A MULTI AGENT BASED ELECTRICITY METER MANAGEMENT SYSTEM

Thushan Sanjeewa Rathnayake

208546U

Master of Science in Artificial Intelligence

Department of Computational Mathematics

University of Moratuwa

Sri Lanka

July 2024

A MULTI AGENT BASED ELECTRICITY METER MANAGEMENT SYSTEM

Thushan Sanjeewa Rathnayake

208546U

Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Artificial Intelligence

Department of Computational Mathematics

University of Moratuwa
Sri Lanka

July 2024

DECLARATION

I declare that this is my own work and this thesis does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other University or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: Date: 21st July 2024

The above candidate has carried out research for the Master's thesis under my supervision. I confirm that the declaration made above by the student is true and correct.

Name of Supervisor:

Dr. A T P Silva

Signature of the Supervisor: Date: 22/7/2024

ABSTRACT

Electricity grid is evolving from a one-way power flowing traditional grid to a two-way power flowing Smart Grid with embedded generators, battery storages and electrical vehicles driven by the need for addressing climate change challenges. Following these changes of the electricity grid, electricity meters also have evolved from mechanical meters to Smart Meters that are capable of measuring two-way power flow, communicating in both ways and controlling power flow. Advanced Metering Infrastructure is becoming the key system in Smart Grid, making Smart Meter the key component in Smart Grid. Moving from traditional grid to Smart Grid has to face the challenge of managing millions of different types of meters in the same system. This research work using Multi Agent Technology will address this critical challenge.

For the purpose of providing a solution for the challenge of managing electricity metering system, planning and policy evaluation part in management were considered. The planning activities were divided into several planning subtasks. The considered subtasks of management in this study consist of preparation of annual maintenance plan, preparation of material plans and evaluation of metering policies. By its nature the considered problem is currently tackled by the power system professionals, it was identified their approach involve dividing planning tasks among different individuals with different authorities and responsibilities. The inspiration for using Multi Agent Planning technology derived from that. Further, it was seen that the domain knowledge and the problem solving knowledge can also be separated. This led to modeling the domain knowledge as an OWL ontology and the problem solving knowledge into the skills of the agents in the designed Multi Agent System. The system takes metering ontology, meter database, meter issues, working days of employees and stock levels as inputs to the system for providing outputs of maintenance plan, material plan and policy evaluation report. Designed Multi Agent System consists of four types of agents namely; MA agent, EA agent, SK agent and ES agent. MA agent acts as the interface agent and takes inputs from the user. Based on the user inputs, MA agent assigns EA agent planning or policy evaluation tasks. Upon receiving instructions from MA agent, EA agent prepares Activity Plans comparing meter data base and meter issues with the SPARQL queries on OWL ontology. Subsequently, depending on MA agent's instructions, EA agent allocates tasks of material planning and preparation of Activity Schedule to SK agent and ES agents respectively. Once the ES agents prepared their Activity Schedules, EA agent merges those into one schedule. OWL ontology was created using Protégé ontology editing tool and the Multi Agent System was developed in Microsoft VScode. The developed system was evaluated using a set of data relevant to actual smart meters. Evaluation was done in three steps; manually by human employees, using a spreadsheet and using the developed system. Results revealed that the developed system can produce required plans more quickly with much more details.

Key Words: MAS, MAP, Ontology, AMI, Electricity Metering

DEDICATION

It is with great pleasure that I am dedicating this thesis to all who have helped me in the journey of education including my supervisor Dr. A T P Silva, all of my teachers, my wife, parents, brothers and my good friends.

ACKNOWLEDGEMENT

My sincere gratitude to Dr. A T P Silva for supervising this project and giving invaluable guidance and timely feedback along the way.

Further, I would like to appreciate all the lecturers of Department of Computational Mathematics for the guidance given to me throughout this program.

Ceylon Electricity Board, my place of work, is acknowledged for inspiring me to carry out this project in a practical way and the extended support.

Finally, heartfelt gratitude to my family for standing by me and for believing in me and also my colleagues and friends for the support and companionship throughout.

.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Description
CEB	Ceylon Electricity Board
LECO	Lanka Electricity Company Private Limited
AI	Artificial Intelligence
PUCSL	Public Utilities Commission of Sri Lanka
MAS	Multi Agent Systems
MAP	Multi Agent Planning

CHAPTER 1

INTRODUCTION

1.1 Prolegomena

Electricity has been playing a key role in everyone's life being the form of energy that is used for lighting, cooking, cooling and heating, communication, entertainment, driving industrial machines and more trending applications of mobility. In recent years, Electricity consumption is around 20% of the world's final energy consumption [1]. Global electricity production in 2021 had been around 28,520 TWh [2] while that in Sri Lanka had been around 16.7 TWh [3].

Power system is the field of engineering that deals with electricity generation, transmission and distribution and control the flow of energy in the electricity power network [4]. At the generation, electricity is generated while Transmission carries the electricity to load centers and the distribution sector distributes power to the end consumers. Due to the liberation of the energy market it allowed anyone to buy and sell energy [5]. Driven by the need to reduce carbon emissions, decentralized electricity generation and electrification of the transportation sector the concept of one-way power flowing traditional grid concept had been changed. In order to cater for those challenges, the electric grid concept has been evolved to the Smart Grid which mainly relies on distributed generation focusing on renewable energy generation. Along with that, the concept of prosumers emerged. Prosumers are engaged in consumption and generation at same time requiring two way power flow through the same energy meters [6].

Ever since the concept of electricity grid emerged, electricity metering has been a key component in the electricity market due to the requirement of measuring the service provided. First known energy meter had been patented by Samuel Gardiner in 1872 [6]. Since then, it has evolved to smart meter which is accepted as the building block of smart grid [7]. It has been already accepted that the smart grid is becoming the next version of conventional grid with the capabilities of two-way power and information flow [8]. This implies investigation on electricity metering is a worthy subject which would be the building block of the future conventional grid, that means the smart grid.

There are a number of issues that have been addressed relevant to electricity metering including Consumer billing, Energy theft detection, Demand side management(DSM), Demand Response(DR), Energy Management (EM) and Security. However, addressing the challenge of managing the metering infrastructure is a problem yet to be addressed in the literature [7]. Liberation of the energy market has enabled distributed generation shifting consumers to prosumers. With the involvement of a number of independent entities to the electricity grid, coordination and control of the grid is becoming ever challenging. Due to the capability of Multi Agent Systems (MAS) to coordinate and control a large number of entities, MAS are expected to make the Smart Grid a reality. Inspired from the current employment in CEB, exposure in the considered industry has motivated me to provide a solution to the metering system management issue using multi agent technology.

This chapter presents aims of this study, objectives, background and motivation for providing a solution for electricity metering system, problem definition, brief discussion on Multi Agent System based solution for electricity metering system management, resource requirement and finally the structure of the rest of the thesis.

1.2 Aims

Aim of this project is to design and develop an electricity metering management system using Multi Agent Technology.

1.3 Objectives

This project defines the following objectives for the Multi Agent based solution for electricity meter management.

i. Critical Review of Literature in electricity metering.
ii. In-depth study about technologies used in management of electricity metering.
iii. Design and develop a Multi Agent System based solution for electricity meter management.
iv. Evaluate the Multi Agent System based electricity meter management solution.

1.4 Background and Motivation

Currently, Both the Ceylon Electricity Board (CEB) and the private sector engage in power generation while Transmission is solely handled by CEB. Power distribution in Sri Lanka is handled by CEB and Lanka Electricity Company (LECO). Public Utility Commission of Sri Lanka (PUCSL) functions as the regulator to the electricity industry. For a better overview, the structure of the Sri Lankan electricity industry is given in Figure 1.1 which was extracted from *"Sri Lanka Energy sector assessment, strategy, and road map"* report issued by Asian Development Bank [9].

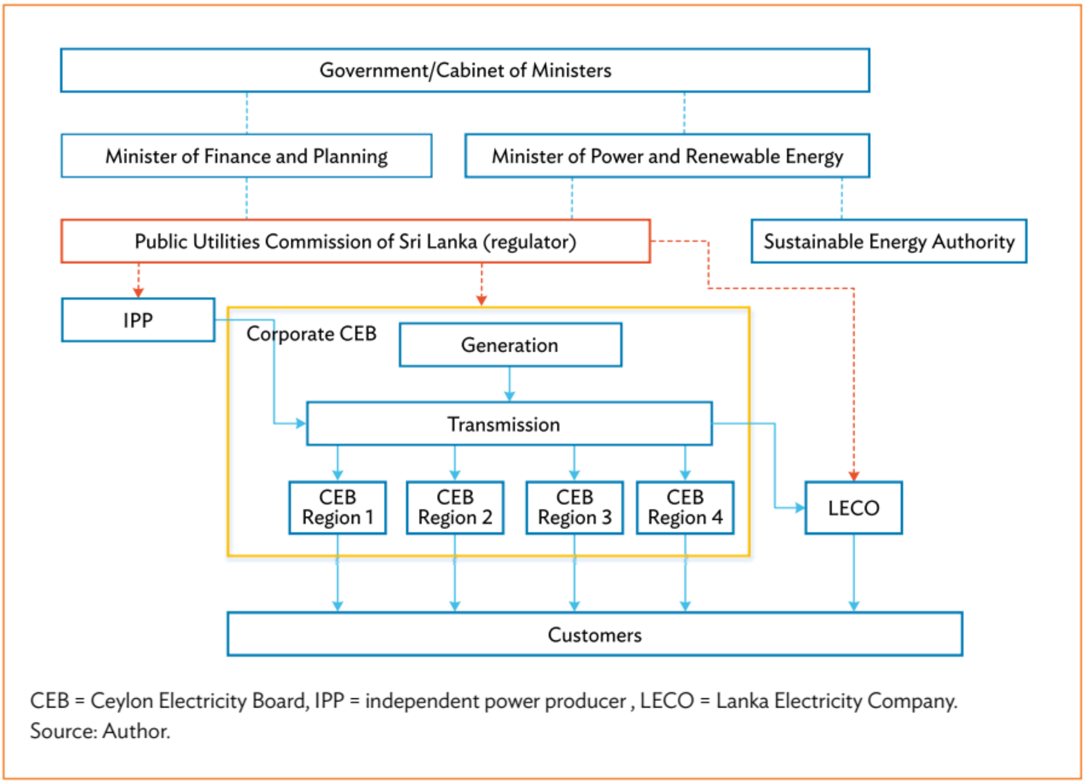

Figure 1.1: Arrangement of Sri Lankan Electricity Industry[9].

Out of the five distribution licenses issued in Sri Lanka CEB holds four licenses while LECO holds one license [9]. By the end of the end of 2022, CEB served 6,936,735 customers while LECO had served 615,647 customers by the end of 2022 [3] [10]. This implies that, more than 7.5 million customers have been served accounting for more than 7.5 million electricity meters in Sri Lanka. By this time, in Sri Lanka, almost all the major types of meters are in use, ranging from old mechanical energy meters to the latest smart meters. Among all the challenges, procuring required software technologies, skilled human resources, finding an amicable solution for the loss of jobs to human meter readers, lack of necessary funds seems to have caused Sri Lanka to make slow progress towards implementing a real Smart Grid.

Key components of a Smart Meter consist of measuring system, communication modules, antennas, power flow controlling relays, primary batteries to assist memory of the meter and displays. Each of these brings more failure points to the metering system. On the other hand, the regulator of the power systems demands ensuring the accuracy of meters regularly on behalf of the public. Even though smart metering helps to cut down the cost of visiting meters for taking readings, at the same time, it increases the risk of energy theft. As the number of meters to be managed increases, it requires more human resources and different strategies to manage different types of meters spread around a large geographic area. Management of the metering system not only includes maintenance activities but also evaluating proposals regarding metering policies which may be either triggered by change of metering technologies, communication technologies or concern on metering accuracy. Being involved in different types of meters for different types of consumers spread in a large geographical area requires a distributed and flexible approach in managing the metering system. Having the number of metering units in millions makes it even difficult for planning of management activities to be done by human beings. Thus, requirement of the computational power of machines to join with human intelligence signifies the use of AI.

1.5 Problem in brief

Literature review provided evidence that the metering system had evolved to provide a number of issues relevant to the modern power system -Smart Grid. However, as the variety of meters increase with the introduction of Smart Meters, managing the metering system has been a challenge that requires a solution for Electricity Metering System Management. Considering the distributed nature, higher number of metering and the possibility of using modeled expert knowledge as an ontology a Multi Agent based Metering System Management solution is explored in this study.

1.6 Agent based solution for Electricity Metering System management

In finding a solution for the management of Electricity Metering Systems, first it was seen as a set of sub tasks that happen in the management process. Those include: planning for annual maintenance activities; evaluation of policy proposals for the changes in metering system; material planning; labor planning; managing vehicles, tools, machinery and buildings like supportive resources. Out of those sub tasks, planning for annual maintenance activities, evaluation of policy proposals for the changes in metering system and material planning were identified as somewhat easily modelable yet difficult to handle by average human beings as the number of meters scale up to several thousands and millions.

When going deeper into these subtasks, it was identified that these problems can be solved easily by modeling the field knowledge as ontology and modeling problem solving knowledge as a Multi Agent Planning task. As the distribution of consumer locations spread to several thousands of square kilometers, scaling up the quantities to several hundred thousand or millions, it requires clustering these consumer meters to areas and managing them in a manner based on location or consumer category. Depending on the location or any other clusters, applying metering policies and private information of the working employees may vary. Therefore, to capture and utilize these differences in policies and planning data at the same time, taking up challenges of scaling up of the quantity and diversity of meters, Multi Agent Planning was found to be a good candidate for providing a solution for this problem.

In developing the Electricity Metering Management system, it utilized the power of OWL ontology to capture domain knowledge and Multi Agent Planning for problem solving tasks. Developing a solution was evaluated in several stages. First, the build knowledge base in ontology was tested with SPARQL queries to test the ability to provide solution, material, time required, assigned employees and also the inconsistencies. Then the Multi Agent Planning system was evaluated agent wise and also the overall solution in three stages. It was found that the developed system can provide a satisfactory outcome.

1.7 System Requirements

In implementation of this solution for the Electricity Metering System Management, following hardware and software had been used.

- Personal computer with Core i5 processor running windows operating system.
- Microsoft VSCode
- Python programming language
- Protégé ontology editor

1.8 Structure of the thesis

This thesis is structured in seven chapters. Chapter 1 contextualizes research work describing problem background and motivation, defining problem, aims and objectives as well as briefing the platform for implementation of the solution. Chapter 2 is the literature review that briefs the evolution of energy metering, significant research work happened under the umbrella of electricity metering and the problem definition of this study. Chapter 3 discusses the Ontology and Multi Agent Systems which are the technology used for electricity metering system management development. Chapter 4 describes the approach of Multi Agent System used for metering system management. The design of the Multi Agent System based solution for metering system management is explained in Chapter 5 with relevant design diagrams. Designed solution is implemented using the OWL ontology and SPADE module in python. The implementation has been detailed and discussed in Chapter 6. Chapter 7 discusses the results of the implemented solution for Electricity Metering system management while Chapter 8 provides the conclusion of research work and direction for the future work.

1.9 Summary

In this chapter, we narrowed down the problem domain to Electricity Metering System Management and briefly discussed the research work that has happened in the considered domain. Also, the aims and objectives of this study with the resources required for implementing the designed system was briefed. At the end of this chapter, the structure of the rest of the document is also provided. The next chapter is the literature review, and it will be discussing the research work that has already been done in the field of Electricity Metering.

CHAPTER 2

LITERATURE REVIEW OF ELECTRICITY METERING

2.1 Introduction

In the previous chapter, it was defined the aims and objectives of the study, discussed the background and the motivation of the study, briefed the problem, briefed the Multi Agent based solution, identified the resource requirement and finally described the structure for the rest of the document. In this chapter, it is critically reviewed the research work that has been carried out in the domain of electricity metering highlighting their contributions and areas that require further research work. At the end of this chapter, the research problem is also defined.

2.2 Smart Grid

Electricity is a form of energy that is widely used for lighting, cooking, cooling and heating, communication, entertainment, driving industrial machines and mobility. As a sub field of electricity, Power System is identified as the field of engineering that deals with electricity generation, transmission, distribution and control the flow of energy in the power network [4]. Generation is referred to as a subsector of the power system that is used to generate electricity at larger power plants. This bulk generated electricity had been collected by the transmission network and bulk transported to the load centers for the distribution of electricity to end consumers by the power distribution sector. These electricity grids were first introduced in the 1800s even though it took until the 1960s to widely adapt this concept [7]. This concept is now recognized as the traditional grid where the electricity is generated in a centralized manner catering consumers through a one-way power floor.

The climate change issue has triggered the changes in the paradigm of the traditional grid due to the requirement of reducing carbon emission, decentralization of energy generation and electrification of transport sector [6]. Decentralizing electricity generation allows to harvest renewable energy even at a small scale wherever it is available. Because of this reason, the energy market has been liberated allowing anyone to buy and sell energy [5]. This upgrade of traditional grid to an intelligent grid that includes distributed energy production, integrating intermittent nature of renewable energy, battery storages, electrical vehicles, intelligent building and other types of consumers are known as the Smart Grid. The main objectives of the smart grid encompass improving grid efficiency, reducing carbon emission, improving power system security and reliability[8]. Smart Grid technology is considered as an integration of information technology, communication technology and power system engineering [8]. Smart Grids are sometimes referred to as Intelligent Energy Networks (IENs) and smart energy meters are recognized as the building blocks of them [7].

2.3 Multi Agent system with ontology in electricity industry

Multi Agent Systems are good candidates for modeling systems that involve interaction of different individuals. However, use of different languages and vocabularies have led to numerous issues in interpreting messages in agent communication making the interoperability of agents a challenging task. There are standard Agent Communication Languages to facilitate meaningful conversation among agent. Yet, it alone will not guarantee that two multi agent systems or agents will share any meaningful information if they use different ontologies in communication. Multi Agent systems together with ontology had been used in many domains of the electricity industry including Power system, Energy Management, Demand Side Management, Grid Control, Electricity Market and Micro grids [11].

