:79-86.
- Gulhane, N. P. and Chavan, K. V. (2014), 'A critical technological literature review of - Photovoltaic / Thermal Solar Collectors System', *International Journal of Current Engineering and Technology*, ISSN 2277 – 4106.
- Haghir, S. *et al.* (2021) 'BIM based decision-support tool for automating design to fabrication process of freeform lattice space structure', *International Journal of Space Structures*, 36(3), pp. 164–179. doi: 10.1177/09560599211033867.
- Helal, M.A.; Anderson, N.; Wei, Y.; Thompson, M. (2023). 'A Review of Biomass-to-Bioenergy Supply Chain Research Using Bibliometric Analysis and Visualization. *Energies* 2023, 16, 1187. <https://doi.org/10.3390/en16031187>
- Hall, W., and O'Hara, K. (2009), 'Semantic Web', In, Robert A. Meyers (ed.), *Encyclopaedia of Complexity and Systems Science*, Springer.
- Hameed, H. (2020). Quantitative and qualitative research methods: Considerations and issues in qualitative research. <https://doi.org/10.13140/RG.2.2.36026.82883>
- Hamil, S. (2021). 'What is BIM', National Building Specification (NBS). <https://www.thenbs.com/knowledge/what-is-building-information-modelling-bim>
- Harmelen, F., Hendler, J., Horrocks, I., Lassila, O., McGuinness, D. (2000), 'The semantic web and its languages', *IEEE Intelligent Systems*, pp. 67-73.
- Hartley, G. (2015), 'Maintaining the UK's Solar Success Story', Energy Saving Trust', www.energysavingtrust.org.uk/maintaining-uks-solar-success-story
- Hassan, H.A.; Rahoma, U.A.; Elminir, H.K.; Fathy, A.M. (2015) Effect of Airborne Dust Concentration on the Performance of PV modules. *J. Aston*. 2005, 13, 124–138.
- Hawkins, A. (2023), 'Solar Panel Costs'. <https://www.architecturaldigest.com/reviews/solar/solar-panel-costs#:~:text=Types%20of%20Solar%20Panels&text=However%2C%20they%20are%20also%20the, costs%20between%20%246%2C000%20and%20%249%2C000> (Accessed July, 2023).
- Hay, J.E. (1979) Calculation of monthly mean solar radiation for horizontal and inclined surfaces. *Sol. Energy* 1979, 23, 301–307.
- Hendler, J., & Berners-Lee, T. (2010). From the Semantic Web to social machines: A research challenge for AI on the World Wide Web. *Artificial Intelligence*, 174, 156-161.
- Horrocks, I., Krötzsch, M., Simančík, F. (2013), 'A Description Logic Primer'.
- Hermit, (2015) <http://hermit-reasoner.com/>
- Hernandez-Moro, J. & Martinez-Duart, J. (2013), 'Analytical model for solar PV and CSP electricity costs: Present LCOE values and their future evolution', *Renewable and Sustainable Energy Reviews* 20, 119-132.
- Hippolyte, J. L. *et al.* (2018) 'Ontology-driven development of web services to support district energy applications', *Automation in Construction*, 86(October 2017), pp. 210–225. doi: 10.1016/j.autcon.2017.10.004.
- Hirth, L. (2015a), 'The Market Value of Solar Power: Is Photovoltaics Cost-Competitive', *IET Renewable Power Generation* 9(1).
- Hirth, L., Ueckerdt, F. & Edenhofer, O. (2015b): "Integration Costs Revisited – An economic framework of wind and solar variability", *Renewable Energy* 74, 925–939.
- Hlomani, H. & Stacey, D., 2014. Approaches, methods, metrics, measures, and subjectivity in ontology evaluation: A survey. *Semantic Web Journal*, Volume 1, pp. 1-5.
- Hontl, G. M., & Abonyi, J. (2019, June 12). A Review of Semantic Sensor Technologies in Internet of Things Architectures. *Complexity*.

- Horrocks, I. (2008), 'Ontologies and the semantic web', *Communications of the ACM*, pp. 58-67.
- Horrocks, I. (2011). 'Tool Support for Ontology Engineering', In Dieter Fensel, editor, *Foundations for the Web of Information and Services*, pages 103-112. Springer.
- Horrocks, I. Krötzsch, M., Simančík, F. (2013), 'A Description Logic Primer'.
- Horrocks, I., Broekstra J., Klein, M., Decker, S., Fensel, D., Harmelen, F. (2001), 'Enabling knowledge representation on the Web by Extending RDF Schema', In *Proc. of the Tenth International World Wide Web Conference (WWW 2001)*, pp467-478.
- Horrocks, I., Glimm, B., Motik, B., Stoilos, G., Wang, Z. (2014a). 'Hermit: An OWL 2 Reasoner', *Journal of Automated Reasoning*, pp. 245-269.
- Horrocks, I., Motik, B., Nenov, Y., Piro, R., and Olteanu, D. (2014b), 'Parallel OWL 2 RL Materialisation in Centralized, Main-Memory RDF Systems', *Informal Proceedings of the 27th International Workshop on Description Logics*, Vienna, Austria, volume 1193 of *CEUR Workshop Proceedings*, pages 311-323. CEUR-WS.org.
- Hossain, S., Hossain, M. M., Reazul, ARNM (2011), 'Optimization and Modeling of a Hybrid Energy System for Off-Grid Electrification' 2011 IEEE
- Hosseini, M. R., Roelvink, R., Papadonikolaki, E., and Edwards, D. J. (2018) 'Integrating BIM into facility management: Typology matrix of information handover requirements', *International Journal of Building Pathology and Adaptation*, 36(1), pp. 2-14. doi: 10.1108/IJBPA-08-2017-0034.
- Huang, Z., Jowers, C., Dehghan-Manshadi, A. & Dargusch, M. S., 2020. Smart Manufacturing and DVSM Based on an Ontological Approach. *Computers in Industry*, 3 February.117(103189).
- Hu, S., Wang, J., Hoare, C., and Li, Y. (2021), 'Building energy performance assessment using linked data and cross-domain semantic reasoning', *Automation in Construction*, 124, p. 103580. doi: <https://doi.org/10.1016/j.autcon.2021.103580>.
- Hussain, C. M., Paulraj, M. S. and Nuzhat, S. (2022) 'Chapter 2 - Source reduction, waste minimization, and cleaner technologies', in Hussain, C. M., Paulraj, M. S., and Nuzhat, S. B. T-S. R. and W. M. (eds). Elsevier, pp. 23-59. doi: <https://doi.org/10.1016/B978-0-12-824320-6.00002-2>.
- El Idrissi, B. et al. (2022) 'RDF/OWL storage and management in relational database management systems: A comparative study', *Journal of King Saud University - Computer and Information Sciences*, 34(9), pp. 7604-7620. doi: <https://doi.org/10.1016/j.jksuci.2021.08.018>.
- IEA (2019). Available online: <https://www.iea.org/renewables2019/> (accessed on 21 October 2019).
- IEA, (2021). *Solar PV*. [Online] Available at: <https://www.iea.org/reports/solar-pv> [Accessed August 2021].
- IEA, (2022a). Global CO2 Emissions Rebounded to their Highest Level in History in 2021, <https://www.iea.org/news/global-co2-emissions-rebounded-to-their-highest-level-in-history-in-2021#:~:text=CO2%20emissions%20in%20the%20United,less%20than%201%25%20in%202021> (Accessed March 08, 2022).
- IEA. (2022b). *Renewable Energy Market Update*. Retrieved from <https://www.iea.org/reports/renewable-energy-market-update-may-2022>. (Accessed May 2022).
- IEA. (2023). *CO2 Emissions in 2022*. Retrieved from International Energy Agency: <https://www.iea.org/reports/co2-emissions-in-2022> (Accessed March 2023)
- IEA. (2023b). *Buildings*. Retrieved from IEA: <https://www.iea.org/reports/buildings>. Retrieved February 2023
- IfcOpenShell, 2020. *ifcOpenShell*. [Online] Available at: <http://ifcopenshell.org>[Accessed March 2021].
- IEEE (2018). IEEE Draft Guide for Array and Battery Sizing in Stand-Alone Photovoltaic (PV) Systems. 2018. Available online: <http://ieeexplore.ieee.org/oxfordbrookes.idm.oclc.org/stamp/stamp.jsp?tp=&arnumber=8315174&isnumber=8315173> (accessed on 15 January 2020).
- IHA. (2020). *Six emerging hydropower technology trends*. Retrieved February 2023, from International Hydropower Association: <https://www.hydropower.org/blog/six-emerging-hydropower-technology-trends> (Accessed April 15, 2020)
- IHA. (2022). *Hydropower Status Report*. Retrieved 2023, from International Hydropower Association: https://assets-global.website-files.com/5f749e4b9399c80b5e421384/63a1d6be6c0c9d38e6ab0594_IHA202212-status-report-02.pdf
- Ingrams, S., (2021). *Solar Panel Brand Review*. [Online] Available at: https://www.which.co.uk/reviews/solar-panels/article/best-solar-panel-brands/solar-panel-brand-reviews-a1rAN1R8nYFD?gclid=Ci0KCQjwqpLBhDQARIsA00a6aK3UVdI6Hy4YHst1_ltoybbxnezSmG2iLLl87YldUYXFOqeLiFZn0aAmrCEALw_wcB&gclid=aw.ds[Accessed October 2021].