A framework called TOOCC has been developed for simulating and analyzing energy systems integrating different multi agent simulation systems using public ontologies for the interoperability of these systems. Purpose of using ontology has been to provide common vocabulary among different systems. The developed tool has been tested using data in a University of Portugal. This decision support system has been able to jointly simulate scenarios combining multiple systems relevant to the Energy Management[12]. In another study ontology driven MAS based energy management solution has been developed for real-time monitoring and controlling a micro grid. The agents have been allocated for central coordination, building management, meteorological information sharing, Battery management and providing computation services to other agents. These agents have assigned the goals of cooperatively achieving goals of continuous supply of system goals, minimizing energy cost of residential and maintaining resident’s comfort levels. In this development ontologies have been used to provide meaning to the messages passing between agents. The developed solution has been evaluated using simulations and reducing system’s operating cost while ensuring user needs have been proven under different weather and pricing conditions [13]. Optimum dispatch of Integrated Energy Sources and Demand Response in a building have been explored in a study using ontology driven Multi Agent technology. Ontologies have been used for communication among agents providing meaning to the symbols used. The proposed model has been tested using computer simulations. The test results have the proven proposed model’s capability in reducing domestic energy demand while improving user satisfaction. For this, the model utilized coordinated demand response and distributed energy generation management with the system’s constraints [14]. Using ontology for Multi Agent based Electricity Market has also been studied. One of the publicly available such ontology is Electricity Market Ontology (EMO). An extension of EMO called EPX has been developed with the introduction of “BlockOrder” for EPEX market for the interoperability of agents. The concept had been tested using Multi-Agent Simulator of Competitive Electricity Markets (MASCEM) modeling and simulation tools. Scenarios for the case study had been developed by RealScen called tool that use data extracted from several European market operators [15].

2.4 Evolution of Electricity Metering

The first known electricity meter had been patented by Samuel Gardiner in 1872 that counted time for assessing energy supplied for street lamps. Later, in 1881, Thomas Edison had patented his first electricity meter based on the electromechanical effects of DC current. As time passed, the AC power system emerged and in 1889 Otto Titusz Blathy had patented an induction electricity meter for measuring AC power. Since the second half of the 20th century electronic meters have come to the market evolving up to the smart meter that is used today [6]. As the meters are innovated the metering systems also have been evolved. The meter reading paradigms had been evolved as Manual Meter Reading (MMR), Electronic Meter Reading (EMR), Automated Meter Reading (AMR) and Advanced Metering Infrastructure (AMI). In MMR, meter readers physically visited the consumer premises and readings were taken. With the advancement of electronic meters up to level they can communicate in Radio Frequencies (RF) through RF transmitters, meter readings were gathered using RF receivers either with a hand held device, installed in a vehicle or tower based radio networks. As the time passed, the whole billing system had been automated by coupling the billing with meter reading systems that utilized RF networks, Power Line Carriers or telephone networks which is called AMR [5]. The latest paradigm of metering system, which is called AMI, combines software systems, database management, communication, monitoring systems and Smart Meters (SM). Smart Meters are a component of AMI and both the SMs and AMI are considered as key components in Smart Grids [6].

2.5 Advance Metering Infrastructure

Smart Grid is sometimes referred to as a soft grid that combines many applications. The key applications of SGs are management and control of power storage, providing service to electrical vehicles, distributed management of power distribution systems, demand response for managing electricity usage by consumers during peak hours, customer services management and AMI [16]. In the scope of AMI many issues have been already addressed. Those include Consumer billing, Energy theft detection, Demand side management(DSM), Demand Response(DR), Energy Management (EM) and Security [7].

Concept of Demand Side Management (DSM) dates back to the 1970s and encompasses areas such as flexible load curves, peak clipping, valley filling, load shifting, strategic conservation and strategic load growth. Demand Response (DR) is considered a subset of DSM strategies. It has been defined as "*programs and activities that decrease peak demand using advanced metering, dynamic pricing and supporting technologies*" [17]. Demand Response strategies primarily focus on reducing overall power system cost including cost of power system operator, cost of consumer, reducing environmental damage, reducing carbon emission and greenhouse gas emission. DR programs had been categorized to two; direct load control and price based control. In direct load control, the utility has control over some of the consumer loads and switching them off to keep the power system stable. This approach seems to be outdated due to the drawback of compromising consumer welfare. On the other hand, price base control schemes have emerged to promote customers to actively engage in demand control. Price base control schemes are categorized as incentivize schemes, Time of Use schemes, Real Time Pricing, Customized Pricing and dynamic pricing [18]. These pricing schemes also fall under solving customer pricing problems as an AMI application.

Energy Management (EM) systems that evolved from Control Centers emerged in the 1960s for balancing supply and demand. Nowadays, EM function is focused on optimal allocation of different power sources, both renewables and non-renewable, for the customers while ensuring power system reliability, security and safety. It can be seen that some of DSM activities are also coming under EM systems [19].

Energy thefts are happening at a number of places and in a number of ways on the power systems. Those thefts are mainly happening as meter tampering, feeder tapping, bill alteration and Cyber-attacks. These energy theft detections have also been widely studied and provided solutions utilizing a number of techniques which are coming under Support Vector Machine based techniques, Artificial Neural Network based techniques, Fuzzy logic based techniques, State estimation techniques, Static Game theory and Dynamic Game Theory techniques [20].

As the traditional grid is moving to a Smart Grid with the implementation of AMI, it heavily uses the internet for collecting, processing and transferring data exposing the power system to many more security threats than ever before. From the power system security point of view, AMI architecture can be described in three layers, namely; hardware layer, data layer and communication layer. In all of these layers of AMI, there exist a number of vulnerabilities. These vulnerabilities expose AMI to various attacks including; denial of service, sending malicious codes to affect firmware of Smart Meters, injection of false data to the system, manipulation of data in utility data centers, data theft while data transmission and interference to communication channels. Most of these attacks have been studied and counter measures have been already given [21].

As the AMI scaling up it has to ensure interoperability of smart meters and other legacy equipment. Literature identifies maintaining a large number of energy meters would be a challenge in the future and it is an open issue for the moment [7].

2.6 Problem definition

Electricity infrastructure is changing from traditional grid to smart grid making both the electricity and communication flow in two ways. With the transformation from traditional grid to smart grid, the role of the electricity meters has evolved from a device that is used only for electricity measurement to a more complex device that measures electricity, communicates in both directions and controls the power flow.

Because of this transformation of electricity meters, concerns on electricity meters are increasing to legal requirements, measuring requirements, communication requirements and power system controlling requirements making the management of electricity metering systems more complex. Literature highlighted that even though there are a number of issues already solved, managing a larger metering system is a challenge that requires a solution. Hence, the research problem is defined as how to manage the electricity metering system using Multi Agent technology.

2.7 Summary

This chapter critically reviews the research work that has been undertaken in the problem domain of Electricity Metering. The contributions, used technologies and the open problems in the domain have been discussed and summarized. At the end of this chapter the research problem has also been defined. In the next chapter, that is Chapter 3, it will be discussed in depth about technologies utilized for developing a solution for the Electricity Metering System management problem.

CHAPTER 3

TECHNOLOGY ADOPTED FOR ELECTRICITY METER MANAGEMENT SYSTEM

3.1 Introduction

In the previous chapter it was reviewed the research work done on Electricity Metering. The scope covered by the previous research works had been discussed finally defining the research problem. Due to the concerns of global warming, the traditional electricity grid is moving from unidirectional power flow to bidirectional power flow allowing consumers to become prosumers in a Smart Grid. AMI and Smart Meters have been identified as the key components in the Smart Grid. Number of studies had done different aspects of the AMI. However, having to manage a large number of meters is considered as a future problem. In this study managing metering systems is viewed as a collection of maintenance activity planning and evaluation of impact on policy changes. This involves reasoning about maintenance activities before doing. Therefore, the problem has been modeled as a Multi Agent Planning (MAP) problem in the domain of Multi Agent Systems (MAS) that involves a number of entities for the process of planning with different knowledge and authorities. The reasoning part of the planning is powered with knowledge representation of suitable maintenance activities for issues as OWL ontology. There are two key AI technologies involved in this study, Multi Agent Systems and Ontology. In this chapter, it is discussed in depth about the used technologies; Multi Agent Systems and ontology. An overview of Multi Agent Technology, Multi Agent Planning and ontology as a technology for knowledge representation will be discussed throughout this chapter.

3.2 Overview of Multi Agent System (MAS)

Multi Agent Systems (MAS) is a problem solving technique that solves a complex task by subdividing the task into smaller tasks and assigning them to autonomous entities known as agents [22]. MAS are coming under broad field Distributed Artificial Agents (DAI) that encompasses the AI techniques used for complex computing problem solving. Under the DAI, it can be found three fields based on the methods used to solve complex problems: Multi Agent Systems (MAS), Distributed Problem Solving (DPS) and Parallel AI (PAI). PAI is about using task parallelism to develop algorithms, architectures and languages to increase efficiency of classical algorithms. DPS divides tasks into subtasks and then allocate to a set of cooperative computing entities (referred to as nodes) that share knowledge, resources and pre-defined communications with limited flexibility. At the same time a bit similar to DPS, MAS solve complex problems by using autonomous entities known as agents that contrast to nodes in DPS due to flexibility inherent from learning and autonomous decision making [22].

Main constituent of the MAS is Software agents that have been evolved from Multi Agent Systems (MAS). Power of MAS seems to be discussed with the smartness of the agents that had been derived from creating a greater value by coordinating their actions and cooperation than the value gained as individual stand-alone agents [18]. Agents in MAS stand out compared to other computing entities in DAI due to their ability to learn new context and actions from the other agents and environment. Then these agents use gathered knowledge to decide and perform action on the environment to solve the assigned task [22].

Agents are the building blocks of Multi Agent Systems. By the mid of 1990s, the term "agent" seems to have been considered an umbrella term or class that encompasses different types of agents due to the difficulty of precisely defining the term "agent"[23]. However, even though it has been still a generic term that applied in many disciplines by 2016 research works have contributed to narrow down the definition of agent: *An **entity** which is placed in an **environment** and senses different **parameters** that are used to make a decision based on the goal of the entity. The entity performs the necessary **action** on the environment based on this decision"*[22]. This definition consists of four key words, **Entity**: the type of agent, **Environment**: place where agent is located, **Parameters**: types of data that the agent can sense from the environment (including those from other agents), **Actions**: actions performed by agents that result in changing environment [22].

Agents do have features that make agents capable of solving complex problems and broaden applicability. Those are **Sociability**: to improve performance of agents by sharing their knowledge and requesting information from other agents to reach goals, **Autonomy**: agent's capability to make independent decisions and take appropriate actions, **Proactivity**: ability of agents to use its history, sensed parameters, and information of other agents to predict the possible future actions. Even though agents can make their own decisions and take actions due to their autonomy, the real benefits of agents are harnessed by employing them to collaboratively solve a complex task known as Multi Agent Systems(MAS) [22]. Due to the capability of MAS to model and solve complex tasks, it has evolved into a number of domains. For the purpose of solving the problem of managing Electricity Metering System, here onwards, it would be narrowed down to the technology Multi Agent Planning; which is a subdomain of MAS and at the same time that of AI Planning.

3.3 Multi Agent Planning

Planning deals with an explicit deliberate process that chooses and organizes actions by anticipating their outcomes aiming at achieving some expected objectives. Automated planning or AI planning, a subset of planning, deals with the reasoning side of acting. As the planning knowledge or the responsibility distributed among agents, deviating from the classical planning or which may refer to as single-agent planning to multi Agent planning has emerged along with the distributed problem solving concept. Concept of distributed planning are twofold: Cooperative Distributed Planning, known as cooperative and distributed Multi Agent Planning (MAP), where planning and execution are distributed among multiple participants while on the other hand Negotiated Distributed Planning focus only on coordination and scheduling action among multiple agents. Where there may be multiple definitions, the Multi Agent Planning tasks are found to be defined as *"collective effort of multiple agents toward achieving a common goal, irrespective of how the goals, the knowledge and the agents' abilities are distributed in the application domain"*. In the domain of Distributed Artificial Intelligence (DAI), MAP has to deal with reasoning and coordination of planning activities among agents to come up with a global plan [24].

As one of MAS, MAP also has certain features that contrast one planner from another. Those are; Agent Distribution, Computational process, Plan Synthesis Scheme, communication mechanism, heuristic search and privacy preservation. In the context of MAP, Agent Distribution is twofold. It is either planning for multiple acting agents by one agent or planning by multiple agents. In the first approach it lacks privacy while in the second approach privacy can be preserved and great focus is going for coordination of the planning activities. Same as Agent Distribution, Computational processes are also twofold. From the computational point of view, it recognizes MAP solvers as Centralized MAP and Distributed MAP. In centralized MAP all planning computation happens in one machine as a single agent planning where it does not require communication among agents for planning activity coordination. On the other hand, Distributed MAP relies on communication among agents for planning activity coordination (planning synchronization) [24].

Plan Synthesis Schemes are focused on coordination of planning activities. In most of the MAP tasks, none of the agents has sufficient competence, resources, authority or information to do the planning for the entire task. Therefore, it requires multiple agents to participate in planning and coordinate planning activities. Coordination strategies are also twofold; Unthreaded Planning and Coordination or Interleaved Planning and coordination. In Unthreaded strategy planning and coordination activities are happening as two separate tasks in sequential manner, either as Pre-Planning Coordination or Post-Planning Coordination. Pre-Planning Coordination does the coordination activities before planning requiring no plan merging technique as post-planning activity while Post-Planning Coordination requires plane merging technique for removing inconsistencies among local solutions of agents as it does the coordination after distributed planning. In contrast to Unthreaded Planning and Coordination, Interleaved Planning and Coordination interleave the planning and coordination activities making it more suitable for complex tasks with group goals and higher coordination [24].

Communication mechanisms can be either Internal or External. Internal Communication is implemented among the agent that runs on the same machine which can be a simulated multi-agent communication also. External communication happens when the planning agents have been deployed in different machines connected through either network sockets or TCP/IP protocols which require message passing protocols such as IEEE FIPA [24].

Heuristics functions provide informative estimates of cost of reaching goals in goal searches. Heuristics can be either local or global. Sometimes, heuristics are kept local to improve privacy. Privacy deals with preservation of sensitive information of agents. Privacy preservation schemes can be either Induced or Imposed. In Induced privacy preservation, planners automatically identify the private elements of the planning task by nature while Imposed privacy schemes explicitly describe private and public information parts [24].

3.4 Introduction to Ontology

Ontology is a knowledge representation technique that had been introduced as a theory about the nature of existence by the Greek philosopher Aristotle [20]. Ontology has many definitions. In the Artificial Intelligence community, ontology is identified as a *"knowledge representation technique that describes entities and their properties, relationships, constraints and behavior that are not only machine-readable but also machine-understandable"*. In computer science, ontology is defined as " *a formal, explicit specification of shared conceptualization*"[11].

There are functions of ontologies: providing common glossaries to communicate among different individuals; supporting interoperation among various modeling methods, languages and software; acting as a reusable knowledge representation technique; separating domain knowledge from operational knowledge and supporting knowledge acquisition and sharing [25].

There are many languages to represent knowledge in the form of ontology. Ontology Web Language (OWL) is one of them and has been widely used. OWL is an implementation of Descriptive Logic which is a subset of First Order Logic. In this project, OWL will be used as ontology languages and Protégé would be used as the tool for developing ontology. Ontology uses Classes, Properties and individuals to represent knowledge. Classes are used to represent concepts in a domain and Properties describe various features and attributes of the concepts and Axioms are used as rules to define classes. In OWL ontology subclasses are used to specialize its superclass. Individuals represent the objects in the domain. Properties can also be seen as relationships between individuals. There are three types of Properties in OWL; Object Properties, Data Properties and Annotation properties. Object Properties are used to describe relationships between individuals while Data Property relates individuals to data types. Annotation Properties are used as descriptions of entities and cannot be reasoned about [25].

3.5 Protégé as a tool to manipulate ontology

Protégé is a free and open source ontology editor developed by Stanford University. It supports a number of ontology languages including OWL, RDF, and XML. In addition to editing ontology, Protégé provides reasoners as plugin tools to support automated reasoning, finding inconsistencies and errors. It also provides a visualization tool for ontologies and a facility to query the built ontologies. Protégé provides both web based ontology editing tools and desktop tools. However, Web Protégé does not yet support reasoners and Desktop Protégé was used for this project [25].

3.6 Summary

This chapter discussed the theoretical background of the technology used for the study- Multi Agent Systems and Ontology. Next chapter is about the approach for a Multi Agent System based solution for Electricity Meter Management that discusses the hypothesis, input, output, process, users and the features of the solution for the problem under study.

CHAPTER 4

APPROACH OF MULTI AGENT BASED ELECTRICITY METER MANAGEMENT SYSTEM

4.1 Introduction

The earlier chapter provided an in depth discussion on Multi Agent Planning under Multi Agent Systems and Ontology; those are the adapted technologies in providing solutions for Electricity Metering System Management in this study. In this chapter, it is described the approach of the Multi Agent System based Electricity Meter Management System under the headings hypothesis, inputs, outputs, process, users and features.

4.2 Hypothesis

The hypothesis of this project is that the activities of electricity metering system management can be modeled using Multi Agent technology. Inspiration behind this hypothesis came from the power of multi agent system's ability to model coordinated actions of autonomous entities in a complex environment to achieve a common objective.

4.3 Inputs for the Electricity Metering Management System

Inputs for the considered system are csv file format or OWL ontology. Those are meter database, stock level of required materials, locations of the consumer premises and utility base stations, working calendars of assigned employees who are implementing plans and finally the expert knowledge as OWL ontology.

4.4 Outputs

The designed system is capable of providing the output of maintenance activity plan, material plan and also an activity schedule. These activity schedules consist of maintenance activities that have been planned on a particular consumer electricity meter, assigned employee, planned date, required material and also time required for those activities.