- Isiaq, S. and Osman, T. (2014), 'Ontology Modelling Methodology for Temporal and Interdependent Applications', 2014 UKSim-AMSS 16th International Conference on Computer Modeling and Simulation.
- Izquierdo, Y. T., Garcia, G., Menendez, S., Andre, L., Lemos, M., Finamore, A., Oliviera, C., and Casanova, M. (2021) 'Keyword search over schema-less RDF datasets by SPARQL query compilation', *Information Systems*, 102, p. 101814. doi: <https://doi.org/10.1016/j.is.2021.101814>.
- Jansson, P.M.; Whitten, K.; Schmalzel, J.L. (2011), Photovoltaic Module Shading: Smart Grid Impacts. In Proceedings of the IEEE Sensors Applications Symposium, San Antonio, TX, USA, 22–24 February 2011; pp. 323–328.
- Javed, K., Khan, A. H., & Tubbassum, L. (2019). Critical Analysis of Software Development Methodologies based on Project Risk Management. *International Journal of Academic Research in Business and Social Sciences*, 9(12), 253–263
- Jeong, Y. (2018). A Study on the BIM Evaluation, Analytics, and Prediction (EAP) Framework and Platform in Linked Building Ontologies and Reasoners with Clouds. *Advances in Civil Engineering*.
- Jenkins, D. (2015), 'Renewable Energy Systems; The EarthScan Expert Guide to Renewable Energy Technologies for Home and Business', Routledge. Date retrieved, August 2015.
- Jetlund, K., Onstein, E. & Huang, L., 2020. IFC Schemas in ISO/TC 211 Compliant UML for Improved Interoperability between BIM and GIS. *International Journal of Geo-Information*, 23 April.9(278).
- Jiménez, A., Suárez-Figueroa, M. C., Mateos, A., Fernández-López, M., Gómez-Pérez, A. (2012), 'A MAUT Approach for Reusing Ontologies', *IEEE 28th International Conference on Data Engineering Workshops*. Spain.
- Jojusolar, (2021). *A HUB OF HIGH EFFICIENCY AT OXFORD BROOKES UNIVERSITY*. [Online] Available at: https://www.jojusolar.co.uk/portfolio_page/a-hub-of-high-efficiency-at-oxford-brookes-university/ [Accessed 10 October 2021].
- Jordan, D.C.; Marion, B.; Deline, C.; Barnes, T.; Bolinger, M.(2020) PV Field Reliability Status–Analysis of 100,000 Solar Systems. *Prog. Photovolt.* 2020, 28, 739–754. *Buildings* 2020, 10, 192 19 of 22
- Jung, S., Cha, H. S. and Jiang, S. (2020) 'Developing a Building Fire Information Management System Based on 3D Object Visualisation', *Applied Sciences*, 10(3). doi: 10.3390/app10030772.
- Kalay, M. Ş., Kiliç, B. and Sağlam, Ş. (2022) 'Systematic review of the data acquisition and monitoring systems of photovoltaic panels and arrays', *Solar Energy*, 244, pp. 47–64. doi: <https://doi.org/10.1016/j.solener.2022.08.029>.
- Kalemi, E. V. et al., (2020). *ifcOWL-DfMA a New Ontology for the Offsite Construction Domain*. s.l., s.n., pp. 105-117.
- Kalyanpur, A., Parsia, B., Sirin, E., Cuenca Grau, B., Hendler, J. (2006), Swoop: a web ontology editing browser *J Web Semant*, 4 (2) pp. 144-153
- Kantamneni, A. and Brown, L. (2018) *An Ontology for Solar Irradiation Forecast Models*. doi: 10.5220/0006937202630270.
- Kazem, H. A., Chaichan, M. T., Al-Waeli, A. H. A., and Sopian, K. (2020), 'A review of dust accumulation and cleaning methods for solar photovoltaic systems', *Journal of Cleaner Production*, 276, p. 123187. doi: <https://doi.org/10.1016/j.jclepro.2020.123187>.
- Kazem, H. A., Chaichan, M. T., Al-Waeli, Al-Badi, R., Fayad, M. A. and Golami, A. (2022) 'Dust impact on photovoltaic/thermal system in harsh weather conditions', *Solar Energy*, 245, pp. 308–321. doi: <https://doi.org/10.1016/j.solener.2022.09.012>.
- Keet, C. M. (2018). *An Introduction to Ontology Engineering*. Cape Town: Independent. <https://open.umn.edu/opentextbooks/textbooks/590> [Google Scholar](#)
- Kent, L., Snider, C., Gopsil, J., and Hicks, B. (2021) 'Mixed reality in design prototyping: A systematic review', *Design Studies*, 77, p. 101046. doi: <https://doi.org/10.1016/j.destud.2021.101046>.
- Khaertdinova, A., Maliashova, A., & Gadelshina, S. (2021). Economic development of the construction industry as a basis for sustainable development of the country. *E3S Web of Conference*, 10021.
- Khalid, M.; Janjua, A.K.; Khalid, H.A. (2019), Effect of PV Panel Orientation on Batteries in a Solar Generation System. In Proceedings of the International Conference on Power Generation Systems and Renewable Energy Technologies (PGSRET), Islamabad, Pakistan, 10–12 September 2019.
- Khan, M. I., Asfand, F., & Al-Ghamdi, S. G. (2023). Progress in research and technological advancements of commercial concentrated solar thermal power plants. *Solar Energy*, 249, 183-226.
- Khan, M. R. B., Jidin, R., Pasupuleti, J., Shaaya, S. A. (2014), 'Optimal Combinations of PV, Wind, Micro-Hydro and Diesel Systems for a Seasonal Load Demand, Malaysia.
- Khan, N. et al. (2020) 'Visual Language-Aided Construction Fire Safety Planning Approach in Building Information Modeling', *Applied Sciences*, 10(5). doi: 10.3390/app10051704.

- Khankasikam, K. (2010), 'Knowledge capture for Thai word segmentation by using CommonKADS', *Computer and Automation Engineering (ICCAE)*, 2010 The 2nd International Conference on (Volume: 1), 978-1-4244-5586-7, Pages 307 – 311.
- Khosrojerdi, F., Gagnon, S., & Valverde, R. (2021). settings Order Article Reprints Open Access Article Proposing an Ontology Model for Planning Photovoltaic Systems . *Machine Learning Knowledge Extraction*, 3, 582-600.
- Kibert, C. J. (1994), "Preface. In Proceedings of First International Conference of CIB TG 16 on Sustainable Construction", Tampa, Florida.
- Kingston, J., Shadbolt, N., and Tate, A. (1996), 'CommonKADS Models for Knowledge-Based Planning', American Association for Artificial Intelligence.
- Kofler, M. J., Reinisch, C. and Kastner, W. (2012) 'A semantic representation of energy-related information in future smart homes', *Energy and Buildings*, 47, pp. 169–179. doi: <https://doi.org/10.1016/j.enbuild.2011.11.044>.
- Kotis, K. I., Vouros, G. A. and Spiliotopoulos, D. (2020) 'Ontology engineering methodologies for the evolution of living and reused ontologies: status, trends, findings and recommendations', *The Knowledge Engineering Review*. 2020/01/31, 35, p. e4. doi: DOI: 10.1017/S0269888920000065.
- Koutroulis, E. and Kolokotsa, D. (2010), 'Design optimization of desalination systems power-supplied by PV and WIG energy sources', *Desalination*, vol. 258, 2010, pp. 171-181.
- Krijnen, T., & Beetz, J. (2018). A SPARQL Query Rngine for Binary-Formatted IFC Building Models. *Automation in Cosntructio*, 95, 46- 63.
- Krijnen, T., VanBerlo, L. & Zentsev, F., (2020). *buildingSMART / NextGen-IFC*. [Online] Available at: <https://github.com/buildingSMART/NextGen-IFC>[Accessed March 2021].
- Küçük, Dilek and Küçük, Doğan (2018) *OntoWind: An Improved and Extended Wind Energy Ontology*.
- Kumaresh, V., Malhotra, M., Ramakrishna, N. and Prabu, R. (2014), 'Literature Review on Solar MPPT Systems', *Advance in Electronic and Electric Engineering*. ISSN 2231-1297, Volume 4, Number 3 (2014), pp. 285-296 © Research India Publications.
- Kurtz, S.; Newmiller, J.; Kimber, A.; FLOTtemesch, R.; Riley, E.; Dierauf, T.; McKee, J.; Krishnani, P. (2013) *Analysis of Photovoltaic System Energy Performance Evaluation Method*; National Renewable Energy Laboratory, Energy Efficiency & Renewable Energy. U.S. Department of Energy: Denver, CO, USA, 2013.
- Lawal, K. O. (2012), 'Hydro-based, Renewable Hybrid Energy System for Rural/Remote Electrification in Nigeria'.
- Lai, F. *et al.* (2023) 'Green building technologies in Southeast Asia: A review', *Sustainable Energy Technologies and Assessments*, 55, p. 102946. doi: <https://doi.org/10.1016/j.seta.2022.102946>.
- Lee, H. (2015), 'A Bottom-up Analysis of a High Concentration PV Modules', NREL.
- Lee, J. (2022) 'Advanced BIM Application in Construction and Buildings', *Buildings*, 12(8), pp. 12–14. doi: 10.3390/buildings12081148.
- Lee, S.-K., Kim, K.-R., & Yu, J.-H. (2014). BIM and Ontology-Based Approach for Building Cost Estimation. *Automation in Construction*, 41, 96-105.
- LEED. (2023). *LEED-certified green buildings are better buildings*. Retrieved from LEED: <https://www.usgbc.org/leed> (Accessed February 12 2023)
- Lefrançois, M. (2017), Planned ETSI SAREF Extensions based on the W3C&OGC SOSA/SSN-compatible SEAS Ontology Patterns. In Proceedings of Workshop on Semantic Interoperability and Standardization in the IoT, SIS-IoT, Amsterdam, Netherlands, July 2017.
- Lenzerini, M., Lepore, L. and Poggi, A. (2021) 'Metamodeling and metaquerying in OWL2QL', *Artificial Intelligence*, 292, p. 103432. doi: <https://doi.org/10.1016/j.artint.2020.103432>.
- Leonics (2013), 'How to Design Solar PV System', http://www.leonics.com/support/article2_12j/articles2_12j_en.php. Date viewed May 2015.
- Li, Z. and Huang, X. (2022) 'Simulation analysis on operation performance of a hybrid heat pump system integrating photovoltaic/thermal and air source', *Applied Thermal Engineering*, 200, p. 117693. doi: <https://doi.org/10.1016/j.applthermaleng.2021.117693>.
- Liao, B. and Li, L. (2022) 'How can green building development promote carbon emission reduction efficiency of the construction industry?—Based on the dual perspective of industry and space', *Environmental Science and Pollution Research*, 29(7), pp. 9852–9866. doi: 10.1007/s11356-021-16380-2.
- Liao, L. (2021) 'Research on the Application of BIM Technology in the Cost Management of Construction Projects', *IOP Conference Series: Earth and Environmental Science*, 783(1), p. 12098. doi: 10.1088/1755-1315/783/1/012098.

- Liu, G., Rasul, M. G., Amanullah, M. T. O., Khan, M. M. K. (2011), 'Feasibility study of stand-alone PV-wind-biomass hybrid energy system in Australia'
- Liu, G.; Xiao, X.; Qi, H.; Song, H.; Zhao, S.; Cao, D. (2019), Performance Evaluation of Photovoltaic System under Harsh Field Conditions. In Proceedings of the 8th IEEE International Symposium on Next-Generation Electronics, Zhengzhou, China, 9–10 October 2019.
- Liu S, Zarate P. (2014), Knowledge based decision support systems: a survey on technologies and application domains. In: Zarate P, Kersten GE, Hernandez JE, eds. Lecture Notes in Business Information Processing 180 LNBIP. Cham: Springer; 2014:62-72.
- Liu, Z.-J., Pyplacz, P., Ermakova, M., & Konev, P. (2020). Sustainable Construction as a Competitive Advantage. *Sustainable Construction Investments - Technical and Organisational Implications*, 12(15).
- Lo, S.L.Y.; How, B.S.; Leong, W.D.; Teng, S.Y.; Rhamdhani, M.A.; Sunarso, J. (2021), Techno-economic analysis for biomass supply chain: A state-of-the-art review. *Renew. Sustain. Energy Rev.* 2021, 135, 110164.
- Lu, H., He, B. and Zhao, W. (2023) 'Experimental study on the super-hydrophobic coating performance for solar photovoltaic modules at different wind directions', *Solar Energy*, 249, pp. 725–733. doi: <https://doi.org/10.1016/j.solener.2022.12.023>.
- Lu, W., Yuan, H. Li, J.J., Hao, J.L., Mi, X. and Ding, Z. (2010), 'An Empirical Investigation of Construction and Demolition Waste Generation Rates in Shenzhen City, South China', *Journal of Waste Management*.
- Ma, J.; Pan, X.; Man, K.L.; Li, X.; Wen, H.; Ting, T.O. (2018), Detection and Assessment of Partial Shading Scenarios on Photovoltaic Strings. *Trans. Ind. Appl.* 2018, 54, 6279–6289.
- Ma, T., Yang, H., Lu, L. and Peng, J. (2015), 'Pumped storage-based stand-alone photovoltaic power generation system: Modelling and techno-economic optimization', *Applied Energy*, vol. 137, pp. 649-659.
- Ma, Z., & Liu, Z. (2018). Ontology-and Freeware-Based Platform for Rapid Development of BIM Applications with Reasoning Support. *Automation in Construction*, 90, 1-8.
- Madrazo, L., Sicilia, A. and Gamboa, G. (2012) *SEMANCO: Semantic Tools for Carbon Reduction in Urban Planning*.
- Malek, H. (2014), 'Control of Grid-Connected Photovoltaic Systems Using Fractional Order Operators', A dissertation submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy, Utah State University.
- Marion, B. (2015), Deriving the Solar Resource Using PV Data; National Renewable Energy Laboratory (NREL) Photovoltaic Reliability Workshop: Golden, CO, USA, 2015.
- Martinez-Valencia, L.; Camenzind, D.; Wigmosta, M.; Garcia-Perez, M.; Wolcott, M. (2021), Biomass supply chain equipment for renewable fuels production: A review. *Biomass Bioenergy* 2021, 148, 106054.
- Martins PP, Sell D, Rotta MJR, Ortega AR. (2018), Applying the CommonKADS methodology in the implementation of e-gov projects in the perspective of a software company. *NAVUS—Rev Gestao Tecnol.* 2018;2018 8(2):87-100.
- Mattos, M. M., Heckmann, J. R. and Fernando da Silva, P. (2015), 'An Ontology to the Information Security Management', 2015 Ninth International Conference on Complex, Intelligent, and Software Intensive Systems, Brazil.
- Badger-Charleson, S., and McBeath, A. (2023) *Supporting Research in Counselling and Psychotherapy Qualitative, Quantitative, and Mixed Methods*.
- McGuinness, D. L., Fikes, R., Rice, J., and Wilder, S. (2000), An Environment for Merging and Testing Large Ontologies, Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR2000). Breckenridge, Colorado. April 12 - 15, 2000.
- Menezes, A.C.; Cripps, A.; Buswell, R.A.; Wright, J.; Bouchlaghem, D. (2014). Estimating the energy consumption and power demand of small power equipment in office buildings. *Energy Build.* 2014, 75, 199–209.
- Meteotest (2019). Meteororm Irradiation Data for every Place on Earth. 2019. Available online: <https://meteororm.com/en/> (accessed on 25 December 2019).
- Milton, N. (2012), 'Acquiring Knowledge from Subject Matter Experts, In J. Kantola & W. Karwowski (Eds.), Knowledge Service Engineering Handbook. CRC Press, Boca Raton, Florida, USA.
- Mirshokraei, M., De Gaetani, C. I. and Migliaccio, F. (2019) 'A Web-Based BIM-AR Quality Management System for Structural Elements', *Applied Sciences*, 9(19). doi: 10.3390/app9193984.
- Missimer, M., & Mesquita, P. L. (2022). Social Sustainability in Business Organisations: A Research Agenda. *Sustainability*, 14(2608).
- Mohajan, H. K. (2018). Qualitative Research Methodology in Social Sciences and Related Subjects. *Journal of Economic Development, Environment, and People*, 7(01), 23-48.