4.5 Process

The process of the solution given for the Metering Management system has been developed with the separation of the operating knowledge from the domain knowledge. Domain knowledge or the expert knowledge in the field has been captured into an OWL ontology. SPARQL queries have been used for the reasoning part from the ontology. Separation of domain knowledge from the operating knowledge enables evaluating proposed policies without any modification to the system. Operating knowledge was captured into a Multi Agent Planning system which is an adaptation of Multi Agent System. Overall process diagram of the Electricity Meter management system is shown in Figure 4.1. The Multi Agent Planning system was built similar to the duties of the employees involved in metering system management. These agents include Management Agent (MA), Engineering Agent (EA), Store Keeper (SK) and Electrical Supervisor agent (ES). MA agent receives the instructions for a task. This could be preparation of an annual maintenance plan, evaluation of a policy which may come as ontology to represent a new domain knowledge case or material plan. Once the MA agent receives instructions it passes the instruction to the EA agent. By utilizing the domain knowledge base developed as an ontology it uses SPARQL queries to find solutions for the issues in the metering system with required material and relevant employees which may put into his activity plan. Based on the received instruction, if it is required, EA agent calls for SK agent and EA agent prepares the material plans and activity schedules respectively using the maintenance activity plan that he has already prepared.

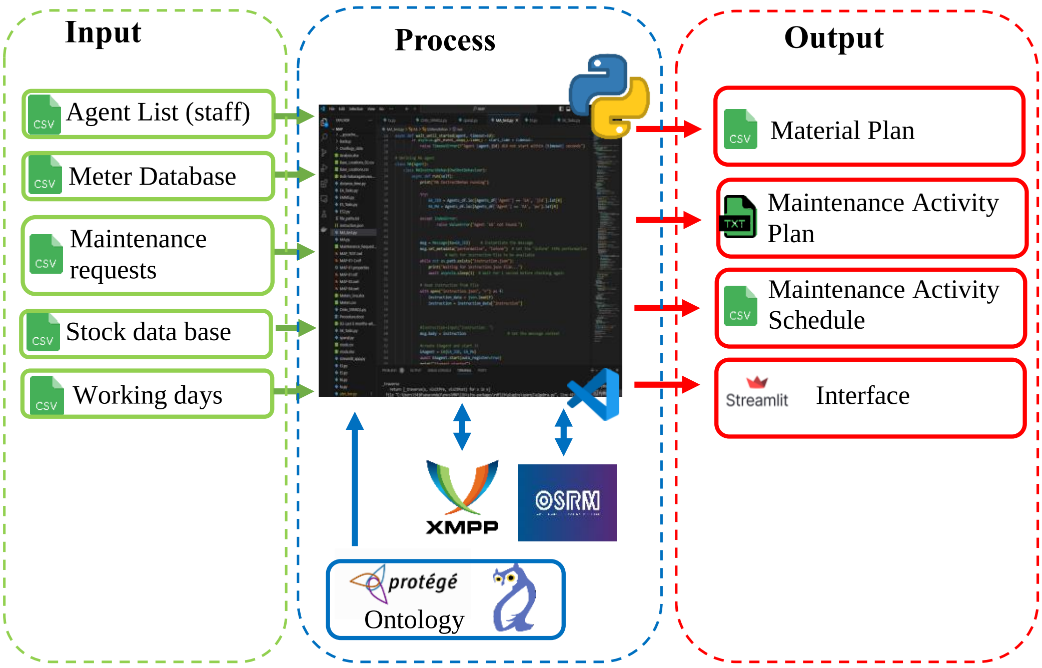

Figure 4.1: Overall Process of Electricity Meter Management System.

4.6 Features

It has the features of metering system maintenance activity scheduling, feasibility studies for policy changes and material plan.

4.7 Users

This system is expected to be used by power distribution system officers, mostly the power system engineers in power distribution sector.

4.8 Summary

This chapter introduces the approach of the MAS based Electricity Meter Management System discussing hypothesis, inputs, outputs, process, features and users. Expanding the concept in approach, the next chapter in detail discusses the design of the Multi Agent System based Electricity Meter Management System.

CHAPTER 5

DESIGN OF MULTI AGENT ELECTRICITY METER MANAGEMENT SYSTEM

5.1. Introduction

In the previous chapter, the approach of the MAS based maintenance scheduler was discussed including hypothesis, input, output, process, features and users. This chapter in detail discusses the experimental design including overall arrangement of agents in MAS based Electricity Meter Management System describing design of each type of agent in the system.

5.2 Top Level Design

The first step in providing a solution for Electricity Meter Management was to study the nature of the problem. The top level task of the problem, that is management, brought down to subtasks of planning for annual maintenance activities; evaluation of policy proposals for the changes in metering system; material planning; labor planning and managing supportive resources like tools, machinery and buildings. As it is not possible to provide a single solution for all of these sub tasks, only few of the important tasks that human employees require power of computing with simulated human intelligence in machines were modeled in this design. Those considered subtasks are planning for annual maintenance activities, evaluation of policy proposals for the changes in metering system and material planning.

In the real world, these considered tasks were found to be done by further subdividing to subtasks of finding solutions for the inconsistencies in metering system compared to metering policies and domain knowledge, planning maintenance activities, preparing material plan and also scheduling planned activities. These subtasks are naturally divided among employees who are in different levels of hierarchy with different accountabilities (goals or duties) and authorities (autonomy). This gave the inspiration to model the problem solving part according to a hierarchy. On the other hand, this planning and scheduling are done for geographically distributed electricity meters which requires localized knowledge and personal data of plan implementing employees. When this distributed authority, autonomy, localized knowledge and privacy of personal data was considered, Multi Agent Systems (MAS) technology was considered for designing the solution. When modeling agents, it was observed that the nature of tasks that are done by the agents are planning tasks. Hence, the modeling of problem solving adopted the Multi Agent Planning (MAP) concepts. Further, modeling of planning tasks inherits the characteristics of Hierarchical Task Network (HTN) Planning also.

Inherently, activities carried out in the operation of a power system must be explainable due to the danger of electricity. Therefore, it is better to design the solution in a symbolic manner. It was found that the knowledge used in these tasks can be divided into domain knowledge and problem solving knowledge. Domain knowledge includes consumer categories, required meter testing frequency for them, suitable meter types for them, issues found with meters, solution for the meter issues, material required for solutions, approximate time required for attending to those solutions which are called activities and also the employees responsible for meter maintenance activities in a particular geographic area. The domain knowledge was modeled in an OWL ontology. Next challenge is modeling the problem solving knowledge, which was modeled and built into agents as planning and scheduling tasks as explained briefly above.

Overall design consists of ontology for domain knowledge and MAS for problem solving. The agents in the designed system are Management Agent (MA), Engineering Agent (EA), Store Keeper (SK) and Electrical Supervisor agent (ES). These agents are depicted in Figure 5.1. MA agent act as the interface agent that receive the instruction for management task to be attended. That may be either preparation of an annual plan, evaluation of policy which may come as ontology to represent a domain knowledge case or material plan. By utilizing different ontology proposed policy changes in the metering system can be evaluated by preparing an activity plan utilizing existing employees while attending to routine and corrective maintenance issues.

Once the MA agent receives instruction for a particular task it passes the instruction to the EA agent. EA agent utilizes the domain knowledge developed as an ontology to query using SPARQL and find solutions for the issues and inconsistencies in the metering system (Meter Database as in Figure 5.1) with required material and responsible employees for actions. Based on the received instruction, if it requires, EA agent calls for SK agent and then EA agent prepare material plans and activity schedules respectively using the maintenance activity plan that he has already prepared. Detailed design of Metering Ontology and the agent wise designs will be discussed in upcoming subsections while the specifications of agents are given below.

MA agent: The role of MA agent is to interact with users and get their requirement and pass the instructions to EA agent. MA agent is capable of sending messages through XMPP server and read user instruction from Streamlit interface. The agent can monitor the status of EA agent and terminate own operation once EA agent stopped.

EA agent: This agent's role is to prepare maintenance activity plan, allocate tasks to SK agent and ES agents and combine maintenance schedules prepared by ES agents based on the instruction received from MA agent. EA agent can read meter database, maintenance request and interact with OWL ontology files which represent the metering policies and field knowledge. EA agent can communicate with SK agent and ES agents via a XMPP server to assign tasks. This agent can sense the status of SK agent and EA agents. EA agent will terminate once the subordinate agents stop after assigned works are completed.

SK agent: The role of SK agent is to prepare material plans that include material demand for planned maintenance activities and prepare material order quantitates. SK agent will read Maintenance activity plan and stock database with minimum stock levels and minimum order quantities. This agent can receive instructions as messages from EA agent through a XMPP server.

ES agent: The role of this agent is to prepare maintenance schedule for the maintenance activities assigned to him. This agent can access their own working calendar which is shared only with them, read assigned maintenance activity plan and interact with locally running Open Source Routing Machine (OSRM) server to get the traveling distance and time between two locations. This agent also receives instructions as messages from EA agent through a XMPP server.

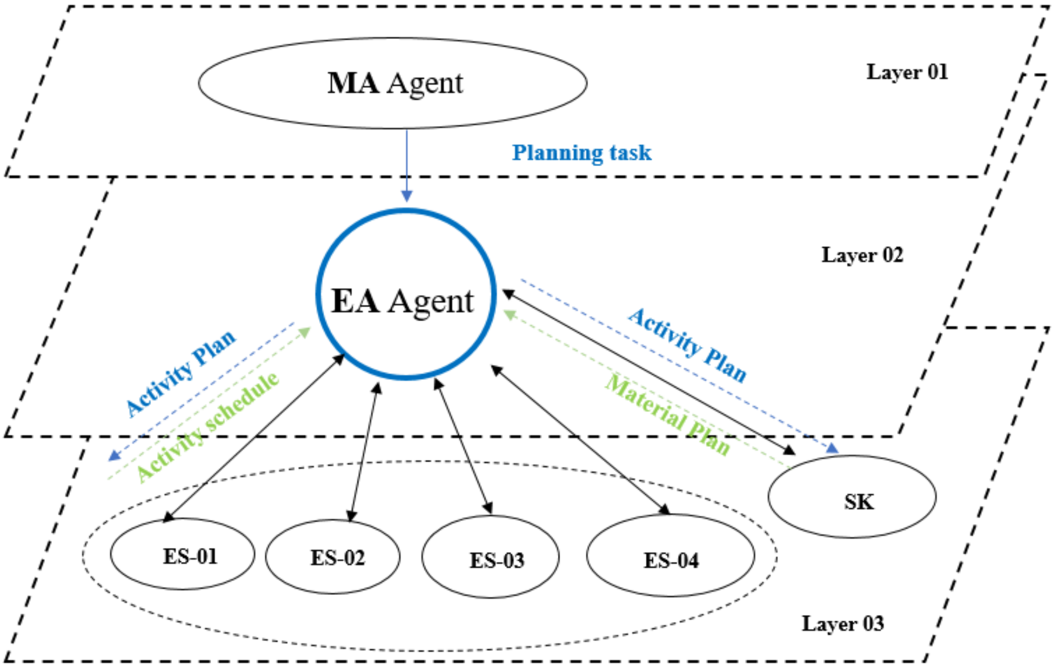

Figure 5.1: Overall design of Electricity Meter Management System.

5.3 Ontology based Knowledge representation

The knowledge required to reason about required planning activities in the Electricity Metering system are conceptualized into the classes of the ontology. Those classes are; ACCURACY_CLASSES, AREAS, CONSUMER_CATEGORIES, DISCO_STAFF, ISSUES, MAINTENANCE_ACTIVITIES, MATERIAL, METER_MODELS and TEST_FREQUNCIES. "CONSUMER_CATEGORIES" class can be identified as the key class in this ontology as the selection of the meter model and then required maintenance activities are based on that. First use of the ontology is to specify the policy requirements on metering. The relevant policy elements for this case are accuracy class and test frequencies for a particular consumer category. These policy requirements have been represented by relating individuals in "CONSUMER_CATEGORIES" classes to specific individuals in "ACCURACY_CLASSES" via object property "has_accuracyClass" and required test frequency via data property "rest_frequency" respectively. While doing maintenance activities certain materials have relevance to a particular consumer category and those relationships are represented using object property "related_material" to the related material in "MATERIAL" class. Combining both policy requirements and technical requirements, suitable meter models are given in the ontology mapping consumer categories to individuals of "METER_MODEL" through the object property "has_meterModel".

The second requirement of this ontology is to identify required maintenance tasks and related material for those activities. For the identified issues, required maintenance tasks have been given via the "has_solution" object property which relate each individual issue in “ISSUES” class to the correct individual maintenance activity in “MAINTENANCE_ACTIVITIES” class. For the purpose of planning the number of consumer premises visited for maintenance activities, required time for each activity is given through a data property “required_time”. Each maintenance activity has its own priority requirement to attend and those are represented in the ontology through data property “due_days”. For the planning of material requirements, each maintenance activity is related to the individual of “MATERIAL” class via the object property “need_material”. It is to be noted that for the same maintenance activity there are several materials of the same type which have to be specific on type of the Electricity Meter. Therefore, to get the correct material it has to match both maintenance activity and consumer category relationships with the individual in “MATERIAL” class.

Relevant to managing metering systems certain issues have to be identified by the human staff while some may be identified by the system through checking consistency. Some of these system identifiable inconsistencies are test date and the meter model issue. Those can be identified by matching the consumer category to the applicable meter model and test frequency for each instance of the metering database.

The third requirement of ontology is to find the relevant staff member depending on the physical area of the consumer. This knowledge had been embedded in the ontology by relating individuals in the “AREAS” class to “DISCO_STAFF” class which represent the relevant staff members in the power distribution sector.

5.4 Design of Multi Agent System

Problem solving part of the developed system was designed as a Multi Agent Planning task. The agents in the system would be as MA agent, EA agent, SK agent and ES agents. It basically utilizes decomposition of planning tasks by EA agent and assigning to SK agent and ES agents. EA agent, ES agent and SK agent have separate functions for their implementation and they are called within the definition of them in the overall Multi Agent system. The algorithm for the overall implementation of the agents would be as follows.

Define wait_until_started() function:

- Get the event loop time as start_time
- While agent is not alive
    - Sleep for 0.1 seconds

Defining MA agent:

- Define MA agents Instruction behavior:
    - Get the ID and pass word for EA agent
    - Wait for instruction from “instruction.json” file
    - Read instructions from “instruction.json” file
    - Create EA agent
    - Start EA agent
    - Wait until EA agent is started

        - Send the instruction to EA agent
        - Wait unit EA agent stop
        - Stop the agent itself
- Define setup function of MA agent:
    - Add MA instruction behavior to MA agent

Define EA agent:

- Define EA agents Receive behavior:
    - Prepare ES list
    - Get the IDs and Pass Words of ES agents in ES list from Agents_df to ES_df
    - Get the IDs and Pass Words of SK agents from Agents_df
    - Wait for 10 seconds to receive a message
    - If message is “Annual Plan”
        - Prepare Maintenance plan
        - Allocate Task for ES agents
        - Create SK agent
        - Start SK agent
        - Pass instruction to SK agent for creating “Material Plan”
        - For ES agents in ES list
            - Get the ID and Pass Word from ES_df
            - Create ES agent
            - Start ES agent
            - Wait until ES agent start
            - Send instruction of Activity Schedule to ES agent
        - Wait until the last ES agent stop
        - Wait until the SK agent stop
        - Combine Schedules prepared by ES agents
    - If message is “Material Plan”
        - Prepare Maintenance plan
        - Create SK agent
        - Start SK agent
        - Pass instruction to SK agent for creating “Material Plan”
        - Wait until the SK agent stop
    - If message is “Policy Evaluation”
        - Prepare Maintenance plan
        - Allocate Task for ES agents
        - Create SK agent
        - Start SK agent
        - Pass instruction to SK agent for creating “Material Plan”
        - For ES agents in ES list
            - Get the ID and Pass Word from ES_df

        - Create ES agent
        - Start ES agent
        - Wait until ES agent start
        - Send instruction of Activity Schedule to ES agent
      - Wait until the last ES agent stop
      - Wait until the SK agent stop
      - Combine Schedules prepared by ES agents
- Define setup function of EA agent:
  - Set message template of EA agent
  - Add EA Receive behavior and message template to EA agent

Define SK agent:

- Define SK agents Receive behavior:
    - Wait for 10 seconds to receive a message
    - If message is "Material Plan":
      - Prepare Material Plan
    - Else print not receiving any instruction
    - Stop SK agent itself
- Define setup function of SK agent:
    - Set message template of SK agent
    - Add SK Receive behavior and message template to SK agent

Define ES agent:

- Define ES agents Receive behavior:
    - Wait for 10 seconds to receive a message
    - Extract ES designation and Instruction from the message
    - If Instruction is "Activity Schedule":
      - Prepare Activity Schedule
    - Else print not receiving any instruction
    - Stop ES agent itself
- Define setup function of ES agent:
    - Set message template of ES agent
    - Add ES Receive behavior and message template to ES agent

Define main() function:

- Start Streamlit subprocess
- Get the ID and pass word of MA agent from Agents_df
- Create MA agent
- Start MA agent
- Wait until MA agent stop

- Remove “instruction.json” file from directory
- Print “Agents finished”

Run main() function.

5.5 Design of Streamlit interface

Using the Streamlit python module user interface was developed to receive instruction from users and also to displace input data and output plans. The algorithm of the streamlit interface would be as below.

Define List_Files(directory) function:

- Initialize an empty list file_paths.
- Traverse the directory tree starting from “directory” using os.walk.
- For each file in the current directory, construct the full file path and append it to file_paths.
- Return the list file_paths.

Define Display_Files(file_path) function

- For each file path in file_paths:
  - Write the file path using Streamlit (st.write).
  - If the file is a CSV file (file_path.endswith('.csv')):
    - Write "Contents of CSV file:".
    - Read the CSV file into a DataFrame using pd.read_csv.
    - Write the DataFrame using Streamlit.
  - If the file is a TXT file (file_path.endswith('.txt')):
    - Write "Contents of TXT file:".
    - Attempt to read the TXT file into a DataFrame using pd.read_csv with tab delimiter and UTF-16 encoding.
    - If reading fails due to a parsing error, write an error message.
  - Write a separator line (st.write("---")).