- Mollenkamp, D. T. (2023). *What is Sustainability? How Sustainabilities Work, Benefits, and Example*. Retrieved from Investopedia: <https://www.investopedia.com/terms/s/sustainability.asp>
- Mottaghi, M.; Bairamzadeh, S.; Pishyae, M.S. (2022), A taxonomic review and analysis on biomass supply chain design and planning: New trends, methodologies and applications. *Ind. Crops Prod.* 2022, 180, 114747.
- Morgan, D. L. (2014). *Integrating Qualitative and Quantitative Methods: A Pragmatic Approach*. SAGE.
- Moreira, J., Pire, L. F., vanSinderen, M. and Daniele, L. (2020) 'SAREF4health: Towards IoT standard-based ontology-driven cardiac e-health systems', *Applied Ontology*, 15, pp. 385–410. doi: 10.3233/AO-200232.
- Mudlapur, A.; Ramana, V.V.; Damodaran, R.V.; Balasubramanian, V.; Mishra, S. (2019) Effect of Partial Shading on PV Fed Induction Motor Water Pumping Systems. *Trans. Energy Convers.* 2019, 34, 530–539.
- Nejad, A. R., Keller, J. A., Guo, Yi., and Sheng, S. (2022) 'Wind turbine drivetrains: state-of-the-art technologies and future development trends', *Wind Energy Science*, 7(1), pp. 387–411. doi: 10.5194/wes-7-387-2022.
- NASA.(2019). OSMA Office of Safety and Mission Assurance. 2019. Available online: <https://sma.nasa.gov/>(accessed on 25 December 2019).
- NBS. (2020). *Uniclass 2015*. Retrieved December 2020, from NBS: <https://www.thenbs.com/our-tools/uniclass-2015> (Accessed December 28 2020)
- Nelson, B. (2010), 'Standards Can Take PV to Its Gold Medal Game,' *Future Photovoltaics*.
- De Nicola, A. & Missikoff, M. (2016). A lightweight methodology for rapid ontology engineering. *Communications of the ACM*, 59(3), 79–86. <https://doi.org/10.1145/2818359>.
- Nicholas-Institute. (2018, November 26). *What are data, information, and knowledge?* Retrieved from Internet of Water Coalition: <https://internetofwater.org/valuing-data/what-are-data-information-and-knowledge/>
- Niknam, M., Jalaei, F., & Karshenas, S. (2019). Integrating BIM and Product Manufacturer Data Using the Semantic Web Technologies. *Journal of Information Technology in Construction (ITcon)* , 24, 424-439.
- Nikiforova O, Kirikova M. (2004), Two-hemisphere model driven approach: engineering based software development. In: Persson A, Stirna J, eds. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol 3084. Berlin, Heidelberg: Springer; 2004:219-233.
- Nissinen, A., Seppala, J., & Heinonen, T. (2022). Make carbon footprints available – And it is not just one value. *Cleaner Logistics and Supply Chain*, 3(100023).
- NOAA. Total Solar Irradiance (TSI). National Centers for Environmental Information. 2019. Available online: <https://www.ngdc.noaa.gov/stp/solar/solarirrad.html#composite> (accessed on 25 December 2019).
- Notton, G.; Lazarov, V.; Stoyanov, L. (2009) Optimal Sizing of a Grid-Connected PV System for Various PV Module Technologies and Inclinations, Inverter Efficiency Characteristics and Locations. *Renew. Energy* 2009, 35, 541–554.
- Noy, N. F., & McGuinness, D. L. (2001). 'Ontology Development 101: a guide to creating your first ontology', Retrieved June 7, 2007, from <http://www-ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness.pdf>.
- Noy, N., Musen, M. (2014) 'PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment' National Conference on Artificial Intelligence.
- Noy, NF, Musen, M. (1999), 'SMART: Automated Support for Ontology Merging and Alignment', In Proceedings of the 12th Workshop on Knowledge Acquisition, Modelling and Management (KAW'99).
- Noy, NF, Musen, M. (2000), 'PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment', In Proceedings of the 17th National Conference on Artificial Intelligence (AAAI'00).
- Noy, NF, Musen, M. (2002), 'PROMPTDIFF: A Fixed-Point Algorithm for Comparing Ontology Versions', In Proceedings of the 18th National Conference on Artificial Intelligence (AAAI'02).
- NREL (2023), Quarterly Solar Industry Update, by the National Renewable energy Laboratory. April, 27, 2023 <https://www.energy.gov/eere/solar/quarterly-solar-industry-update> (Accessed July, 2023)
- Nrip, N., Ghorpade, O. and Musale, B. (2021) 'CURRENT STATE OF SEMANTIC WEB TECHNOLOGIES'.
- Ogunmakinde, O. E., Sher, W. D. & Maund, K., (2016). OBSTACLES TO SUSTAINABLE CONSTRUCTION IN DEVELOPING COUNTRIES. *Radical Innovation in the Built Environment*.

- Okay, K., Eray, S. and Eray, A. (2022) 'Development of prototype battery management system for PV system', *Renewable Energy*, 181, pp. 1294–1304. doi: <https://doi.org/10.1016/j.renene.2021.09.118>.
- Olawumi, T. O. and Chan, D. W. M. (2018) 'Identifying and prioritising the benefits of integrating BIM and sustainability practices in construction projects: A Delphi survey of international experts', *Sustainable Cities and Society*, 40, pp. 16–27. doi: <https://doi.org/10.1016/j.scs.2018.03.033>.
- Oliveira, A., Kohwalter, T. C., Kalinowski, M., and Murta, L. G. P. (2020) 'XChange: A semantic diff approach for XML documents', *Information Systems*, 94, p. 101610. doi: <https://doi.org/10.1016/j.is.2020.101610>.
- OntologySpecificationDraft, (2022), 'Digital Construction - Energy Systems', <https://digitalconstruction.github.io/EnergySystems/latest/#classes-headline>
- OpenEnergyFamily (2020), Open Energy Family - Open Energy Ontology (OEO) <https://github.com/OpenEnergyPlatform/ontology> (Accessed June 11th, 2020)
- Orlandi, F., Meehan, A., Hossari, M., Soumyabrata, D., O'Sullivan, D., & Alskaf, T. (2019). Interlinking Heterogeneous Data for Smart Energy Systems. *PARTicipatory platform for sustainable ENERGY management (PARENT)*.
- OECD, (2023), Organisation for Economic Cooperation and Development, Glossary of Statistical Terms. Available online: <https://stats.oecd.org/glossary/about.asp> (accessed on 12 January 2023).
- Oxford Brookes University, (2020). *Oxford Brookes University ranked as top tier university in sustainable energy*. [Online] Available at: <https://www.brookes.ac.uk/about-brookes/news/oxford-brookes-university-ranked-as-top-tier-university-in-sustainable-energy/>[Accessed 11 October]
- Palmetto, (2021). *How Much Roof Space Is Needed For Home Solar Panels?*. [Online] Available at: <https://palmetto.com/learning-center/blog/how-much-roof-space-is-needed-for-solar-panels>[Accessed September 2021].
- Panzarella, G., Veltri, P. and Alcaro, S. (2023) 'Using ontologies for life science text-based resource organisation', *Artificial Intelligence in the Life Sciences*, 3, p. 100059. doi: <https://doi.org/10.1016/j.aillsci.2023.100059>.
- Parsamehr, M., Perera, U. S., Dodanwala, T. C. and Kaluthantirige, P. P. (2022) 'A review of construction management challenges and BIM-based solutions: perspectives from the schedule, cost, quality, and safety management', *Asian Journal of Civil Engineering*. doi: 10.1007/s42107-022-00501-4.
- Parizi R., Prestes M., Marczak S., and Conte T. (2022), How has design thinking being used and integrated into software development activities? *Asystematic mapping*. *J Syst Softw* 2022:111217. <https://doi.org/10.1016/j.jss.2022.111217>.
- Park, C., Lee, D. Y., Kwon, O. S., & Wang, X. (2013). A Framework for Proactive Construction Defect Management Using BIM, Augmented Reality and Ontology-Based Data Collection Template. *Automation in Construction*, 33, 61-71.
- Patel, A. and Jain, S. (2019) 'Present and future of semantic web technologies: a research statement', *International Journal of Computers and Applications*, 43, pp. 1–10. doi: 10.1080/1206212X.2019.1570666.
- Patil, M., Boraste, S. and Minde, P. (2022) 'A comprehensive review on emerging trends in smart green building technologies and sustainable materials', *Materials Today: Proceedings*, 65, pp. 1813–1822. doi: <https://doi.org/10.1016/j.matpr.2022.04.866>.
- Pascal, H. (2021, January 25). A review of the semantic web field. *Communications of the ACM*, 64(2), 76-83.
- Pauwels, P., & Terkaj, W. (2016). EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. *Automation in Construction*, 63(March), 100–133. <https://doi.org/10.1016/j.autcon.2015.12.003>
- Pauwels, P., Krijnen, T., Terkaj, W. and Beetz, J.(2017) 'Enhancing the ifcOWL ontology with an alternative representation for geometric data', *Automation in Construction*, 80, pp. 77–94. doi: 10.1016/j.autcon.2017.03.001.
- Pauwels, P., (2018), Building Element Ontology, <https://pi.pauwel.be/voc/buildingelement/index-en.html> (Accessed January, 2021).
- Pauwels, P., (2021a). *IFCtoRDF*. [Online] Available at: <https://github.com/pipauwel/IFCtoRDF> [Accessed July 2021].

- Pauwels, P., (2021b). *EXPRESStoOWL*. [Online] Available at: <https://github.com/pipauwel/EXPRESStoOWL>
- Peroni, S., Palmirani, M. & Vitali, F. (2017). UNDO: The United Nations system document ontology. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10588 LNCS, 175–183. https://doi.org/10.1007/978-3-319-68204-4_18
- Petrova, E., Pauwels, P., Svidt, K., and Jensen, R. L. (2019) 'Semantic data mining and linked data for a recommender system in the AEC industry', *Proceedings of the 2019 European Conference on Computing in Construction*, 1, pp. 172–181. doi: 10.35490/ec3.2019.192.
- Pinto, S., Staab, S., and Tempich, C. (2004), 'DILIGENT: Towards a fine-grained methodology for Distributed Loosely-controlled and evolving Engineering of ontologies', R. L. de Mántaras, L. Saitta (Eds.): *Proceedings of the 16th European Conference on Artificial Intelligence*, IOS Press, pp 393-397.
- Pocsova, J., Bednarova, D., Bogdanovska, G., & Mojzisova, A. (2020, November 17). Implementation of Agile Methodologies in an Engineering Course. *Education Sciences*.
- Pra, L. B. D., Dias, J. B. and Kieling, A. G. (2015), 'Comparison between the Energy Required for Production of PV Module and the Output Energy Throughout the Product Life Time', *Journal of Energy and Power Engineering* 9 (2015) 592-597 D DOI: 10.17265/1934-8975/2015.06.011, David Publishing.
- Preventis, A. and Petrakis, E. G. M. (2021) 'CLONE: Collaborative Ontology Editor as a Service in the Cloud', *Procedia Computer Science*, 184, pp. 275–282. doi: <https://doi.org/10.1016/j.procs.2021.04.006>.
- Programcreek, (2023), 'Jena Ontology', <https://www.programcreek.com/java-api-examples/?api=com.hp.hpl.jena.ontology.OntClass>. (Accessed January, 2023).
- Protégé, (2015). Stanford University, <http://protege.stanford.edu/>
- Qin, L. and Atluri, V. (2009) 'Evaluating the validity of data instances against ontology evolution over the Semantic Web', *Information and Software Technology*, 51(1), pp. 83–97. doi: <https://doi.org/10.1016/j.infsof.2008.01.004>.
- QuetteDiaf, S.; Diaf, D.; Belhamel, M.; Haddadi, M.; Louche, A.A. Methodology for Optimal Sizing of Autonomous Hybrid PV/wind System. *Energy Policy* 2007, 35, 5708–5718.
- Raad, J. and Cruz, C. (2015) *A Survey on Ontology Evaluation Methods*. doi: 10.5220/0005591001790186.
- Răboacă, M., Badea, G., Enache, A., Filote, C., Rasol, G., Rata, M., Lavric, A. and Felseghi, R. A. (2019) 'Concentrating Solar Power Technologies', *Energies*, 12, p. 1048. doi: 10.3390/en12061048.
- Rahman, M.M. A., Al Awami, A. T., Rahim, AHMA (2014), 'Hydro-PV-Wind-Battery-Diesel Based Stand-Alone Hybrid Power System', 'International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT)'
- Rashid, Y., Rashid, A., Warraich, M. A., Sabir, S. S., & Waseem, A. (2019, July 24). Contents Full Article Content List Abstract Introduction Background Section I Section II Discussion and Conclusion References Accessing resources off campus can be a challenge. Lean library can solve it Lean Library Figures & Tables Article Metrics Rela. *International Journal of Qualitative Methods*, 18.
- Rasmussen, M. H., Lefrancois, M., Schneider, G. F., and Pauwels, P. (2020) 'BOT: the Building Topology Ontology of the W3C Linked Building Data Group', *Semantic Web*. E. Janowicz, K. doi: 10.3233/SW-200385.
- Rector, A. L., Horridge, M., Brandt, S., Parsia, B. (2014), 'A Domain-Specific Ontology Authoring Environment for a Clinical Documentation System', 'IEEE 27th International Symposium on Computer-Based Medical Systems'.
- Rehau (2014), 'Sustainable Building Technology, Product Catalog', Unlimited Polymer Solutions.
- Rethinking The Future (FTF) (2021). '5 Most Important Green Rating systems in Architecture 2022-RT', Retrieved 04 January 2022, from <https://www.re-thinkingthefuture.com/architects-lounge/a1514-5-most-i...>
- Revizto (2022), 'Top BIM Construction Software Tools in 2022', <https://revizto.com/en/best-bim-software-tools-2021/>
- Rhodes, J. (2014), 'The Chronicle of Evidence-Based Mentoring; On Methods: What's the difference between qualitative and quantitative approaches?', <http://chronicle.umbmentoring.org/on-methods-whats-the-difference-between-qualitative-and-quantitative-approaches/>
- Rhodes, J.D., Upshaw, C.R., Cole, W.J., Holcomb, C.L., Webber, M.E., (2014), 'A Multi-objective Assessment of the Affect of Solar PV Array Orientation and Tilt on Energy Production and System Economics, *Solar Energy*, 108, pp.28-40.

- Rigo, P. D., Siluk, J. C. M., Lacerda, D. P. Rediske, G., and Rosa, C. B.(2022) 'The competitiveness factors of photovoltaic installers companies and a BSC model', *Solar Energy*, 235, pp. 36–49. doi: <https://doi.org/10.1016/j.solener.2022.02.017>.
- Roaf, S. Fuentes, M. and Thomas-Rees, S. (2012) *Ecohouse*. Routledge, UK.
- Saeedi, K., & Visvizi, A. (2021, February 13). Software Development Methodologies, HEIs, and the Digital Economy. *Education Sciences*, 11(73), <https://doi.org/10.3390/educsci11020073>.
- Salim, A. M. and Abu Dabous, S. (2023) 'A review of critical success factors for solar home system implementation in public housing', *International Journal of Energy Sector Management*, 17(2), pp. 352–370. doi: 10.1108/IJESM-11-2021-0004.
- Salah MS, Ismail O, Kamel A, Hassan H. (2018), From CommonKADS to SOA environment: an adaptation model. *Arab J Sci Eng*. 2018;43(12):7605-7619.
- Sambo, A.S.; Zarma, I.H.; Ugwouke, P.E.; Dioha, I.J.; Ganda, Y.M. (2014), Implementation of Standard Solar PV Projects in Nigeria. *J. Energy Technol. Policy*, 4.
- Sameen, F. and Quratulain, R. (2012), 'A Comparison of Algorithms to Construct Ontologies from Relational Databases', CCGI 2012: The Seventh International Multi-Conference on Computing in the Global Information Technology.
- Samuels, P. (2019). *Qualitative Research Methods*. <https://doi.org/10.13140/RG.2.2.27589.60649>
- Sánchez de Castro, A. (2022) *Semantic Web and Knowledge graphs: History and Evolution*. doi: 10.13140/RG.2.2.13299.66082.
- Sanchez, V., Ramirez, J.M., and Arriaga, G. (2010), 'Optimal sizing of a hybrid renewable system," In: IEEE international conference on industrial technology (ICIT), pp. 949-54.
- Sandberg, M. (2015). *Towards a Knowledge-Based Engineering Methodology for Construction*. <https://doi.org/10.1061/9780784479377.001>
- Sangmesh, B., et al. (2023) 'Development of sustainable alternative materials for the construction of green buildings using agricultural residues: A review', *Construction and Building Materials*, 368, p. 130457. doi: <https://doi.org/10.1016/j.conbuildmat.2023.130457>.
- Sangwongwanich, A.; Yang, Y.; Sera, D.; Zhou, D.; (2018) Blaabjerg, F. On the Impacts of PV Array Sizing on Inverter Reliability and Lifetime. *Trans. Ind. Appl.* 2018, 54, 3656–3667.
- Santirojanakul S. (2018), The development of sports science knowledge management systems through Com- monKADS and digital Kanban board. In: 2018 IEEE Symposium on Computer Applications and Industrial Electronics (ISCAIE 2018); 2018:119-124.
- Santos, G. et al. (2016) *Electricity Markets Ontology to Support MASCEM's Simulations*. doi: 10.1007/978-3-319-39387-2_33.
- SAREF (2020), 'SAREF: the Smart Applications REference ontology', <https://saref.etsi.org/core/v3.1.1/>
- Sarkar, G. (2014) Analysis of shading patterns of solar panels. *Int. J. Res. Eng. Technol.* 2014, 3, 594–599.
- Schachinger, M.; AG, S. Module Size Index. *PV Magazine*. Available online: <https://www.pv-magazine.com/features/investors/module-price-index/> (accessed on 16 November 2019).
- Schevers, H., Mitchell, J., Akhurst, P., Marchant, D., Bull, S., MacDonald, K., Drogemuller, R., and Lining, C. (2015) Towards digital facility for modelling for Sydney Opera house using IFC and semantic web technology, *ITcon 12 (2007)* 347–362. Retrieved April 2015.
- See, R., Karlshoej, J. & Davis, D., 2011. *An Integrated Process for Delivering IFC Based Data Exchange*, s.l.: s.n.
- Schreiber, G., Akkerman, H., Anjewierden, A., de Hoog, R., Shadbolt, N., Van de Velde, W., & Wielinga, B. (2000). *Knowledge Engineering and Management: The CommonKADS Methodology*. MIT Press.
- Schreiber, G., Wielinga, B., Hoog, R., Akkermans, H. (2002), 'CommonKADS: a comprehensive methodology for KBS development,' *IEEE Expert (Volume: 9, Issue: 6)*, Pages 28 – 37, 0885-9000
- Schweighofer, B. et al. (2023) 'Optimised selection of component models for photovoltaic and energy storage system simulations', *Solar Energy*, 249, pp. 559–568. doi: <https://doi.org/10.1016/j.solener.2022.12.007>.
- SEC, (2015), 'Solar Electricity Costs', http://solarcellcentral.com/cost_page.html#solar_cost
- SEIA. (2023, February 3). *Solutions for Today's Energy Needs*. Retrieved from Solar Energy Technologies: <https://www.seia.org/sites/default/files/inline-files/SEIA-Solar-Energy-Technologies-Factsheet-2018-April.pdf>
- Selin, N. E. (2023, February 27). *Wind Power Energy*. Retrieved from Britannica: <https://www.britannica.com/science/wind-power>
- Seme, S.; Sredensek, K.; Stumberger, B.; Hadziselimovic, M. Analysis of the Performance of Photovoltaic Systems in Slovenia. *Solar Energy* 2019, 180, 550–558.
- Sezen, K. and Gungor, A. (2023) 'Comparison of solar assisted heat pump systems for heating residences: A review', *Solar Energy*, 249, pp. 424–445. doi: <https://doi.org/10.1016/j.solener.2022.11.051>.