Streamlit App for File Visualization and Analysis:

- Set the title of the Streamlit app to "File Visualization and Analysis".
- Set the title of the input section to "MA Agent Instruction Input".
- Define a list of instructions: ["Annual Plan", "Material Plan", "Policy Evaluation"].
- Create a dropdown menu for selecting an instruction, storing the selection in instruction.
- If the "Submit" button is clicked:
  - Create a dictionary instruction_data with the selected instruction.
  - Write the dictionary to instruction.json as a JSON file.
  - Display a success message indicating that the instruction was saved.

- Create text input fields for entering the paths of the MAP_INPUT and MAP_OUTPUT directories, with default values.
- If the "List and Display Files" button is clicked:

    - **For MAP_INPUT directory**:
        - Set the subtitle to "Files in MAP_INPUT Directory".
        - Call list_files with the MAP_INPUT directory path and store the result in input_files.
        - If input_files is not empty, call display_files with input_files. Otherwise, display a message indicating no files were found.
    - **For MAP_OUTPUT directory**:
        - Set the subtitle to "Files in MAP_OUTPUT Directory".
        - Call list_files with the MAP_OUTPUT directory path and store the result in output_files.
        - If output_files is not empty, call display_files with output_files. Otherwise, display a message indicating no files were found.

5.6 Design of MA Agent

Key responsibility of the MA agent is to work as an interface agent receiving instruction from human users and instruct EA agent for required tasks. Top level algorithm for the MA agent would be as follows and detailed implementation would be discussed in Chapter 6.

1. Load the user interface.
2. Wait for instruction input from the user.
3. Depending on the user input, allocate the task to EA agent.

5.7 Design of EA agent

EA agent was designed to; identify inconsistencies in Meter Database compared to policy requirement in Metering Ontology as issue, load the reported issue in metering system, find the solution for all issues identified, calculate time required for the activities in solutions, identify deadlines for implementing solutions, identify relevant employees for implementation and then to allocate material planning and activity scheduling subtasks to SK agent and ES agent respectively based on the instruction received from MA agent. If the EA agent allocate tasks to the ES agents, if combine all schedules prepared by ES agents. Top level implementation of EA agent was discussed in aforementioned overall implementation of Multi Agent System and the algorithm for EA agent’s skill sets or the functions would be as below. The quarrying of owl ontology had been defined in another python file.

Algorithm for the maintenance_planning() Function

- Initialize DataFrame:

- Create an empty DataFrame maintenance_activity_df with specified columns.

- Read Input Data:
  - Read metering data from Meters.csv into metering_df.
  - Read maintenance request data from Maintenance_Request.csv into maintenance_request_df.
- Iterate Over Each Consumer:
  - For each row in metering_df:
    - Extract consumer_ID, consumer_category, and meter_model.
  - Initialize Lists and Variables:
    - Initialize Solutions_lst, Materials_lst, Issues_lst as empty lists.
    - Initialize Due_days_int and Time_dec
  - Check Meter Model:
    - Find allowed meter models for the consumer category using Onto_SPARQL.MeterModel_finding function.
    - If the consumer's meter model is not allowed:
      - Add "meter_model_issue" to Issues_lst.
      - Find maintenance task details using Onto_SPARQL.Maintenance_Task_finding function for "meter_model_issue".
      - Append solution and material to respective lists.
      - Update Due_days_int and Time_dec.
  - Check Maintenance Requests:
    - Filter maintenance_request_df for requests matching the current consumer_ID.
    - For each matching request:
      - Find maintenance task details using Onto_SPARQL.Maintenance_Task_finding function.
      - Append solution, material, and issue to respective lists.
      - Update Due_days_int and Time_dec.
  - Find Assigned Employee:
    - Query Onto_SPARQL.Assigned_Employee_finding to find the assigned employee for the consumer's area.
  - Calculate Due Date:
    - Add Due_days_int to the current date to determine Due_date
  - Check Routine Testing:
    - If Solutions_lst is empty:
      - Find the test frequency using Onto_SPARQL.TestFrequency_finding function.
      - Find maintenance task details for "meter_testing_request".
      - Append solution, material, and issue to respective lists.
      - Update Due_date based on Last_TestDate and test frequency.
  - Create New Row:
    - Construct a new row dictionary with the consumer's details and append it to maintenance_activity_df.
- Save to File:
  - Define the output file path.
  - Save maintenance_activity_df to Activity_Plan.txt as a tab-delimited UTF-16 encoded file.

Algorithm for the task_allocation() Function

- Read Input Data:
  - Read the activity plan from Activity_Plan.txt into ES_task_list_df.
- Define Folder Path:
  - Set the folder path for output files.
- Allocate Tasks by Employee:
  - For each employee ("ES_01", "ES_02", "ES_03", "ES_04"):
    - Filter ES_task_list_df for tasks assigned to the current employee.
    - Define the output file path for the employee's tasks.
    - Save the filtered DataFrame to a tab-delimited UTF-16 encoded file.

Algorithm for the combin_schedules() Function

- Read Individual Schedules:
  - Read each employee's schedule file (ES_01_schedule.txt, ES_02_schedule.txt, ES_03_schedule.txt, ES_04_schedule.txt) into respective DataFrames.
- Combine Schedules:
  - Concatenate the individual DataFrames into combined_schedule_df.
- Save Combined Schedule:
  - Save combined_schedule_df to combined_schedule.csv.
- Return Combined Schedule:
  - Return the combined_schedule_df.

Required function with SPARQL quarries for the EA agent was developed in a separate python file and the algorithm of those functions would be as follows.

Algorithm for the TestFrequency_finding() Function

- Define SPARQL Query:
  - Construct a SPARQL query to find the test frequency for a given Consumer_Category using the meter_onto:test_frequency property.
- Execute SPARQL Query:
  - Run the SPARQL query against the ontology.
- Extract Test Frequency:
  - Iterate over the query results to retrieve the Test_Frequency.
- Return Test Frequency:
  - Return the extracted test_frequency.

Algorithm for the Maintenance_Task_finding() Function

- Define SPARQL Query:
  - Construct a SPARQL query to find the solution, material, time, and due days for a given issue and Consumer_Category using relevant properties in the ontology.
- Initialize Default Values:
  - Set default values for solution_name, material_name, time, and due_days.
- Execute SPARQL Query:

  - Run the SPARQL query against the ontology.
- Extract Values:
  - Iterate over the query results to retrieve and parse the solution, material, time, and due_days.
- Return Values:
  - Return the extracted solution_name, material_name, time, and due_days.

Algorithm for the Assigned_Employee_finding() Function

- Define SPARQL Query:
  - Construct a SPARQL query to find the employee designation and name for a given Area using the meter_onto:assigned_employee and meter_onto:has_name properties.
- Execute SPARQL Query:
  - Run the SPARQL query against the ontology.
- Extract Employee Details:
  - Iterate over the query results to retrieve and parse the Employee_Designation.
- Return Employee Details:
  - Return the assigned_employee details.

Algorithm for the MeterModel_finding()Function

- Define SPARQL Query:
  - Construct a SPARQL query to find allowed meter models for a given Consumer_Category using the meter_onto:has_meterModel property.
- Initialize List:
  - Initialize an empty list Meter_Models.
- Execute SPARQL Query:
  - Run the SPARQL query against the ontology.
- Extract Meter Models:
  - Iterate over the query results to retrieve and parse the meter_model.
  - Append each meter_model to Meter_Models.
- Return Meter Models:
  - Return the list of Meter_Models.

5.8 Design of ES agent

The basic function of the ES agent is to schedule the given maintenance activities based on the priority requirement given by due days, average monthly consumption, time required for the given activities, traveling time and the working days of the assigned employee. Algorithm for the function of the ES agent is given below.

1. Wait to receive an Activity Plan from EA agent.
2. Order the Activity Schedule according the deadlines for acting and then according to average electricity consumption of the customer.
3. Decide planning date for the Activities based on his personal working calendar and time required for activities.
4. Prepare the Activity Schedule.

The algorithms for the functions of ES agent is given below.

Algorithm for the avtivity_scheduler(ES_designation) Function

- Load Data from CSV Files:
    - Load the working days data, base locations data, and task list data for the specified ES.
- Preprocess Data:
    - Convert date strings in the working days DataFrame to date objects.
    - Sort the task list DataFrame by due date and average consumption.
- Initialize Schedule DataFrame:
    - Create an empty DataFrame for the schedule with specified columns.
- Set Initial Variables:
    - Set the initial base location to the office location.
    - Initialize variables for working time, traveling distance, and traveling time.
- Main Scheduling Loop:
    - While the task list is not empty:
        - Get the Earliest Working Day:
            - Filter the calendar to get only working days.
            - Select the earliest working day and remove it from the calendar.
        - Select the First Consumer:
            - Select the first consumer from the task list.
            - Set the starting location to the base location.
            - Get the coordinates of the selected consumer.
            - Calculate the driving distance and time to the consumer.
        - Update Working Time:
            - Add the traveling time and the time required for maintenance tasks to the working time.
        - Extract Task Details:
            - Extract details such as province, area, assigned employee, consumer category, average consumption, and maintenance activities for the selected consumer.
        - Add to Schedule:
            - Append the task details to the schedule DataFrame.
            - Remove the selected consumer from the task list.
            - Update the base location to the consumer's location.
        - Secondary Loop for Same Area Tasks:
            - While the working time is less than 12 hours:
                - Check for Tasks in the Same Area:
                    - If there are tasks in the same area, select the next consumer.
                    - Calculate the driving distance and time to the consumer.
                    - Update the working time with traveling times
                    - If the working time is less than 13 hours, add the task to the schedule and update the base location.

            - If not, revert to the previous consumer.
        - Break if No More Tasks in the Same Area:
            - If there are no tasks in the same area or if the working time exceeds 12 hours, break the loop.
    - Update for Next Day:
        - Increment the planning date.
        - Check if the next day is a working day and set the destination accordingly (either to the area accommodation or back to the office).
        - Calculate the driving distance and time to the end-of-day destination.
        - Update the last task's traveling distance and total time with the end-of-day travel details.
    - Reset Variables for Next Iteration:
        - Reset the traveling distance and time for the next iteration.
- Save and Return Schedule:
    - Save the schedule DataFrame to a CSV file.
    - Return the schedule DataFrame.

Supportive Functions Used in the Main Function of EA agent's skills

- get_driving_distance_and_time:
    - Calculate the driving distance and time between two GPS coordinates using the OSRM API.
    - Convert the response to distance in kilometers and duration in hours.
- get_gps_coordinates:
    - Retrieve GPS coordinates for a given location from a DataFrame.
    - Return default coordinates if the location is not found.

5.9 Design of SK agent

SK agent is responsible for extracting material requirements for an Activity Plan, deciding order quantities and prepare material plan. His top level design algorithm would be.

1. Wait to receive an Activity Plan from EA agent.
2. Extract material requirement from received Activity Plan.
3. Read the stock levels, reorder levels and minimum order quantities.
4. Prepare a material plan including stock level, material requirement and order quantity.

The algorithms for the functions of SK agent is given below.

Algorithm for material_demand() Function

- Load Activity Plan Data:
    - Read the activity plan data from a specified CSV file into a DataFrame.

- Explode Materials List:
  - Extract the 'Materials' column and convert the list of materials into separate rows using explode.
- Count Material Occurrences:
  - Count the occurrences of each material in the exploded DataFrame.
- Rename Columns:
  - Rename the columns to 'Materials' and 'Material_Demand'.
- Return:
  - Return the DataFrame containing material demand counts.

Algorithm for material_plan() Function

- Get Material Demand:
  - Call the material_demand function to get the material demand DataFrame.
- Load Stock Data:
  - Read the stock keeping data from a specified CSV file into a stock_df.
- Merge DataFrames:
  - Merge the stock_df and material_demand_df on the 'Materials' column.
- Calculate Order Quantity:
  - Use the following conditions to calculate the 'Order_Qty':
    - If the current stock level minus material demand is greater than the minimum stock level, set 'Order_Qty' to 0.
    - If the current stock level minus material demand plus the minimum stock level is less than 0, set 'Order_Qty' to the minimum order quantity minus current stock level plus material demand.
    - If the current stock level minus material demand is less than the minimum order quantity, set 'Order_Qty' to the minimum order quantity.
    - Otherwise, set 'Order_Qty' to 0.
- Create Material Plan data frame:
  - Create material_plan_df with columns for 'Materials', 'Stock_Level', 'Material_Demand', and 'Order_Qty'.
- Save and Return Material Plan:
  - Save the material_plan_df to a specified CSV file.
  - Return the material_plan_df.

5.10 Summary

This chapter detailed the design of the Ontology driven Multi Agent based Electricity Meter Management system. The design is basically divided into two parts: domain knowledge modeling and problem solving knowledge modeloing. Domain knowledge was modeled as an OWL ontology. Problem solving was implemented as a Multi Agent Planning system with MA agent, EA agent, ES agents and SK agent. Chapter 6 in the next section will explain the detail implementation of the design discussed in this chapter.

CHAPTER 6

IMPLEMENTATION OF MULTI AGENT ELECTRICITY METER MANGEMENT SYSTEM

6.1 Introduction

The Design chapter which was the previous chapter detail explained the experimental design including overall design, ontology design and the design of the agents. In this chapter, it is discussed the implementation of the design in a python based environment using SPADE platform.

6.2 implementation of knowledge base as an ontology

Ontology knowledge base has conceptualized the concepts related to electricity meter management into class and those have been given below. The class are: “ACCURACY_CLASSES”, “AREAS”, “CONSUMER_CATEGORIES”, “DISCO_STAFF”, “ISSUES”, “MAINTENANCE_ACTIVITIES”, “MATERIAL”, “METER_MODELS”, “TEST_FREQUNCIES”.

These classes were created as classes in Protégé as shown in Figure 6.1.

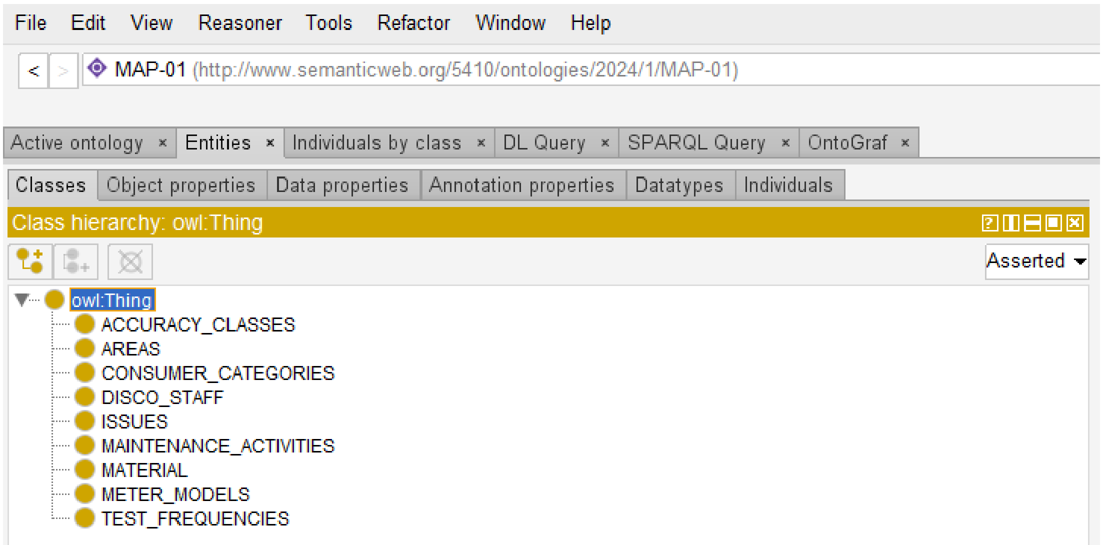

Figure 6.1: Classes in the OWL ontology tool Protégé.

For each conceptual class the individuals have been added as instances in protégé. As an Example of the class named “AREAS” and its individuals are shown in Figure 6.2. All the instances or individuals are also can be seen at once in the “Individuals” tab under “Entities” tab which is shown in Figure 6.3.

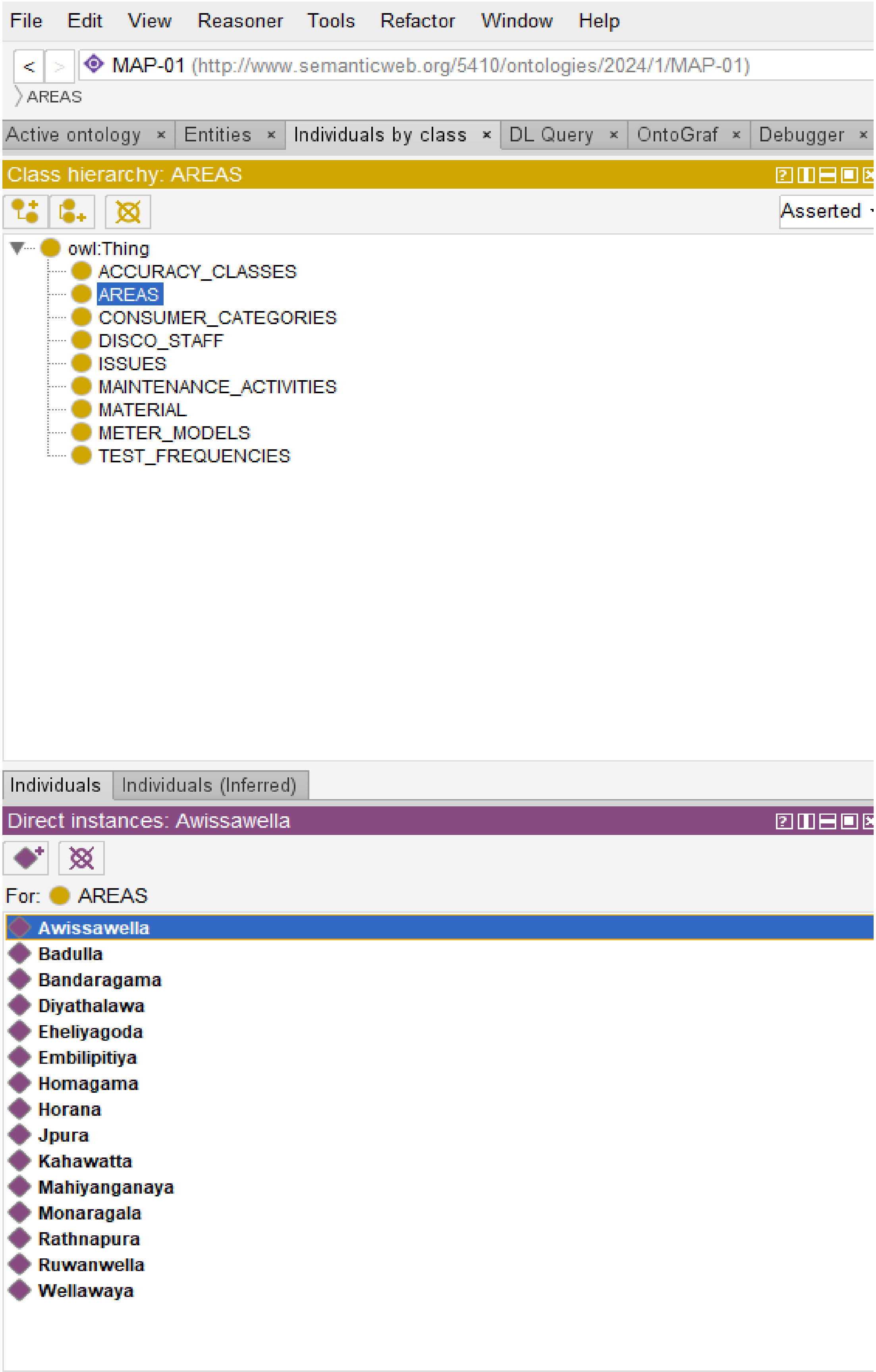

Figure 6.2: Class named “AREAS” and its instances in Protégé.