- Shadbolt, N. R. (2005), 'Eliciting Expertise', In J. R. Wilson & N. Corlett (Eds.), *Evaluation of Human Work* (3rd ed.). CRC Press, Boca Raton, Florida, USA.
- Shadbolt, N. R., & Burton, M. (1995), 'Knowledge Elicitation', In J. R. Wilson & N. Corlett (Eds.), *Evaluation of Human Work: A Practical Ergonomics Methodology*. Taylor & Francis, London, England.
- Shadbolt, N. R., & Smart, P. R. (2015), 'Knowledge Elicitation', In J. R. Wilson & S. Sharples (Eds.), *Evaluation of Human Work* (4th ed.). CRC Press, Boca Raton, Florida, USA.
- Shadbolt, N., Tope, O., John, D., Alistair, D., and Hugh, G. (2014), 'Linking social, open, and enterprise data', In *WIMS14: 4th International Conference on Web Intelligence, Mining and Semantics*, Thessaloniki, GR, 02 - 04 Jun 2014. ACM8pp.
- Shadbolt, Nigel and Smart, Paul R (2015) *Knowledge Elicitation: Methods, Tools and Techniques*. In, Wilson, John R and Sharples, Sarah (eds.) *Evaluation of Human Work*. Boca Raton, Florida, USA. CRC Press, pp. 163-200.
- Shafique, M., Luo, X. and Zuo, J. (2020) 'Photovoltaic-green roofs: A review of benefits, limitations, and trends', *Solar Energy*, 202, pp. 485–497. doi: <https://doi.org/10.1016/j.solener.2020.02.101>.
- Sharma, S. and Sengar, N. (2022) 'Review of solar PV training manuals and development of survey based solar PV system training formats for beginners', *Solar Energy*, 241, pp. 72–84. doi: <https://doi.org/10.1016/j.solener.2022.05.055>.
- Siddiqui, A. A. *et al.* (2021) 'Building Information Modelling for performance-based Fire Safety Engineering analysis – A strategy for data sharing', *Journal of Building Engineering*, 42, p. 102794. doi: <https://doi.org/10.1016/j.jobe.2021.102794>.
- Da-Silva, M.K. (2019), Comparative Study of Sky Diffuse Irradiance Models Applied to Photovoltaic Systems. In *Proceedings of the International Conference on Smart Energy Systems and Technologies (SEST)*, Porto, Portugal, 9–11 September 2019; pp. 1–5.
- Silva, J.L.; de Mesquita, B.D.; Moreira, H.S.; Kitayama, M.; Villalva, M.G. (2019) A review and analysis of technologies applied in PV modules. In *Proceedings of the IEEE PES Innovative Smart Grid Technologies Conference–Latin America (ISGT Latin America)*, Campinas, Brazil, 11 November 2019.
- Sinitskaya, E. *et al.* (2019) 'Examining the Influence of Solar Panel Installers on Design Innovation and Market Penetration', *Journal of Mechanical Design*, 141(4). doi: 10.1115/1.4042343.
- Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A. & Katz, Y. (2007), 'Pellet: A practical OWL-DL reasoner', *Journal of Web Semantics* 5(2).
- Sivestre, S., Kichou, S., Chouder, A., Nofuentes, G., Karatepe, E., (2015), 'Analysis of current and voltage indicators in grid-connected PV systems working in faulty and partial shading conditions', *Energy*, pp.1-9.
- Smith B., (2003), 'Ontology', *Blackwell guide to the philosophy of computing and information* (2003), pp. 155-166
- SoftwareAdvice (2022). 'BIM Software', <https://www.softwareadvice.co.uk/directory/3417/bim/software>
- Solar-Energy. (2023, February). *Learn Online from the Experts*. Retrieved from Solar Energy International: <https://www.solarenergy.org/>
- Solar-Feeds, (2021). *Pros and Cons of Green Buildings: Do the Advantages Outweigh the Disadvantages*. [Online] Available at [https://www.solarfeeds.com/mag/pros-and-cons-of-green-buildings/#Are there any downsides to green building](https://www.solarfeeds.com/mag/pros-and-cons-of-green-buildings/#Are%20there%20any%20downsides%20to%20green%20building) [Accessed October 2021].
- Song, X., Liu, J., Wang, X. (2014), 'A Novel Automatic Ontology Construction Method Based on Web Data', 2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IEEE Computer Science.
- Sorce. (2019), Sorce: Solar Radiation and Climate Experiment. 2019. Available online: <http://lasp.colorado.edu/home/sorce/> (accessed on 25 December 2019).
- Spataru, S.S.; Kerekes, T.; Teodorescu, R. (2015), Diagnostic method for photovoltaic systems based on light I-Vmeasurements. *Solar Energy* 2015, 119, 29–44.
- Spoladore, D. and Pessot, E. (2022) 'An evaluation of agile Ontology Engineering Methodologies for the digital transformation of companies', *Computers in Industry*, 140, p. 103690. doi: <https://doi.org/10.1016/j.compind.2022.103690>.
- SRH. (2023, February). *Solar Winter School* . Retrieved from Berlin University of Applied Sciences: <https://www.srh-berlin.de/en/short-courses/training-solar-winter-school/>

- Staab, S. and Struder, R. (2010), 'Handbook on Ontology', 'International handbooks on Information Systems'.
- Staab, S., Studer, R. and Sure, Y. (2003), 'On-To- Knowledge Methodology", S. Staab and R. Studer, (Eds.): Handbook on Ontologies, Springer, pp. 117-132.
- Stand-Alone Photovoltaic Systems; (2015), 'A Handbook of Recommended Design Practices', Sandia National, Photovoltaic Design Assistance Center, Sandia National Laboratories, Albuquerque, New Mexico. Retrieved August 2015.
- StanfordUniversity. (2020). *A free, open-source ontology editor and framework for building intelligent systems*. Retrieved December 2020, from Protege: <https://protege.stanford.edu>
- Stang, S. *et al.* (2021) 'Spatially optimised distribution of household rainwater harvesting and greywater recycling systems', *Journal of Cleaner Production*, 312, p. 127736. doi: <https://doi.org/10.1016/j.jclepro.2021.127736>.
- Stephan, G.s. Pascal, H.s. Andreas A.s. (2007), 'Knowledge representation and ontologies Semantic web services: Concepts, technologies, and applications (2007), pp. 51-105, [10.1007/3-540-70894-4_3](https://doi.org/10.1007/3-540-70894-4_3)
- Stevens, R., Vigo, M., Bail, S., Jay, C. (2014), 'Overcoming the Pitfalls of Ontology Authoring: Strategies and Implications for Tool Design', *International Journal of Human-Computer Studies*, pp. 835 – 845.
- Stolk, S. & McGlinn, K., (2020). *Validation of IfcOWL datasets using SHACL*. Dublin, s.n., pp. 91-104.
- Studer, R., Benjamins, V. R., & Fensel, D. (1998). Knowledge engineering: principles and methods. *Data Knowl Eng* 25(1-2):161-197. *Data & Knowledge Engineering*, 25, 161–197. [https://doi.org/10.1016/S0169-023X\(97\)00056-6](https://doi.org/10.1016/S0169-023X(97)00056-6)
- Suárez-Figueroa, M.C. (2012), 'NeOn Methodology for Building Ontology Networks: Specification, Scheduling and Reuse', *Dissertations in Artificial Intelligence*, vol. 338, ISBN: 978-3- 89838-338-7. IOS Press, 2012.
- Suetsugu, M.; Takahashi, Y.; Fujiwara, K. (2018) Power Generation and Safety Performance of Photovoltaic System when Mixing Photovoltaic Modules with Different Installation Angles. In Proceedings of the IEEE 7th World Conference on Photovoltaic Energy Conversion (WCPEC) (A Joint Conference of 45th IEEE PVSC, 28th PVSEC & 34th EU PVSEC), Waikoloa Village, HI, USA, 10–15 June 2018; pp. 2351–2356.
- Sun, Z. *et al.* (2020) 'Domain Ontology Construction and Evaluation for the Entire Process of Software Testing', *IEEE Access*, 8, pp. 205374–205385. doi: 10.1109/ACCESS.2020.3037188.
- SunPower, 2021. *Record Breaking Solar Panels for your home or buisness*. [Online] Available at: <https://sunpower.maxeon.com/uk/> [Accessed October 2021].
- Sustainable Sources, (2015) Photovoltaic Systems. <http://photovoltaics.sustainablesources.com/> Date retrieved April 2015.
- Ta, D. C. and Thi, T. P. (2015), 'Identifying Semantic and Syntactic Relations from Text Documents', The 2015 IEEE RIVF International Conference on Computing & Communication Technologies Research, Innovation, and Vision for Future (RIVF)
- Tao, J., Ding, L. and McGuinness, D. L. (2009) 'Instance Data Evaluation for Semantic Web-Based Knowledge Management Systems', in *2009 42nd Hawaii International Conference on System Sciences*, pp. 1–10. doi: 10.1109/HICSS.2009.263.
- Tan, J., Cai, J. and Li, J. (2022) 'Recycling of unseparated construction and demolition waste (UCDW) through geopolymer technology', *Construction and Building Materials*, 341, p. 127771. doi: <https://doi.org/10.1016/j.conbuildmat.2022.127771>.
- Tanesab, J.; Parlevliet, D.; Whale, J.; Urmee, T. (2019), The Effect of Dust with Different Morphologies on the Performance Degradation of Photovoltaic Modules. *Sustain. Energy Technol. Assess.* 2019, 31, 347–354.
- Tang, D. and Liu, K. (2022) 'Exploring the Application of BIM Technology in the Whole Process of Construction Cost Management with Computational Intelligence.', *Computational Intelligence & Neuroscience*.
- Tang, K. H. D., Foo, C. Y. H. and Tan, I. S. (2020) 'A review of the green building rating systems', *IOP Conference Series: Materials Science and Engineering*, 943(1), p. 12060. doi: 10.1088/1757-899X/943/1/012060.
- Tchouanguem Djuedja, J. F. *et al.* (2019) 'Integration of environmental data in BIM tool & linked building data', *CEUR Workshop Proceedings*, 2389, pp. 78–91.
- Terkaj, W. & Pauwels, P., (2015a). *EXPRESS ontology*. [Online] Available at: <https://w3id.org/express> [Accessed May 2020].

- Terkaj, W. & Pauwels, P. (2015b). *LIST Ontology*. [Online] Available at: <https://w3id.org/list> [Accessed May 2020].
- Terkaj, W. and Pauwels, P. (2017) 'A method to generate a modular ifcOWL ontology,' *CEUR Workshop Proceedings*, 2050(September 2017).
- The National Renewable Energy Laboratory, DOE National Laboratory. December 2003 DOE/GO-102003-1844, Accessed August 2015. www.eere.energy.gov
- The WorldWatch Institute (2015), Global Energy and Resources, <http://vitalsigns.worldwatch.org/trends/global-economy-resources>. Date Accessed: June 15th, 2015.
- Thomas, R. V, Nair, D. G. and Enserink, B. (2023) 'Conceptual framework for sustainable construction,' *Architecture, Structures and Construction*. doi: 10.1007/s44150-023-00087-8.
- Trimble, (2020). *Trimble inc*. [Online] Available at: <https://www.sketchup.com>[Accessed 9 March 2021].
- Tsarkov, D. & Horrocks, I. (2006), 'Fact++ description logic reasoner: System description', in U. Furbach & N. Shankar, eds, 'Automated Reasoning, Third International Joint Conference, IJCAR 2006, Seattle, WA, US App. 292 - 297.
- Tudorache, T., Nyulas, C., Noy, N. F., & Musen, M. A. (2013), 'WebProtégé: A collaborative ontology editor and knowledge acquisition tool for the Web, *Semantic Web*, 4(1), 89-99.
- Udemy, (2022), 'A Broad Selection of Courses', <https://www.udemy.com/courses/search/?q=how+to+design+photovoltaic+systems+PV&sort=relevance&src=ukw>
- Ugwu, C. and Eze, V. (2023) 'Qualitative Research', 8, pp. 20–35.
- UKPower (2020). Compare energy prices per kWh. UKPower. 2020. Available online: https://www.ukpower.co.uk/home_energy/tariffs-per-unit-kwh (accessed on 9 January 2020).
- UL Standards. Available online: https://standardscatalog.ul.com/standards/en/standard_4703_1 (accessed on 26 November 2019).
- USC. (2020, November 1). *Research Guides*. Retrieved January 2021, from USCLibraries: <https://libguides.usc.edu/writingguide/quantitative>
- Uschold, M. and Gruninger M. (1996) "Ontologies: Principles, Methods and Applications," *Knowledge Engineering Review*, vol. 11, pp. 93-136, 1996.
- Uschold, M. and King, M. (1995) 'Towards a Methodology for Building Ontologies', in.
- Usman, Z. *et al.* (2020b) 'A Critical Appraisal of PV-Systems' Performance', *Buildings*, 10(11). doi: 10.3390/buildings10110192.
- Usman, Z. S. *et al.* (2018) 'Integration of an ontology with ifc for efficient knowledge discovery in the construction domain', *eWork and eBusiness in Architecture, Engineering and Construction - Proceedings of the 12th European Conference on Product and Process Modelling, ECPPM 2018*, pp. 429–437. doi: 10.1201/9780429506215-54.
- Usman, Z. S. *et al.* (2020a) 'Developing a Process Map for the Design of a PV System', *Lecture Notes in Networks and Systems*, 72, pp. 285–301. doi: 10.1007/978-3-030-13697-0_22.
- VanBerlo, L., (2019). *Future technology Presentation buildingSMART Summit Beijing*. [Online] Available at: <https://www.linkedin.com/pulse/future-technology-presentation-buildingsmart-summit-léon-van-berlo>[Accessed March 2021].
- Venkateswari, R.; Sreejith, S. Factors Influencing the Efficiency of Photovoltaic Systems. *Renew. Sustain. Energy Rev.* 2019, 101, 376–394.
- Vieira, O.L.; Silva, F.J.; Machano, L.A.; Correia, A.N.; Vitoriano, C.T.; Oliveira, A.H.; Franklin, R.P.D.; Menezes, J.C.F. Comparison of Photovoltaic Plant Performance Estimation Models: A Case Study for Fortaleza, Brazil. *IEEE Lat. Am. Trans.* 2018, 16, 1898–1906.
- Villazón-Terrazas, B., Vilches-Blázquez, L. M., Corcho, O. and Gómez-Pérez, A. (2011), 'Methodological Guidelines for Publishing Government Linked Data", D. Wood, (Ed.): *Linking Government Data*, New York: Springer, pp. 27-49.
- Vitro (2011), 'Vitro - Integrated Ontology Editor and Semantic Web Application', <http://vitro.mannlib.cornell.edu/> Date Accessed: 14th March 2015.
- Vo, M. H. L. and Hoang, Q. (2020) 'Transformation of UML class diagram into OWL Ontology', *Journal of Information and Telecommunication*, 4(1), pp. 1–16. doi: 10.1080/24751839.2019.1686681.
- W3C, (2015), 'OWL Web Ontology Language Current Status', http://www.w3.org/standards/techs/owl#w3c_all
- W3C. (2012a). *OWL 2 Web Ontology Language Primer (Second Edition) W3C Recommendation*. Retrieved January 2023, from <https://www.w3.org/TR/owl2-primer/> (Accessed on December 11, 2017)

- W3C. (2012b). *Semantic Web Standards OWL*. Retrieved January 2023, from W3C: <https://www.w3.org/OWL/> (Accessed on December 11, 2017)
- W3C. (2020a). *Linking Data to Knowledge*. Retrieved from Semantic Web Company: <https://semantic-web.com>
- W3C. (2020b). *Semantic Web*. Retrieved December 2020, from W3C: <https://www.w3.org/standards/semanticweb/>
- Wakelin, J. (2015), 'Product and technology. Bladeless Wind Turbines', Energy Saving Trust', www.energysavingtrust.org.uk/bladeless-wind-turbines
- Wang, J. (2011), 'Study on Text Retrieval Based on Domain Ontology Extensions Query'. 2011.
- Wang, Y. *et al.* (2023) 'A review on new methods of recycling waste carbon fiber and its application in construction and industry', *Construction and Building Materials*, 367, p. 130301. doi: <https://doi.org/10.1016/j.conbuildmat.2023.130301>.
- Wagner, A. *et al.* (2022) 'Building product ontology: Core ontology for Linked Building Product Data', *Automation in Construction*, 133, p. 103927. doi: <https://doi.org/10.1016/j.autcon.2021.103927>.
- Wagner, A. and Ruppel, U. (2019) *BPO: The Building Product Ontology for Assembled Products*.
- Warrender, J. & Lord, P., 2015. *How, What and Why to test an ontology*. Dublin, Ireland, s.n.
- WCED. (1987). *Report of the World Commission on Environment and Development: Our Common Future*.
- Werbrouck, J., Senthilvel, M., Beetz, J., and Pauwels, P. (2019), Querying heterogeneous linked building datasets with context-expanded GraphQL queries. 2019 Proceedings of the 7th Linked Data in Architecture and Construction Workshop, LDAC 2019. In CEUR workshop proceedings 2389. p21-34.
- Whitlock, R., (2021). *UK solar capacity grows by 545 MW in 2020*. [Online] [Accessed November 2021].
- WIPO. (2022). *Green Technology Book 2022 Solutions for climate change adaptation*. Retrieved from World Intellectual Property Organization: <https://www.wipo.int/edocs/pubdocs/en/wipo-pub-1080-en-green-technology-book.pdf>
- WorldGBC, (2019). *New report: the building and construction sector can reach net zero carbon emissions by 2050*. [Online] Available at: <https://www.worldgbc.org/news-media/WorldGBC-embodied-carbon-report-published> [Accessed October 2021].
- Xia, H. *et al.* (2022) 'Study on city digital twin technologies for sustainable smart city design: A review and bibliometric analysis of geographic information system and building information modelling integration', *Sustainable Cities and Society*, 84, p. 104009. doi: <https://doi.org/10.1016/j.scs.2022.104009>.
- Xiao, Y.-Q., Li, S.-W. and Hu, Z.-Z. (2019) 'Automatically Generating a MEP Logic Chain from Building Information Models with Identification Rules', *Applied Sciences*, 9(11). doi: 10.3390/app9112204.
- Xu, G. (2010) Research on Ontology Construction Method based on Relational Database [D]. 2010.
- Yang TH, Ku CY, Yen DC, Hsieh WH. (2016), An improved strategic information management plan for medical institutes. *Comput Stand Interfaces*. 2016;45:26-36.
- Yolcan, O. O. and Kose, R. (2023) 'Photovoltaic module cell temperature estimation: Developing a novel expression', *Solar Energy*, 249, pp. 1–11. doi: <https://doi.org/10.1016/j.solener.2022.11.020>.
- Younis, A., Abdeljalil, A. and Omer, A. (2023) 'Determination of panel generation factor using peaks over threshold method and short-term data for an off-grid photovoltaic system in Sudan: A case of Khartoum city', *Solar Energy*, 249, pp. 242–249. doi: <https://doi.org/10.1016/j.solener.2022.11.039>.
- Younis, A. and Alhorr, Y. (2021) 'Modeling of dust soiling effects on solar photovoltaic performance: A review', *Solar Energy*, 220, pp. 1074–1088. doi: <https://doi.org/10.1016/j.solener.2021.04.011>.
- Yu, L. (2011), 'A Developer's Guide to the Semantic Web', Springer Science & Business Media, 3642159702, 9783642159701, <https://books.google.com.ng/books?id=G0toUPABjEC>
- Yun, W., Zhang, X., Li, Z., Liu, H., & Han, M. (2021, January 4). Knowledge Modelling: A survey of processes and techniques. *International Journal of Intelligent Systems*(36), 1686-1720.
- Zaihideo, F.M.; Mekhilef, S.; Seyedmahmoudian, M.; Horan, B. Dust as an unalterable deteriorative factor affecting PV panel's efficiency: Why and how. *Renew. Sustain. Energy Rev*. 2016, 65, 1267–1278.
- Zeginis, D., Hasnain, A., Loutas, N., Deus, H. F., Fox, R., Tarabanis, K. (2012), 'A collaborative methodology for developing a semantic model for interlinking Cancer Chemoprevention linked-data sources.
- Zeman, M. (2015), 'Photovoltaic Systems', Chapter 9 in *Solar Cells*, Delft University of Technology. Date retrieved May 2015.
- Zhang, C., Beetz, J., & De Vries, B. (2018). BimSPARQL: Domain-Specific Functional SPARQL Extensions for Querying RDF Building Data. *Semantic Web*, 9 (6), 829-855.