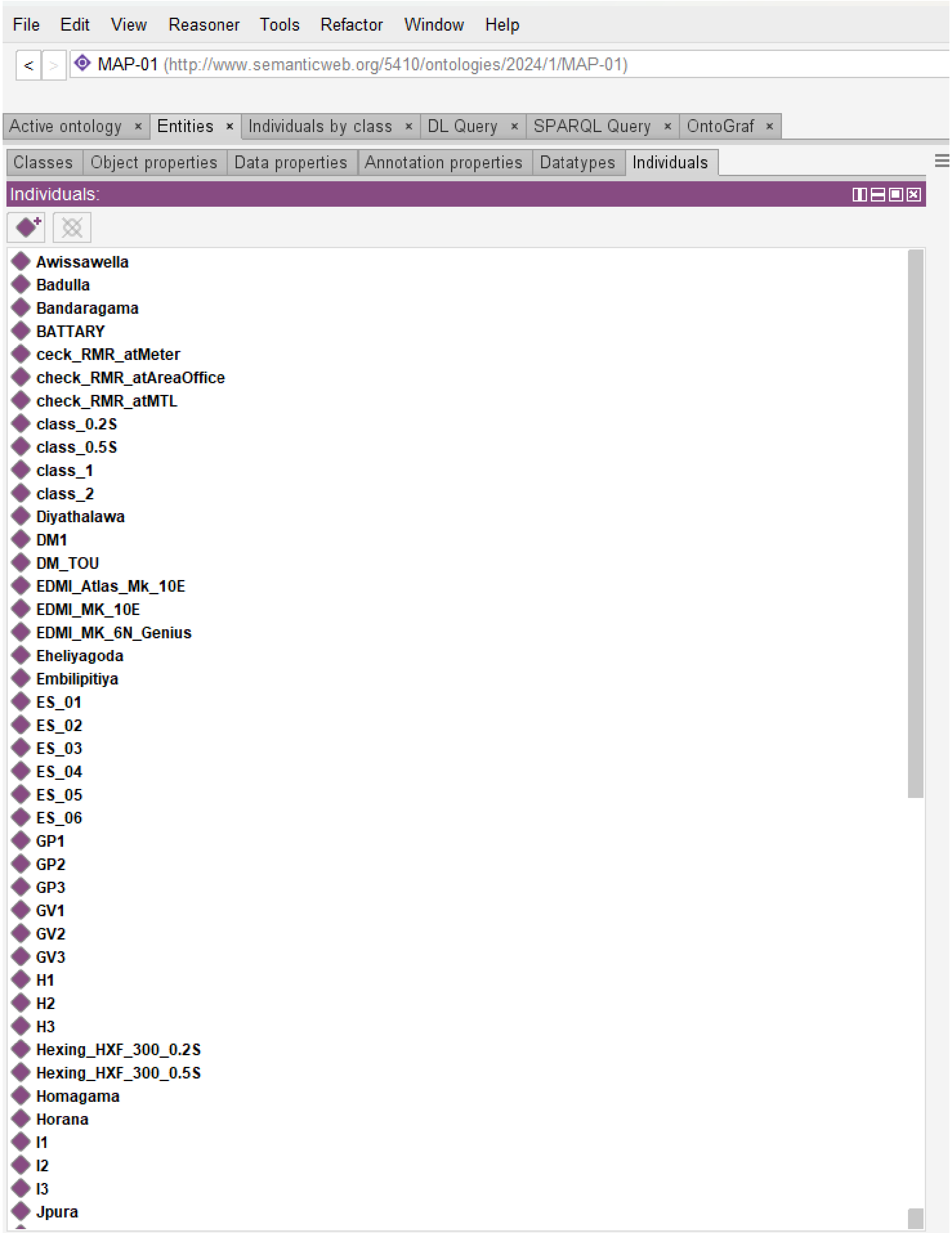

Figure 6.3: All the individual view in Protégé.

Relationships between individuals of different classes have been defined through object property assertion. Used object properties can be seen in the "Object properties" tab under "Entities" tab in Protégé which is given in Figure 6.4. The relevant data values for the instances/ individuals have been asserted through data property assertion and the used data properties are given Figure 6.5 as can be visualized on the "Data properties" tab under the "Entities" tab in Protégé.

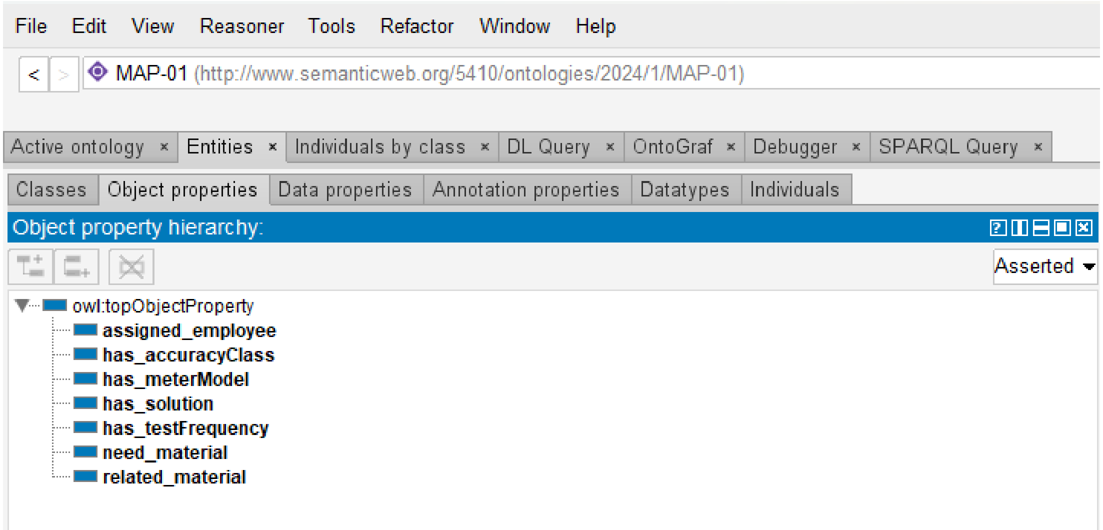

Figure 6.4: Hierarchy of object properties in Protégé.

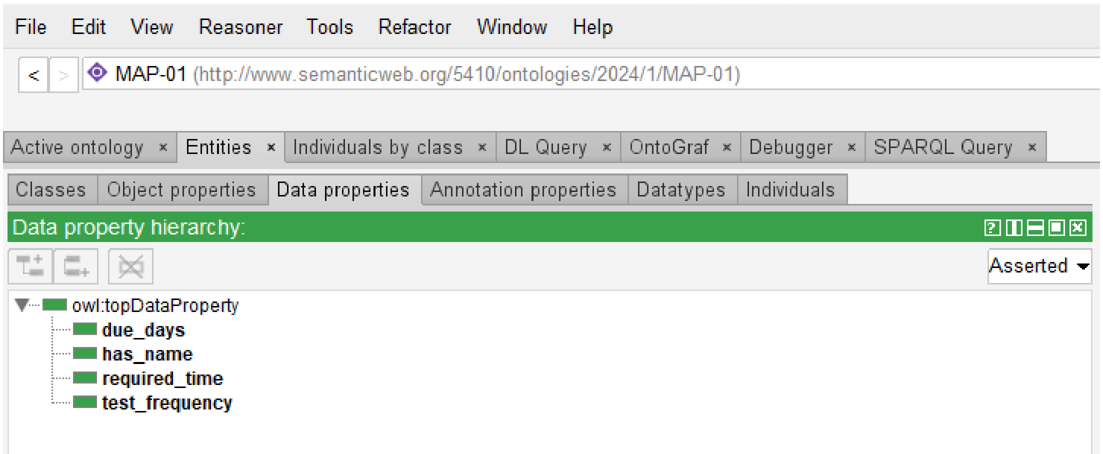

Figure 6.5: Hierarchy of data properties in Protégé.

Relationships among individuals can be visualized from the “OntoGraf” window of protégé which is shown in Figure 6.6.

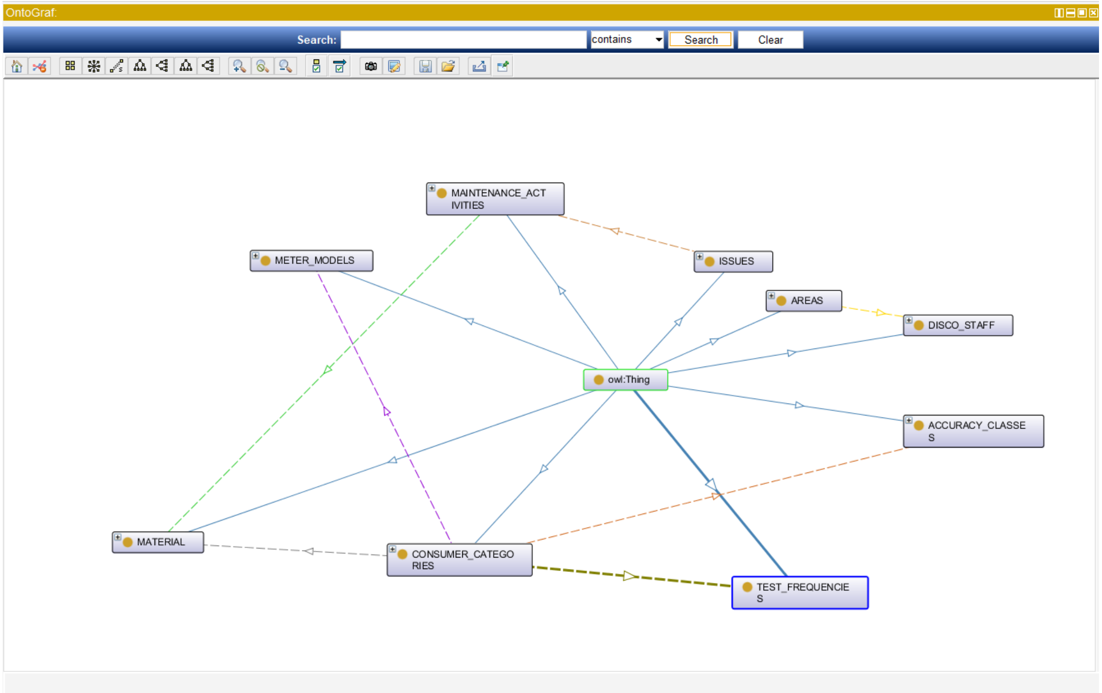

Figure 6.6: Relationship among classes in Protégé.

6.3 Implementation of MA agent

MA agent is responsible for receiving user inputs and allocating tasks for the EA agent based on that. MA agent is being implemented in Microsoft VSCode using python. Implementation of MA agent is embedded into the master file of implementing the overall Multi Agent System using SPADE module. Implementation of MA agent in the master file for MAS base Electricity Meter Management is shown in Figure 6.7.

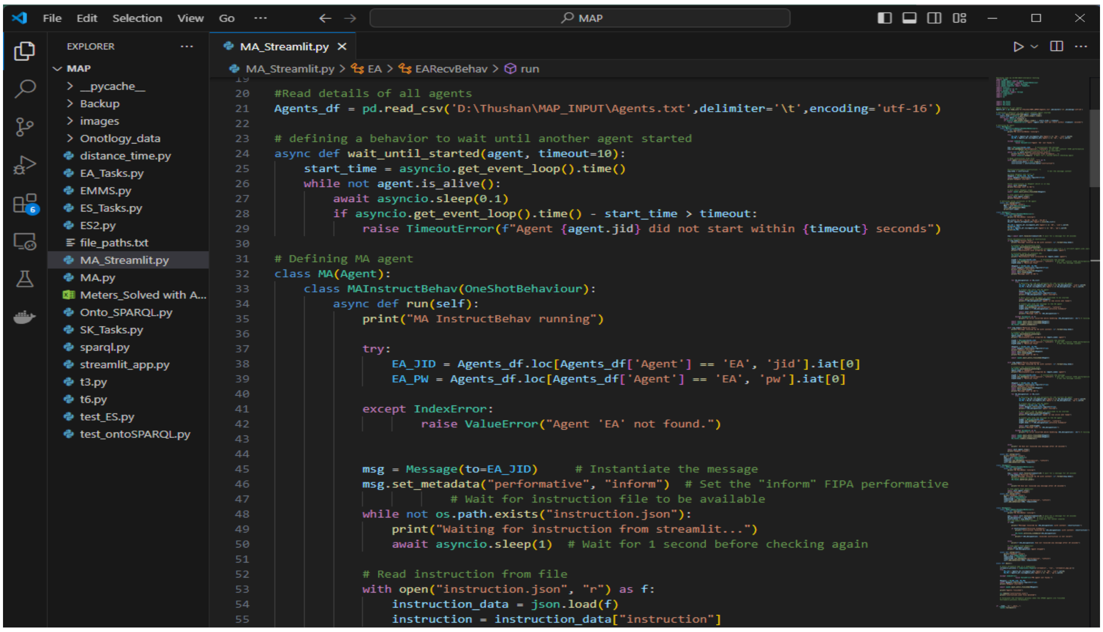

Figure 6.7: Implementation of MA agent and overall Multi Agent System.

6.4 Implementation of EA Agent

Implementation of skills/functions of EA agent was done on Microsoft VSCode using python language. Codes for the EA agent's function were developed into two python files. The first file contains functions for reasoning part of the agent that queries the Metering Ontology using SPARQL. The second file contains agents' sensing and acting functions. The inputs or the sense parameters for this EA agent include OWL Metering Ontology file, Metering database as a CSV file and the records of Meter Issues as another CSV file.

In the first python file of EA agent, it was defined the functions for finding assigned employees, function for finding allowed meter models, function for finding test frequencies and functions to find relevant maintenance tasks for each issue. In each of those functions Metering Ontology was queried using SPARQL. In order to implement SPARQL queries in python, "rdflib" module was used. Implementation of this python file for the reasoning part is shown in Figure 6.8.

The second python file of EA agent contains functions for loading metering database, maintenance activity planning, allocation of Activity Plans to EA agents for preparing Activity Schedules and merging produce Activity Schedules by ES agents. Implementation of these functions are depicted in Figure 6.9.

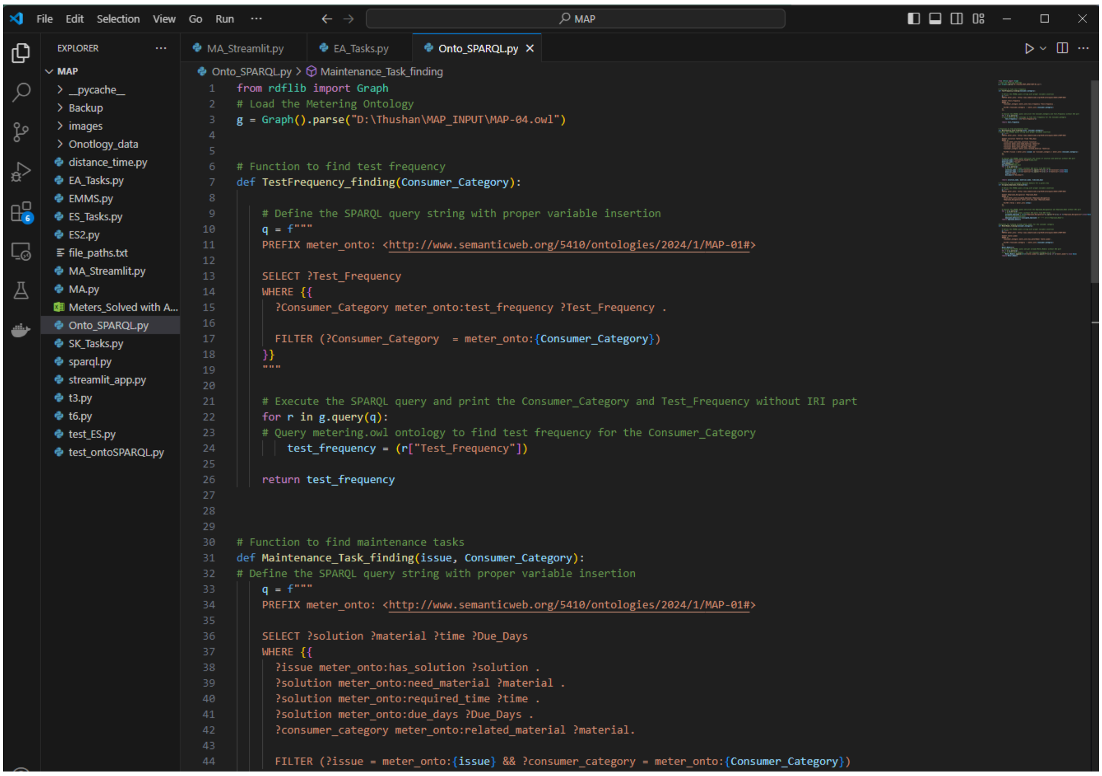

Figure 6.8: Implementation of EA agent's reasoning functions that query Metering ontology with SPARQL.

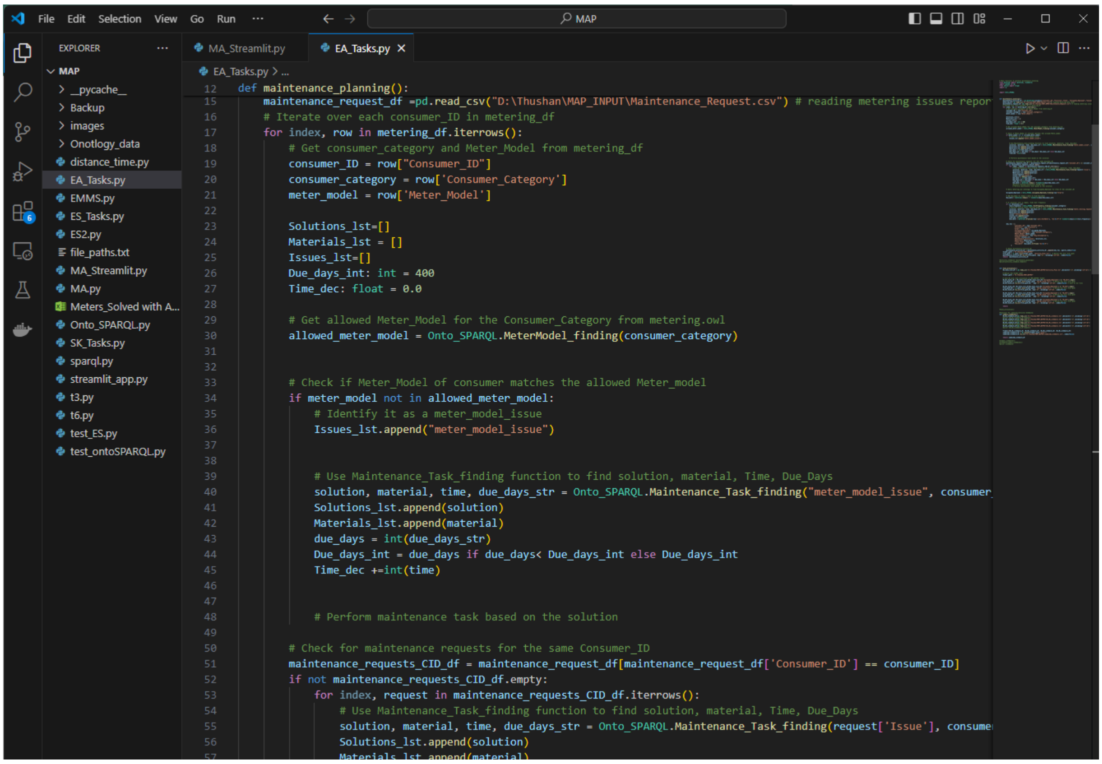

Figure 6.9: Implementation of EA agent's sensing and acting functions.

6.5 Implementation of ES agents

ES agents are responsible for preparing Activity Schedules using the Activity Plan given by EA agent. EA agent senses the environment by loading Activity Plan as txt file, working calendar as CSV file and locations of consumer premises and base locations as a CSV file. After sensing necessary parameters ES agents use activity scheduler function to create an Activity Schedule. Activity scheduler function first orders the activity plan based on due date and then based on average consumption of the consumer. Then the activities are scheduled into an Activity Schedule based on the time required for activities and his working calendar interacting with OSRM server. Implementation of ES agent's functions are also done on Microsoft VSCode using python programing language as shown in Figure 6.10.

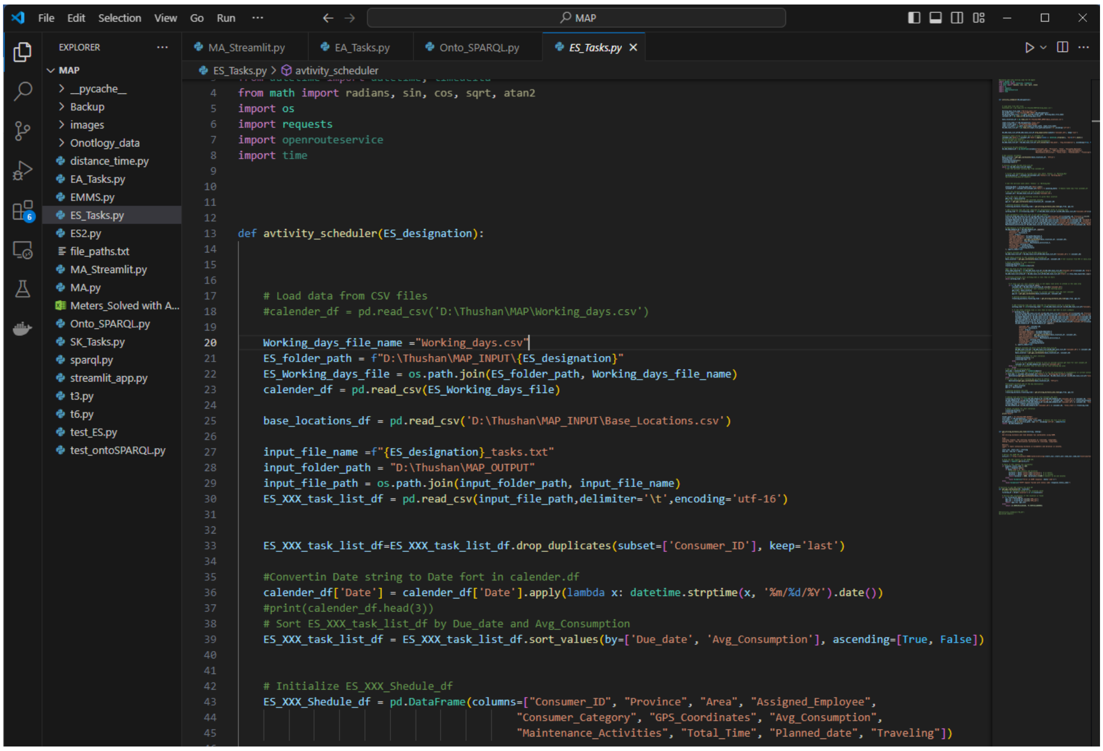

Figure 6.10: Implementation of ES agents sensing and acting functions.

6.6 Implementation of SK agent

SK agent's responsibility is to create a Material Plan using the Activity Plan given by the EA agent. In preparation of Material Plan, SK agent extract material requirements for the given Activity Plan, load the stock level with minimum stock level and minimum order quantity. SK agent provide a material plan with stock level, material requirement for planned activities and material order requirement. SK agents is also implemented on Microsoft VSCode using python which is shown in Figure 6.11.

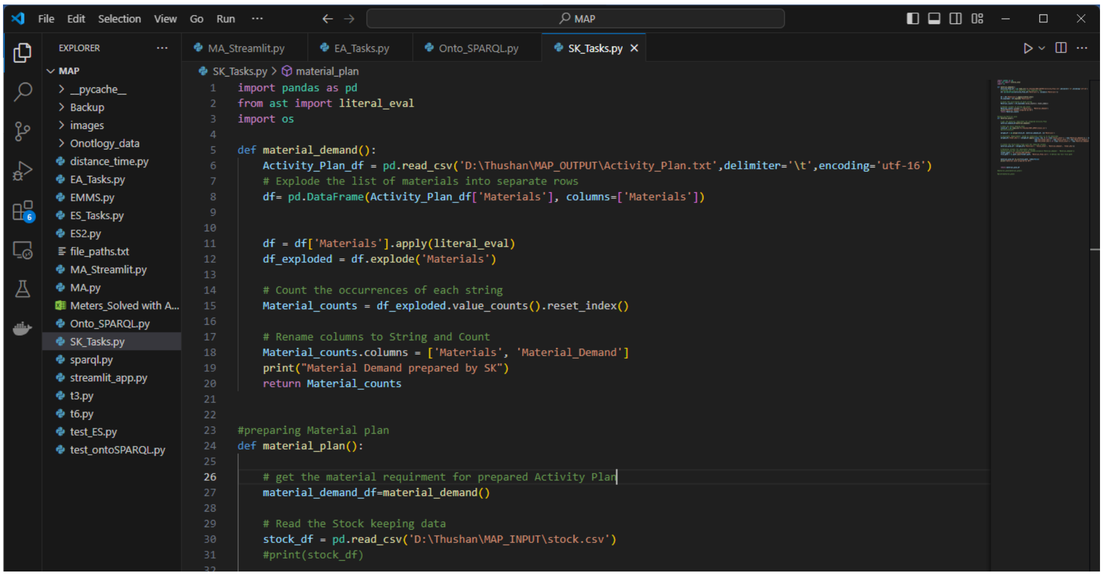

Figure 6.11: Implementation of SK agents sensing and acting functions.

6.7 Summary

Overall Implementation of the experimental design was done on Microsoft VSCode using python language. In addition to that, Protégé ontology editor was used to prepare the domain knowledgebase (metering ontology) that is utilized in the MAS based metering management system. In the next chapter, that is Chapter 7, it will discuss the evaluation of the developed Electricity Meter Management System.

CHAPTER 7

EVALUATION OF MULTI AGENT ELECTRICITY METER MANGEMENT SYSTEM

7.1 Introduction

Previous chapter explained the implementation of the multi agent electricity meter management system in VSCode. The developed system was tested using the data from one of the CEB distribution divisions smart meter data. Results of the evaluation are discussed in detail in this chapter.

7.2 Experimental Design

The overall design of the solution is divided into three parts: ontology and SPARQL query, Agent based system and overall system. Ontology and SPARQL query parts, the Agent wise functions and the overall system were tested. The details of testing are given below under separate headings.

7.3 Testing of Ontology

First step of developing the meter management system was to develop the domain knowledge base as an ontology using Protégé tool. Therefore, ontology was tested before everything else was tested. First step of testing ontology was to check for consistency. Test for the consistency was done using Protégé debugger as shown in Figure 7.1. The debugging process results in the build ontology consistent and coherent. In the developed metering management system ontology was used with SPARQL queries to find the suitable maintenance activities, material, due days, materials and relevant employees responsible for those activities. The ability to query the knowledge base was tested on the Protégé tool itself using SPARQL queries.

Further, in order to verify the ability of ontology together with SPARQL queries Following tests were done. The results of the below mentioned testing have proved that the ontology together with SPARQL queries can provide the required knowledge for the required activities in the overall system.

Testing 7.2: Testing the ability to obtain allowed meter models was done for a given consumer category and it was possible to obtain the correct allowed meter models (Figure 7.2).

Testing 7.3: It was tested to obtain the test frequency specified for a consumer category with SPARQL queries and it provided the correct test frequency (Figure 7.3).

Testing 7.4: Once the Area of the consumer is given, the ability to give the assigned employee’s designation and the name was checked. the results were satisfactory and it is shown in Figure 7.4.

Testing 7.5: If there is any issue that had been identified with a particular consumer category, ability to get the solution, required materials, time required and due days were checked using SPARQL queries. Query results were satisfactory and it is shown in Figure 7.5.

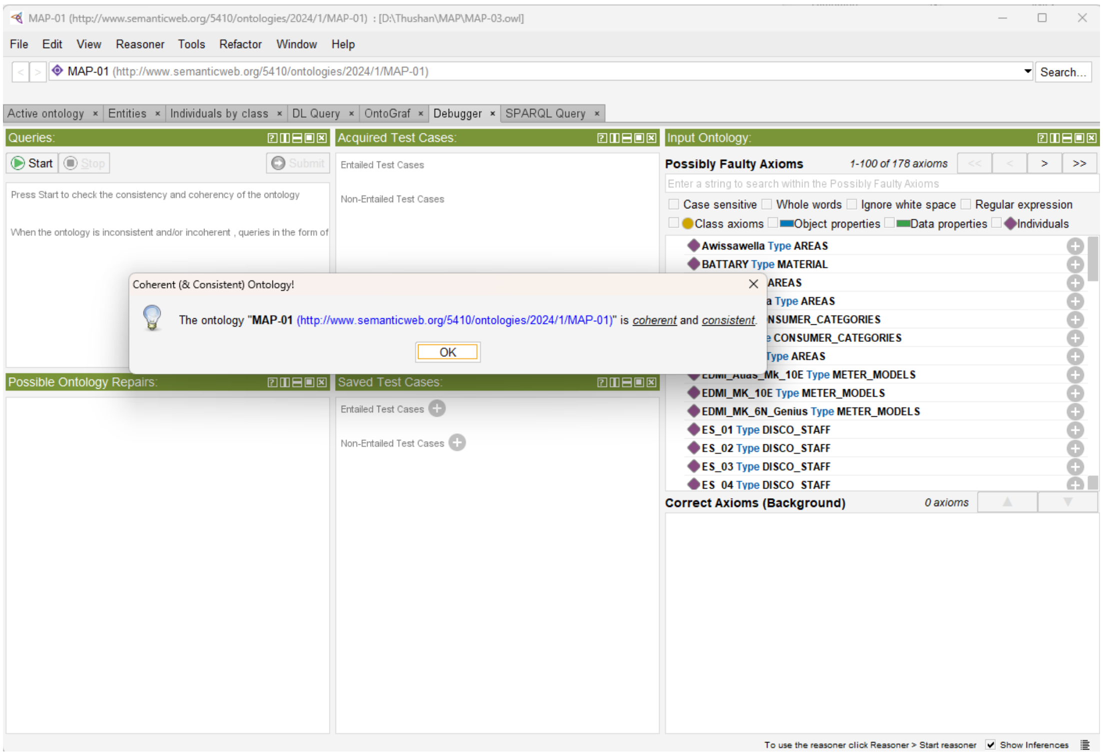

Figure7.1: Checking consistency of ontology using Protégé debugger.

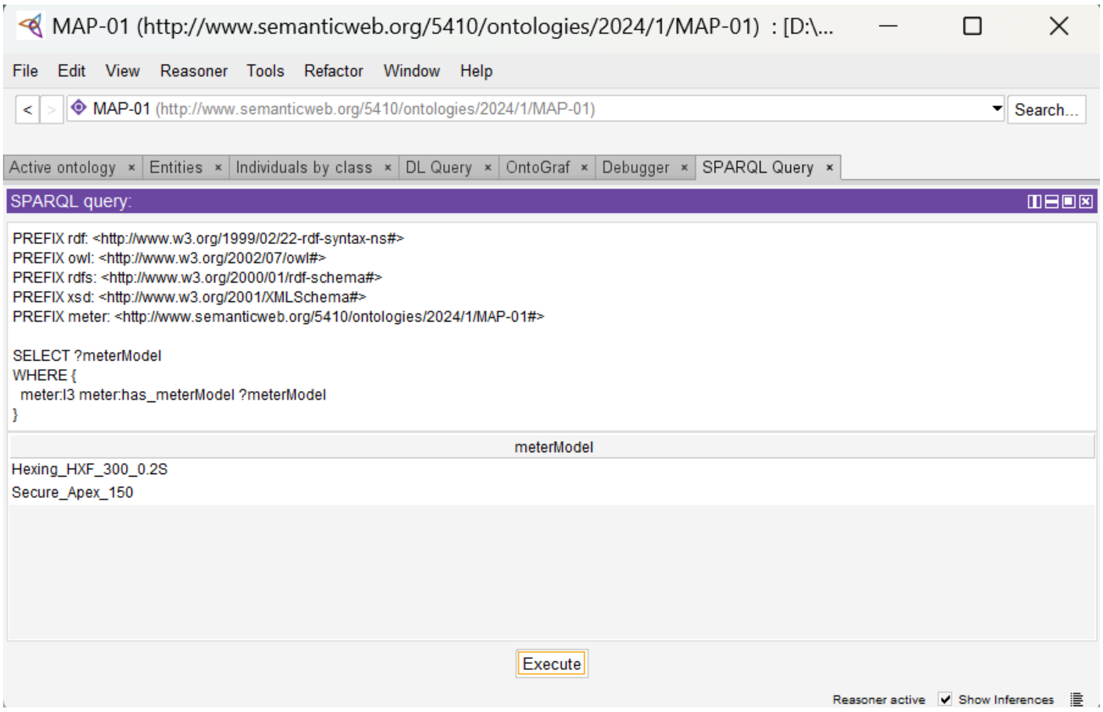

Figure7.2: Testing the ability to get allowed meter models for a given consumer category using SPARQL query in Protégé.