- Zhang, S., Boukamp, F., & Teizer, J. (2015). Ontology-Based Semantic Modeling of Construction Safety Knowledge: Towards Automated Safety Planning for Job Hazard Analysis (JHA). *Automation in Construction*, 52, 29–41.
- Zhao, N. *et al.* (2023) 'Emerging information and communication technologies for smart energy systems and renewable transition', *Advances in Applied Energy*, 9, p. 100125. doi: <https://doi.org/10.1016/j.adapen.2023.100125>.
- Zheng, H., Li, S., Chaloo, R., and Proano, J., (2014), 'Shading and bypass diode impacts to energy extraction of PV arrays under different converter configurations, *Renewable Energy*', 68, pp. 28-66.
- Zheng, Z.; Zhuang, Z.; Lian, Z.; Yu, Y. (2017) Study on Building Energy Load Prediction Based on Monitoring Data. *Proc. Eng.* 2017, 205, 716–723.
- Zhou, J. *et al.* (2020) 'Operational performance of a novel heat pump coupled with mini-channel PV/T and thermal panel in low solar radiation', *Energy and Built Environment*, 1(1), pp. 50–59. doi: <https://doi.org/10.1016/j.enbenv.2019.08.001>.
- Zima, D. (2015). MODERN METHODS OF SOFTWARE DEVELOPMENT. *Task Quarterly*, 19(4), 481-493. October 1
- Zulkifli, NA, (2014), 'Probabilistic Analysis of Solar Photovoltaic Output Based on Historical Data, IEEE 8th International Power Engineering

APPENDIX

Appendix 1: Energy load Guide

- Make a list of all appliances used in the building
- Find the wattage of each appliance. This is often stamped on the appliance; if it is not, multiply the appliance's amperage vs voltage
- Track the number of hours each appliance runs for per day. Keep a log to calculate an average
- The spreadsheet calculates the daily and annual energy consumption
- This can be done for every room in the building and a summary can be obtained as seen below.

Note: 1. You could install electricity usage monitors between the appliances and the outlet. Read figures for a week and input into the table.

Note: 2. For more detailed data, you may consider installing a WHOLE-building energy monitoring system. The monitors are installed directly to the building's main circuit breaker. Some monitors may be connected to the building's wireless network and energy data is viewed on a smart device (computer or mobile phone).

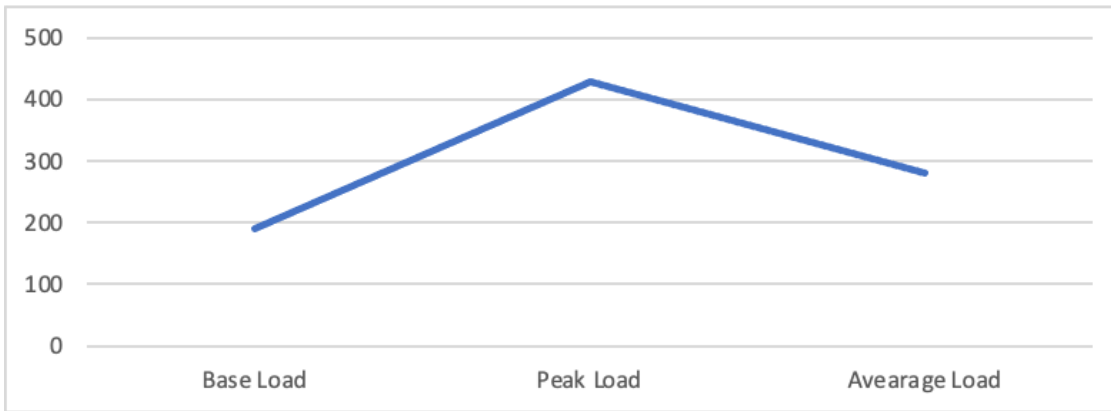
Energy load Guide

Bedroom			Hours Used								Energy Load		
Appliance	Quantity	Wattage	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Average Hours/Day	Daily Energy Consumption (kWh/Day)	Annual Energy Consumption (kWh/Day)	
1										=AVERAGE(F13:L13)	=(E13*M13*D13)/1000	=N13*365	
2										=AVERAGE(F14:L14)	=(E14*M14*D14)/1000	=N14*365	
3										=AVERAGE(F15:L15)	=(E15*M15*D15)/1000	=N15*365	
4										=AVERAGE(F16:L16)	=(E16*M16*D16)/1000	=N16*365	
5										=AVERAGE(F17:L17)	=(E17*M17*D17)/1000	=N17*365	
Total		=SUM(E13:E17)	=SUM(F13:F17)	=SUM(G13:G17)	=SUM(H13:H17)	=SUM(I13:I17)	=SUM(J13:J17)	=SUM(K13:K17)	=SUM(L13:L17)	=SUM(M13:M17)	=SUM(N13:N17)	=SUM(O13:O17)	

Living Room			Hours Used								Energy Load		
Appliance	Quantity	Wattage	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Average Hours/Day	Daily Energy Consumption (kWh/Day)	Annual Energy Consumption (kWh/Day)	
1										=AVERAGE(F22:L22)	=(E22*M22*D22)/1000	=N22*365	
2										=AVERAGE(F23:L23)	=(E23*M23*D23)/1000	=N23*365	
3										=AVERAGE(F24:L24)	=(E24*M24*D24)/1000	=N24*365	
4										=AVERAGE(F25:L25)	=(E25*M25*D25)/1000	=N25*365	
5										=AVERAGE(F26:L26)	=(E26*M26*D26)/1000	=N26*365	
Total		=SUM(E22:E26)	=SUM(F22:F26)	=SUM(G22:G26)	=SUM(H22:H26)	=SUM(I22:I26)	=SUM(J22:J26)	=SUM(K22:K26)	=SUM(L22:L26)	=SUM(M22:M26)	=SUM(N22:N26)	=SUM(O22:O26)	

Kitchen			Hours Used								Energy Load		
Appliance	Quantity	Wattage	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Average Hours/Day	Daily Energy Consumption (kWh/Day)	Annual Energy Consumption (kWh/Day)	
1										=AVERAGE(F31:L31)	=(E31*M31*D31)/1000	=N31*365	
2										=AVERAGE(F32:L32)	=(E32*M32*D32)/1000	=N32*365	
3										=AVERAGE(F33:L33)	=(E33*M33*D33)/1000	=N33*365	
4										=AVERAGE(F34:L34)	=(E34*M34*D34)/1000	=N34*365	
5										=AVERAGE(F35:L35)	=(E35*M35*D35)/1000	=N35*365	
Total		=SUM(E31:E35)	=SUM(F31:F35)	=SUM(G31:G35)	=SUM(H31:H35)	=SUM(I31:I35)	=SUM(J31:J35)	=SUM(K31:K35)	=SUM(L31:L35)	=SUM(M31:M35)	=SUM(N31:N35)	=SUM(O31:O35)	

Summary				
Base	Peak	Average	Total Average Daily Load (kWh/Day)	Total Average Annual Load (kWh/Day)
=MIN(N13:N17,N22:N26,N31:N35)	=MAX(N13:N17,N22:N26,N31:N35)	=AVERAGE(N13:N17,N22:N26,N31:N35)	=SUM(N18,N27,N36)	=SUM(O18,O27,O36)



A graph can be obtained to look like the figure above, to clearly show the base, peak and average load of the building.