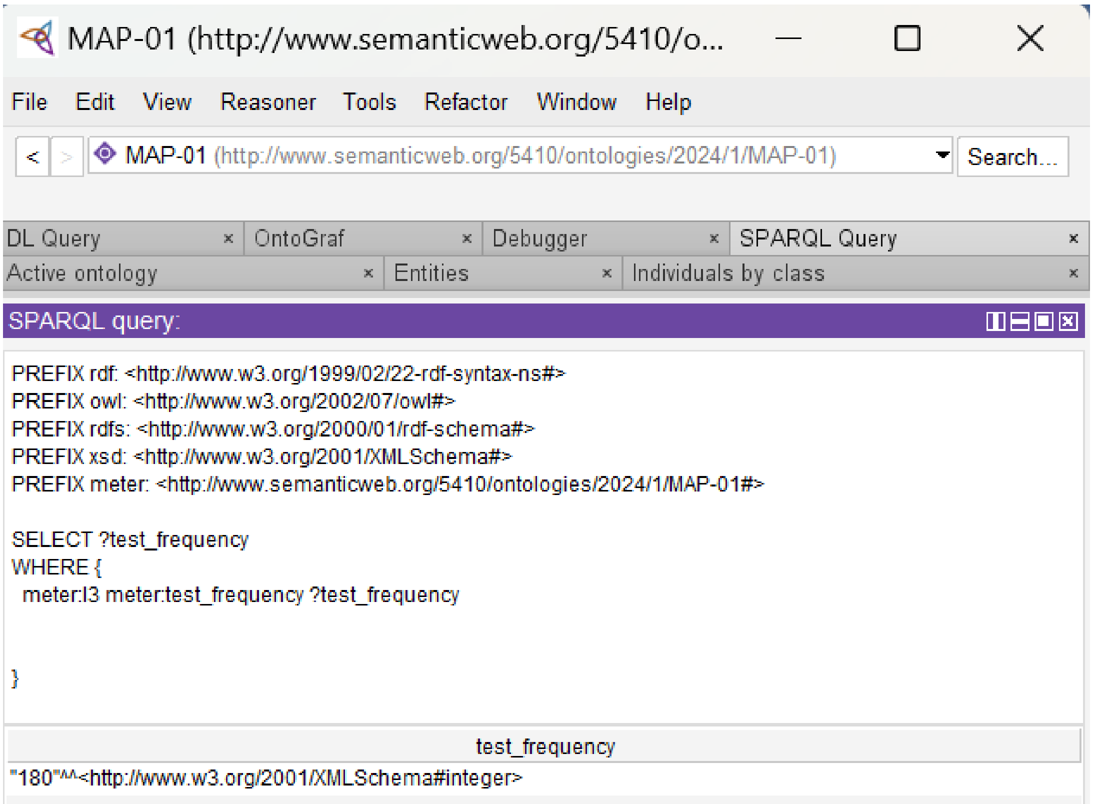

Figure 7.3: Testing whether the test frequency can be obtained for a given consumer category using SPARQL queries in Protégé.

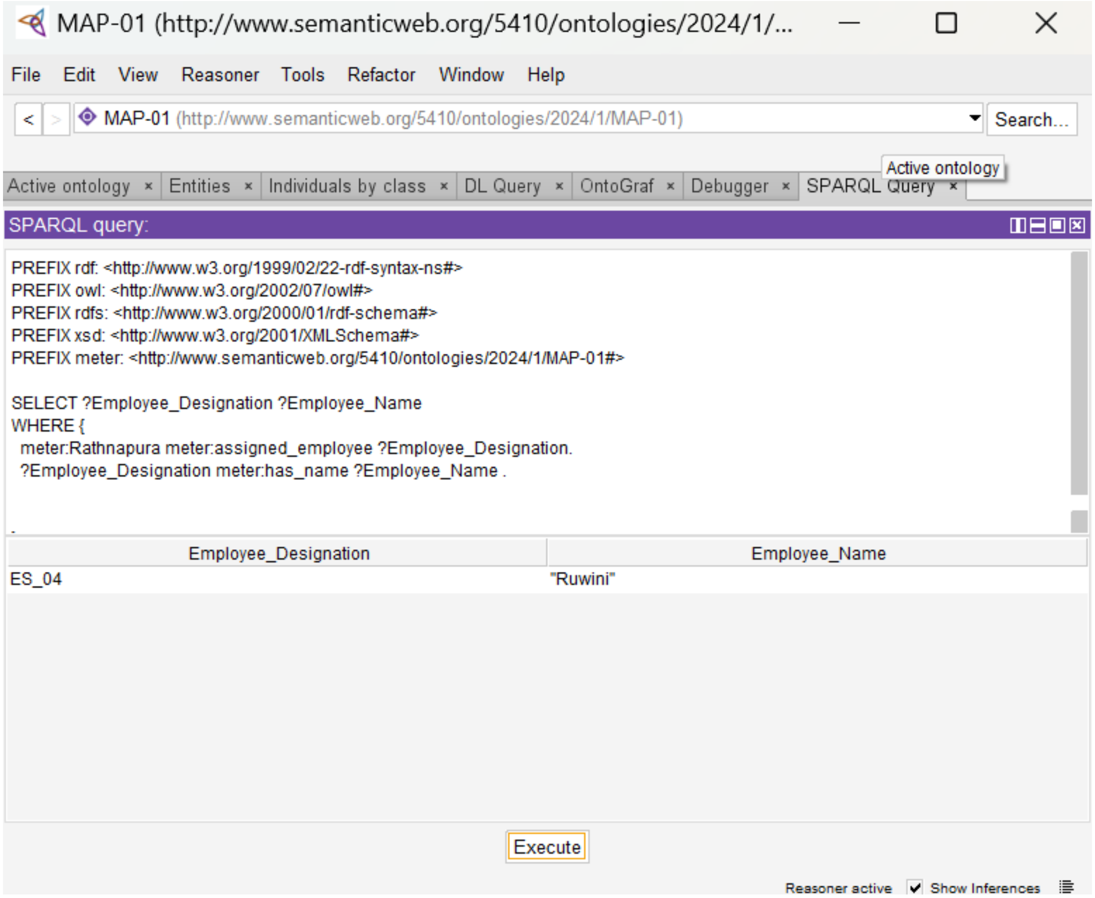

Figure 7.4: Testing to get an assigned employee and his name for a given Area using SPARQL queries in Protégé.

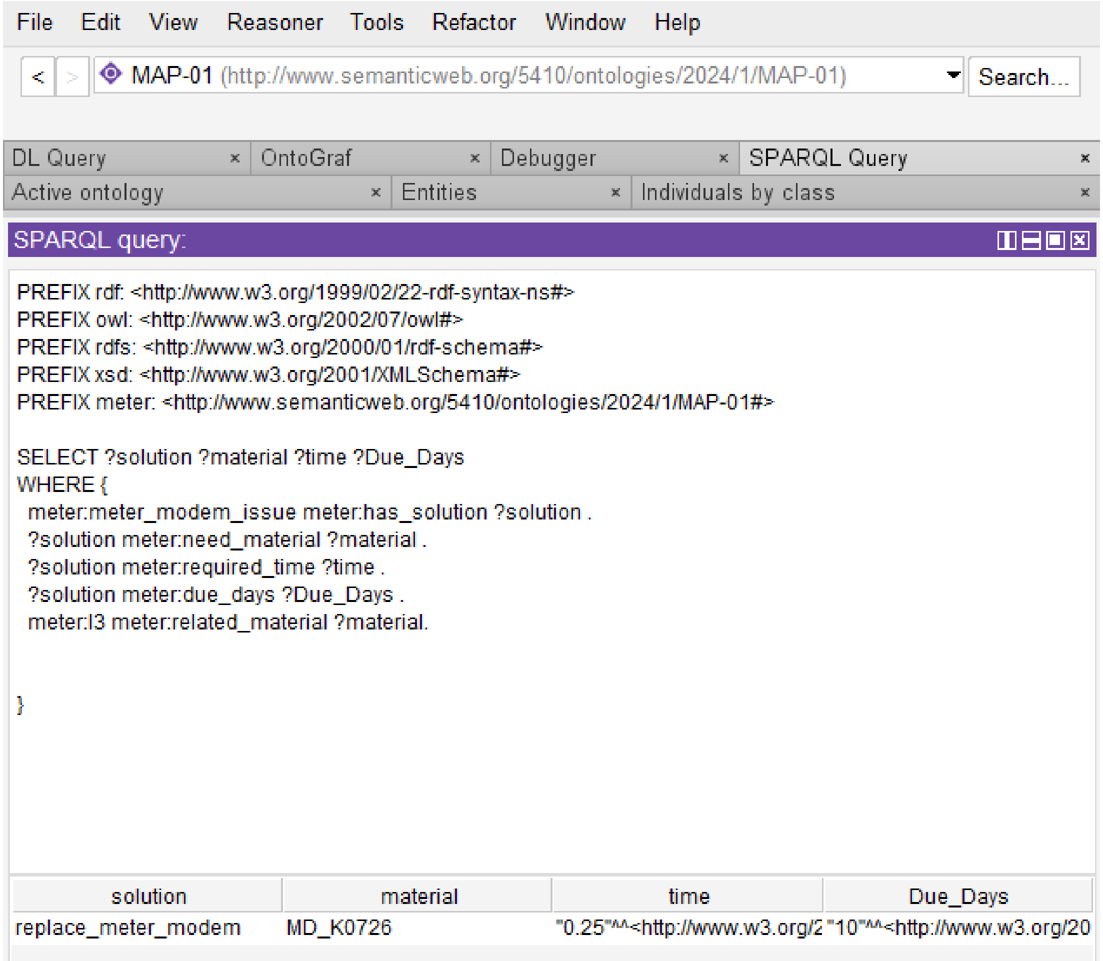

Figure 7.5: Testing whether it is possible to get a solution, required materials, time required and due days once the issue and consumer category is given.

7.4 Testing of EA agent

For the purpose of evaluating the implementation of EA agent, a set of metering data and Maintenance requests relevant to those meters were given in CSV format to the EA agent. These data were taken relevant to a set of smart meters in Sri Lanka. Snapshots of given meter data and maintenance requests are shown in Figure 7.6 and Figure 7.7 respectively. After running the functions of EA agent for preparing Activity Plans from the given metering data and the metering ontology, it could produce an acceptable plan for the maintenance activities. Output of the EA agent is shown in Figure 7.8

Consumer_ID	Province	Area	CSC	Meter_Model	Accuracy_	Consumer	Avg_Cons	Last_TestDate	GPS_Lon	GPS_Lat
UBadBad7010011	UVA	Badulla	Badulla	EDMI_Mk_10E	class_1	GP2	20534	6/9/2023	6.983355	81.05859
UBadBad7010043	UVA	Badulla	Badulla	EDMI_Mk_10E	class_1	I2	32013	6/9/2023	6.984684	81.0586
UBadBad7010045	UVA	Badulla	Badulla	EDMI_Mk_10E	class_1	I2	21154	4/18/2023	6.984789	81.05903
UBadBad7010065	UVA	Badulla	Badulla	EDMI_Mk_10E	class_1	GP2	5090	4/18/2023	6.985839	81.06007
UBadBad7010066	UVA	Badulla	Badulla	EDMI_Mk_10E	class_1	GV2	122514	4/18/2023	6.986462	81.0566
UBadBad7010067	UVA	Badulla	Badulla	EDMI_Mk_10E	class_1	I2	23491	6/9/2023	6.98665	81.05709
UBadBad7010077	UVA	Badulla	Badulla	EDMI_Mk_10E	class_1	I2	53491	6/9/2023	6.986094	81.05741
UBadBad7010097	UVA	Badulla	Badulla	EDMI_Mk_10E	class_1	I2	4277	7/12/2023	6.987764	81.05861
UBadBad7010100	UVA	Badulla	Badulla	EDMI_Mk_10E	class_1	I2	22297	4/10/2023	6.988052	81.06054
UBadBad7010109	UVA	Badulla	Badulla	EDMI_Mk_10E	class_1	GP2	6474	4/19/2023	6.987253	81.06144
UBadBad7010119	UVA	Badulla	Badulla	EDMI_Mk_10E	class_1	I2	24840	10/18/2023	6.987263	81.06218
UBadBad7010126	UVA	Badulla	Badulla	EDMI_Mk_10E	class_1	GP2	3585	10/18/2023	6.991101	81.05861
UBadBad7010127	UVA	Badulla	Badulla	EDMI_Mk_10E	class_1	GP2	1508	7/12/2023	6.989803	81.06129
UBadBad7010129	UVA	Badulla	Badulla	EDMI_Mk_10E	class_1	GP2	14804	4/26/2023	6.990751	81.06146
UBadBad7010131	UVA	Badulla	Badulla	EDMI_Mk_10E	class_1	GP2	1435	10/18/2023	6.991289	81.06728
UBadBad7010135	UVA	Badulla	Badulla	EDMI_Mk_10E	class_1	GP2	1562	5/10/2023	6.994873	81.06305
UBadBad7010141	UVA	Badulla	Badulla	EDMI_Mk_10E	class_1	GV2	17634	4/26/2023	6.999843	81.0605
UBadBad7010148	UVA	Badulla	Badulla	EDMI_Mk_10E	class_1	GV2	13765	7/12/2023	6.987628	81.05659
UBadBad7010152	UVA	Badulla	Badulla	EDMI_Mk_10E	class_1	GV2	12672	7/12/2023	6.988975	81.05657
UBadBad7010160	UVA	Badulla	Badulla	EDMI_Mk_10E	class_1	GV2	2902	4/16/2023	6.989166	81.05772
UBadBad7010161	UVA	Badulla	Badulla	EDMI_Mk_10E	class_1	GV2	39329	6/8/2023	6.990137	81.0561

Figure 7.6: Snapshot of metering data given to EA agent as an input or sensed parameters.

Consumer_ID	Issue
UBadBad70100113	meter_model_issue
UBadBad70100660	meter_burnt_issue
UBadBad70101004	meter_testing_request
UBadBad70101276	meter_communication_issue
UBadBad70101411	meter_SIM_issue
UBadBad70101616	meter_model_issue
UBadBad70101748	meter_burnt_issue
UBadBad70101896	meter_testing_request
UBadBad70102043	meter_communication_issue

Figure 7.7: Snapshot of Maintenance request given to EA agent as an input or sensed parameters.

	Consumer_ID	Province	Area	Assigned_	Consumer	Meter_Model	Avg_Cons	Issues	Maintenance_Activities	Materials	Total_Tim	Due_date
2	UBadBad701001	UVA	Badulla	ES_04	GP2	EDMI_Mk_10E	20534	['meter_model_issue', 'i	['replace_meter', 'replace_met	['K0721', 'K072	2	7/1/2024
3	UBadBad701004	UVA	Badulla	ES_04	I2	EDMI_Mk_10E	32013	['meter_model_issue']	['replace_meter']	['K0721']	1	7/1/2024
4	UBadBad701004	UVA	Badulla	ES_04	I2	EDMI_Mk_10E	21154	['meter_model_issue']	['replace_meter']	['K0721']	1	7/1/2024
5	UBadBad701006	UVA	Badulla	ES_04	GP2	EDMI_Mk_10E	5090	['meter_model_issue']	['replace_meter']	['K0721']	1	7/1/2024
6	UBadBad701006	UVA	Badulla	ES_04	GV2	EDMI_Mk_10E	122514	['meter_model_issue', 'i	['replace_meter', 'replace_met	['K0721', 'K072	2	7/1/2024
7	UBadBad701006	UVA	Badulla	ES_04	I2	EDMI_Mk_10E	23491	['meter_model_issue']	['replace_meter']	['K0721']	1	7/1/2024
8	UBadBad701007	UVA	Badulla	ES_04	I2	EDMI_Mk_10E	53491	['meter_model_issue']	['replace_meter']	['K0721']	1	7/1/2024
9	UBadBad701009	UVA	Badulla	ES_04	I2	EDMI_Mk_10E	4277	['meter_model_issue']	['replace_meter']	['K0721']	1	7/1/2024
10	UBadBad701010	UVA	Badulla	ES_04	I2	EDMI_Mk_10E	22297	['meter_model_issue', 'i	['replace_meter', 'test_theMet	['K0721', 'no_i	2	7/1/2024
11	UBadBad701010	UVA	Badulla	ES_04	GP2	EDMI_Mk_10E	6474	['meter_model_issue']	['replace_meter']	['K0721']	1	7/1/2024
12	UBadBad701011	UVA	Badulla	ES_04	I2	EDMI_Mk_10E	24840	['meter_model_issue']	['replace_meter']	['K0721']	1	7/1/2024
13	UBadBad701012	UVA	Badulla	ES_04	GP2	EDMI_Mk_10E	3585	['meter_model_issue']	['replace_meter']	['K0721']	1	7/1/2024
14	UBadBad701012	UVA	Badulla	ES_04	GP2	EDMI_Mk_10E	1508	['meter_model_issue', 'i	['replace_meter', 'replace_met	['K0721', 'K072	2	7/1/2024
15	UBadBad701012	UVA	Badulla	ES_04	GP2	EDMI_Mk_10E	14804	['meter_model_issue']	['replace_meter']	['K0721']	1	7/1/2024
16	UBadBad701013	UVA	Badulla	ES_04	GP2	EDMI_Mk_10E	1435	['meter_model_issue']	['replace_meter']	['K0721']	1	7/1/2024
17	UBadBad701013	UVA	Badulla	ES_04	GP2	EDMI_Mk_10E	1562	['meter_model_issue']	['replace_meter']	['K0721']	1	7/1/2024
18	UBadBad701014	UVA	Badulla	ES_04	GV2	EDMI_Mk_10E	17634	['meter_model_issue', 'i	['replace_meter', 'replace_SIM	['K0721', 'SIM'	1.25	7/1/2024
19	UBadBad701014	UVA	Badulla	ES_04	GV2	EDMI_Mk_10E	13765	['meter_model_issue']	['replace_meter']	['K0721']	1	7/1/2024
20	UBadBad701015	UVA	Badulla	ES_04	GV2	EDMI_Mk_10E	12672	['meter_model_issue']	['replace_meter']	['K0721']	1	7/1/2024
21	UBadBad701016	UVA	Badulla	ES_04	GV2	EDMI_Mk_10E	2902	['meter_model_issue']	['replace_meter']	['K0721']	1	7/1/2024
22	UBadBad701016	UVA	Badulla	ES_04	GV2	EDMI_Mk_10E	39329	['meter_model_issue', 'i	['replace_meter', 'replace_met	['K0721', 'K072	2	7/1/2024
23	UBadBad701016	UVA	Badulla	ES_04	GP2	EDMI_Mk_10E	4764	['meter_model_issue']	['replace_meter']	['K0721']	1	7/1/2024
24	UBadBad701016	UVA	Badulla	ES_04	GP2	EDMI_Mk_10E	1906	['meter_model_issue']	['replace_meter']	['K0721']	1	7/1/2024
25	UBadBad701016	UVA	Badulla	ES_04	I2	EDMI_Mk_10E	0	['meter_model_issue']	['replace_meter']	['K0721']	1	7/1/2024
26	UBadBad701017	UVA	Badulla	ES_04	GV2	EDMI_Mk_10E	18436	['meter_model_issue', 'i	['replace_meter', 'replace_met	['K0721', 'K072	2	7/1/2024
27	UBadBad701018	UVA	Badulla	ES_04	GP2	EDMI_Mk_10E	347	['meter_model_issue']	['replace_meter']	['K0721']	1	7/1/2024
28	UBadBad701018	UVA	Badulla	ES_04	H2	EDMI_Mk_6N_(	9650	['meter_model_issue']	['replace_meter']	['K0721']	1	7/1/2024

Figure 7.8: Snap shot of Maintenance Activity Plan provided as an output by EA agent.

7.5 Testing of ES agent

Maintenance Activity Plan produced by the EA agent was given for the ES agent to prepare a Maintenance Activity schedule. Required working calendar and the location data set also given for the ES agent. ES agent was capable of producing a fair Maintenance Activity schedule and the sample of results are shown in Figure 7.8.

	Consumer_ID	Province	Area	Assigned_	Consumer	GPS_Coordinates	Avg_Consumption	Maintenance_Activities	Total_Time	Planned_date	Traveling
2	UDiyUva70101504	UVA	Diyathalawa	ES_03	I2	[6.92877492, 80.85863691]	13692	['test_theMeter']	11.12975	1/1/2024	336.679
3	UDiyWel70101857	UVA	Diyathalawa	ES_03	H2	[6.903381141, 80.90205179]	11659	['test_theMeter']	11.984	1/2/2024	184.0032
4	UDiyWel70101911	UVA	Diyathalawa	ES_03	GP2	[6.904057518, 80.89151147]	1909	['test_theMeter']	12.97954167	1/3/2024	170.4934
5	UDiyWel70101512	UVA	Diyathalawa	ES_03	GP2	[6.894486597, 80.91635107]	104	['test_theMeter']	14.09379167	1/4/2024	178.6464
6	UDiyBan70101881	UVA	Diyathalawa	ES_03	GP2	[6.806221232, 81.0014769]	8536	['test_theMeter']	14.884	1/5/2024	201.1179
7	UDiyUva70101911	UVA	Diyathalawa	ES_03	GP2	[6.926733797, 80.93144001]	1909	['test_theMeter']	17.26120833	1/8/2024	208.8407
8	UDiyEll70101857	UVA	Diyathalawa	ES_03	H2	[6.867601575, 81.04556266]	11659	['test_theMeter']	19.01183333	1/9/2024	226.8229
9	UDiyDiy70101628	UVA	Diyathalawa	ES_03	GP2	[6.767129564, 80.95463032]	5364	['test_theMeter']	19.38695833	1/10/2024	188.8142
10	UDiyWel70101504	UVA	Diyathalawa	ES_03	I2	[6.906510368, 80.91080389]	13692	['test_theMeter']	21.035375	1/11/2024	197.8787
11	UDiyWel7010183(	UVA	Diyathalawa	ES_03	I2	[6.859104692, 80.92801181]	5085	['test_theMeter']	22.25641667	1/12/2024	194.6546
12	UDiyEll70101342	UVA	Diyathalawa	ES_03	GP3	[6.890934765, 81.04789044]	129746	['replace_meter', 'test_th	23.96654167	1/15/2024	214.5928
13	UDiyDiy70100907	UVA	Diyathalawa	ES_03	I2	[6.753653725, 80.93861605]	112670	['replace_meter']	24.29245833	1/16/2024	192.8375
14	UDiyDiy70100915	UVA	Diyathalawa	ES_03	I2	[6.758021687, 80.92898656]	107936	['replace_meter']	25.25429167	1/17/2024	161.5255
15	UDiyEll70101288	UVA	Diyathalawa	ES_03	GP3	[6.857707534, 81.02253707]	90114	['replace_meter']	26.82404167	1/18/2024	205.6935
16	UDiyBan70100842	UVA	Diyathalawa	ES_03	GP2	[6.918423635, 80.96216399]	77210	['replace_meter', 'test_th	29.317875	1/19/2024	210.173
17	UDiyDiy70100168	UVA	Diyathalawa	ES_03	GP2	[6.751347842, 80.8729731]	71685	['replace_meter', 'replace	31.18345833	1/22/2024	202.134
18	UDiyDiy70100095	UVA	Diyathalawa	ES_03	I2	[6.74340235, 80.86439198]	50590	['replace_meter']	32.31975	1/23/2024	160.1857
19	UDiyBan70100850	UVA	Diyathalawa	ES_03	GP2	[6.934457646, 80.98947126]	47837	['replace_meter']	35.16129167	1/24/2024	232.3434
20	UDiyDiy70100141	UVA	Diyathalawa	ES_03	GP2	[6.742504978, 80.87387343]	37826	['replace_meter']	36.360125	1/25/2024	201.0768
21	UDiyUva70101350	UVA	Diyathalawa	ES_03	I2	[6.932779793, 80.86650073]	37418	['replace_meter', 'replace	39.972125	1/26/2024	217.5451
22	UDiyWel7010135(	UVA	Diyathalawa	ES_03	I2	[6.843825183, 80.88199536]	37418	['replace_meter', 'replace	41.976375	1/29/2024	198.9557
23	UDiyBan70101776	UVA	Diyathalawa	ES_03	I2	[6.890919413165288, 79.9597	37368	['replace_meter']	42	1/30/2024	181.2859
24	UDiyBan70100389	UVA	Diyathalawa	ES_03	I2	[6.842599691, 80.97601943]	33539	['replace_meter']	51.74220833	1/31/2024	354.1032
25	UDiyBan70100559	UVA	Diyathalawa	ES_03	I2	[6.916439962, 80.98373982]	28826	['replace_meter']	53.097125	2/1/2024	203.1558
26	UDiyBan70100494	UVA	Diyathalawa	ES_03	I2	[6.883501801, 80.97049953]	28757	['replace_meter']	54.01295833	2/2/2024	190.3172
27	UDiyDiy70100184	UVA	Diyathalawa	ES_03	GP2	[6.744239996, 80.89312558]	28105	['replace_meter']	55.41995833	2/5/2024	197.8383

Figure 7.9: Snapshot of the Maintenance Activity Schedule produced by the ES agent.

7.6 Testing of SK agent

Maintenance Activity Plan produced by the EA agent was given for the SK agent to prepare a Material Plan. During the evaluation of SK agent, certain accuracy issues were found and those have been also corrected in the given algorithm. Finally, SK agent was capable of producing a Material Plan with material demand and material order quantity. Stock database given to the SK agent as input is given in Figure 7.9 and the output plan is given in Figure 7.10

1	Materials	Stock_Level	Min_Stock	Min_Order_Qty
2	MD_K0709	4000	3000	12000
3	BATTARY	80	100	200
4	MD_K0726	30	20	100
5	K0709	200	1000	12000
6	K0726	30	50	100
7	MD_K0721	200	150	300
8	MD_K0778	1500	1200	3000
9	K0778	1500	1200	3000
10	K0721	400	300	400

Figure 7.9: Stock data base given to SK agent as an Input.

1	Materials	Stock_Level	Material_Demand	Order_Qty
2	K0726	30	26	100
3	K0721	400	454	400

Figure 7.10: Material plan produce by SK agent.

7.7 Testing of MA agent and the overall system

For the purpose of testing the overall system, the same data set was given to human professionals who used to do the maintenance activity planning tasks. The meter data base consists of 480 numbers of consumer locations and 122 numbers of maintenance requests. The testing took place at three stages, first the human professional was given the task without any methodology, secondly the same task with the methodology used in the developed system partially which was implemented in Microsoft Excel using Excel functions and the third using the developed system. First stage took on average one minute for planning maintenance activity for one consumer meter which would take around eight hours to complete the task. Second stage took around 18 minutes for 480 number of meters without material plan, traveling distance and traveling time. In the third stage the developed system took only 1.5 minutes to prepare maintenance activity plan, maintenance activity schedule and material plan including traveling distance and time. This shows the developed Multi Agent Metering management system can produce management plans quickly with more details than done by human professionals.

7.8 Summary

This chapter evaluated the experimental implementation using data relevant to some of the Smart Meters in Sri Lanka. The experimental results showed that the implemented Metering Ontology, MA agent, EA agent, SK agent and ES agents can provide acceptable output. The overall system also can provide the required

Maintenance Activity plan, Maintenance activity schedules, Material plan and policy evaluation report with an acceptable accuracy. The next chapter concludes the research study highlighting limitations of the system and the further work.

.

CHAPTER 8

CONCLUSION AND FURTHER WORK

8.1 Introduction

Previous chapter discussed the evaluation results of the experimental implementation of the MAS based Electricity Metering Management System. This chapter provides the overall conclusion, objective vice conclusions, limitations of the developed system and further works.

8.2 Conclusions

In this study, the possibility of modeling Electricity Metering Management as a Multi Agent Planning System was explored. Knowledge in the field of Electricity Meter Management was separated into domain knowledge and problem solving knowledge. Domain knowledge was modeled using OWL ontology. Problem solving knowledge was modeled into the agent's skills in the Multi Agent System. This Multi Agent System was modeled specifically as a Multi Agent Planning problem considering the tasks allocated for the agents in the system. Developed system was provided domain knowledge as an OWL ontology, meter details as a CSV file, locations of consumer premises as CSV file, stock details of material as a CSV file, issues related to meters also as a CSV file and also working days of employees that implement maintenance activities as a CSV file.

There were four objectives of this study as below.

I. Critical Review of Literature in electricity metering.
II. In-depth study about technologies used in management of electricity metering.
III. Design and develop a Multi Agent System based solution for electricity meter management.
IV. Evaluate the Multi Agent System based electricity meter management solution.

The first objective was to critically review the literature in electricity metering. After conducting this review, it was able to find the research gap of challenge in managing large metering systems with different type of meters. Therefore, the first objective was achieved. The second objective was to study the technologies used in management of electricity metering systems. Even though there were no studies done on managing electricity metering system, it was found that modeling this problem as a multi agent planning problem is more appropriate capitalizing the natural way of humans attend to this problem. With that second objective also achieved. Third objective was to develop a Multi Agent based solution for electricity meter management. The solution was developed in python using SPADE module and an OWL ontology achieving third objective also. The fourth objective was to Evaluate the Multi Agent System based electricity meter management solution. The developed system was tested using data of 480 numbers. Testing was done in three stages; first by human employee just assigning the task, second using a developed spread sheet by human employee and finally by the developed system. The evaluation proved that the developed system is capable of producing required plans more quickly and with more details achieving the fourth objective.

8.3 Limitations and further work

For the purpose of testing the proposed concept, only a limited set of meters and limited number of issues were considered. Therefore, for the purpose of full implementation of this system, domain knowledge relevant to all types of available meters and the related issues are required to model in the ontology. Also, the time required for maintenance activities and travelling have the space for further fine tuning using the experience in practical implementation of the produced plans. The developed solution being a Multi Agent system, it will be possible to integrate learning in to planning agents as the real implementation of produced plans are going on. Therefore, expanding the ontology for all types of meters, related issues, suitable solutions, required time and material and integrating learning in to agents would be considered as future work.

8.4 Summary

This chapter, the final chapter, provided the conclusion, limitation and future works of the study of modeling Electricity Meter Management as a Multi Agent System. Evaluation of system including domain knowledge modeling as an OWL ontology as well as the implementation of problem solving as a Multi Agent System were considered in drawing the conclusions. Further, the limitations and future developments also had been discussed.

REFERENCES

[1] "Outlook for electricity – World Energy Outlook 2022 – Analysis," IEA. Accessed: Sep. 01, 2023. [Online]. Available: https://www.iea.org/reports/world-energy-outlook-2022/outlook-for-electricity

[2] "Snapshot." Accessed: Mar. 19, 2024. [Online]. Available: https://www.iea.org/world/electricity

[3] "1656311266CEB-Statistical-Digest-2021-cover.pdf." Accessed: Mar. 19, 2024. [Online]. Available: https://www.ceb.lk/front_img/img_reports/1656311266CEB-Statistical-Digest-2021-cover.pdf

[4] A. Chakrabarti and S. Halder, *Power System Analysis: Operation And Control 3Rd Ed.* PHI Learning Pvt. Ltd., 2010.

[5] N. S. Zivic, O. Ur-Rehman, and C. Ruland, "Evolution of smart metering systems," in *2015 23rd Telecommunications Forum Telfor (TELFOR)*, Belgrade, Serbia: IEEE, Nov. 2015, pp. 635–638. doi: 10.1109/TELFOR.2015.7377547.

[6] J. F. Martins, A. G. Pronto, V. Delgado-Gomes, and M. Sanduleac, "Smart Meters and Advanced Metering Infrastructure," in *Pathways to a Smarter Power System*, Elsevier, 2019, pp. 89–114. doi: 10.1016/B978-0-08-102592-5.00004-1.

[7] D. B. Avancini, J. J. P. C. Rodrigues, S. G. B. Martins, R. A. L. Rabêlo, J. Al-Muhtadi, and P. Solic, "Energy meters evolution in smart grids: A review," *Journal of Cleaner Production*, vol. 217, pp. 702–715, Apr. 2019, doi: 10.1016/j.jclepro.2019.01.229.

[8] O. P. Mahela *et al.*, "Comprehensive overview of multi-agent systems for controlling smart grids," *CSEE Journal of Power and Energy Systems*, vol. 8, no. 1, pp. 115–131, Jan. 2022, doi: 10.17775/CSEEJPES.2020.03390.

[9] Asian Development Bank, "Sri Lanka Energy Sector Assessment, Strategy and Road Map:," Asian Development Bank, Manila, Philippines, Dec. 2019. doi: 10.22617/TCS190557-2.

[10] "LECO Annual Report 2022 - English.pdf." Accessed: Apr. 03, 2024. [Online]. Available: https://www.leco.lk/content/files/publication/LECO%20Annual%20Report%202022%20-%20English.pdf

[11] Z. Ma, M. J. Schultz, K. Christensen, M. Værbak, Y. Demazeau, and B. N. Jørgensen, "The Application of Ontologies in Multi-Agent Systems in the Energy Sector: A Scoping Review," *Energies*, vol. 12, no. 16, p. 3200, Aug. 2019, doi: 10.3390/en12163200.

[12] B. Teixeira, T. Pinto, F. Silva, G. Santos, I. Praça, and Z. Vale, "Multi-Agent Decision Support Tool to Enable Interoperability among Heterogeneous Energy Systems," *Applied Sciences*, vol. 8, no. 3, p. 328, Feb. 2018, doi: 10.3390/app8030328.

[13] A. Anvari-Moghaddam, A. Rahimi-Kian, M. S. Mirian, and J. M. Guerrero, "A multi-agent based energy management solution for integrated buildings and

microgrid system," *Applied Energy*, vol. 203, pp. 41–56, Oct. 2017, doi: 10.1016/j.apenergy.2017.06.007.

[14] A. Anvari-Moghaddam, J. M. Guerrero, A. Rahimi-Kian, and M. S. Mirian, "Optimal real-time dispatch for integrated energy systems: An ontology-based multi-agent approach," in *2016 IEEE 7th International Symposium on Power Electronics for Distributed Generation Systems (PEDG)*, Vancouver, BC, Canada: IEEE, Jun. 2016, pp. 1–7. doi: 10.1109/PEDG.2016.7526997.

[15] G. Santos, T. Pinto, I. Praca, and Z. Vale, "EPEX ontology: Enhancing agent-based electricity market simulation," in *2017 19th International Conference on Intelligent System Application to Power Systems (ISAP)*, San Antonio, TX, USA: IEEE, Sep. 2017, pp. 1–6. doi: 10.1109/ISAP.2017.8071411.

[16] S. Ma, H. Zhang, and X. Xing, "Scalability for Smart Infrastructure System in Smart Grid: A Survey," *Wireless Pers Commun*, vol. 99, no. 1, pp. 161–184, Mar. 2018, doi: 10.1007/s11277-017-5045-y.

[17] K. Parvin *et al.*, "The future energy internet for utility energy service and demand-side management in smart grid: Current practices, challenges and future directions," *Sustainable Energy Technologies and Assessments*, vol. 53, p. 102648, Oct. 2022, doi: 10.1016/j.seta.2022.102648.

[18] M. B. Rasheed, M. A. Qureshi, N. Javaid, and T. Alquthami, "Dynamic Pricing Mechanism With the Integration of Renewable Energy Source in Smart Grid," *IEEE Access*, vol. 8, pp. 16876–16892, 2020, doi: 10.1109/ACCESS.2020.2967798.

[19] S. K. Rathor and D. Saxena, "Energy management system for smart grid: An overview and key issues," *Int J Energy Res*, vol. 44, no. 6, pp. 4067–4109, May 2020, doi: 10.1002/er.4883.

[20] M. Ahmed *et al.*, "Energy Theft Detection in Smart Grids: Taxonomy, Comparative Analysis, Challenges, and Future Research Directions," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 4, pp. 578–600, Apr. 2022, doi: 10.1109/JAS.2022.105404.

[21] M. Shokry, A. I. Awad, M. K. Abd-Ellah, and A. A. M. Khalaf, "Systematic survey of advanced metering infrastructure security: Vulnerabilities, attacks, countermeasures, and future vision," *Future Generation Computer Systems*, vol. 136, pp. 358–377, Nov. 2022, doi: 10.1016/j.future.2022.06.013.

[22] A. Dorri, S. Kanhere, and R. Jurdak, "Multi-Agent Systems: A survey," *IEEE Access*, vol. 6, pp. 1–1, Apr. 2018, doi: 10.1109/ACCESS.2018.2831228.

[23] H. S. Nwana, "Software agents: an overview," *The Knowledge Engineering Review*, vol. 11, no. 3, pp. 205–244, Sep. 1996, doi: 10.1017/S026988890000789X.

[24] A. Torreño, E. Onaindia, A. Komenda, and M. Štolba, "Cooperative Multi-Agent Planning: A Survey," *ACM Comput. Surv.*, vol. 50, no. 6, pp. 1–32, Nov. 2018, doi: 10.1145/3128584.

[25] "Protege 5 New OWL Pizza Tutorial V3.2.pdf," Google Docs. Accessed: Apr. 27, 2024. [Online]. Available: https://drive.google.com/file/d/1A3Y8T6nIfXQ_UQOpCAr_HFSCwpTqELeP/view?usp=sharing&usp=embed_facebook