Proceedings Paper DOI: 10.58190/icat.2023.36

PROCEEDINGS OF

INTERNATIONAL CONFERENCE ON ADVANCED TECHNOLOGIES

https://proceedings.icatsconf.org/

11th International Conference on Advanced Technologies (ICAT'23), Istanbul-Turkiye, August 17-19, 2023.

Toward Competent City Management using Multi-agent System (MAS)

Alaa M. Odeh*, Amjad Rutrot +, Rashid Jayousi *

*Department of computer science, Al-Quds University, Jerusalem, Palestine

Alaa.odeh4@students.alquds.edu

+ *Department of computer science, Arab American University (AAUP), Ramallah, Palestine*

amjad.rattrout@aaup.edu

**Department of computer science, Al-Quds University, Jerusalem, Palestine*

rjayousi@staff.alquds.edu

***Abstract*— Multi-agent system plays a crucial role in many aspects of life. This refers to its ability to provide scalable and flexible solutions to complex problems. One of these problems is the management of resources in different cities around the world. In this study, we have used the multi-agent system concepts to build competent city resources management. Because of the time and software complexity that arises from implementing the whole system, we chose to implement one city entity, the energy entity. Petri net is used to model the system mathematically. At the same time, we used CPN tools software to simulate the system behaviour in real life. The mathematical model and simulation results proved that the designed energy MAS would achieve its goals when applied in real life. In the future, we intend to complete the construction of the whole system for better resource management of all entities in the city.**

***Keywords*— multi-agent system, Petri nets, CPN tools, FIPA ACL.**

I. Introduction

Although there are not some crucial issues that we would not be aware of, many serious facts are unknown or known only to a limited number of people. One of these issues is the rapid growth of people worldwide. This issue raises concerns about the production and progress in meeting people's needs [1]. The inadequacy of meeting people's needs worldwide could cause many problems in their lifestyles. These problems are related to the economy, education, health, and many other areas. Overcoming these problems is considered to be one of the essential matters in countering humanity. One prime proposed solution is efficient resource management to satisfy personal needs despite massive population growth. This research suggests a multi-agent solution to how efficient city resource management can be accomplished.

A multi-agent system, in general, is a collection of intelligent agents that work together, forming a system. Multi-agent systems address problems that a single agent or conventional method cannot solve effectively [2]. We propose an approach that uses a group of intelligent agents, where each agent is responsible for accomplishing a particular task. These agents will interact with each other so that, in the end, the main aim of the system is, in this case, to accomplish optimal resource management. The resources in any city can be divided into several categories, and a specific entity handles each type; we suggested dividing the whole system into a group of subsystems where each subsystem expresses an entity in the city and is comprised of multiple agents that interact with each other in the same subsystem and with other agents in different subsystems. This way, each subsystem will guarantee sufficient management of the resources in its entity and eventually ensure competent resource management. Due to several limitations that prevented us from performing a real-life experiment, we used CPN tools software to simulate our system.

This study is organized as follows: Section 2 discussed the related work on employing multi-agent systems in resources management. In section 3, we presented our research's problem statement and objectives. In section 4, we introduced the proposed system model. Section 5 describes the system architecture. In section 6, we constructed the mathematical system model. Section 7 illustrates the simulation in CPN tools software. And in section 8, we discussed the results and future directions.

II. RELATED WORK

The fast-paced expansion of the world's population creates new duties and obligations for humanity. This growth can lead to numerous economic and social difficulties in the future. Addressing these challenges and securing the resources essential for human existence are among humankind's most crucial concerns. In [1], the author presented the problems that may occur because of the rapid population growth and the reasons for them. After analysing the present situation and assessing the available opportunities, they found that the decline of natural resources and adverse environmental changes will result in socioeconomic difficulties, particularly as the world's population continues to overgrow. They indicated that meeting available resources with the rapid growth in people comprises humankind's prime need. [3] In this context, tested and compared different strategies for allocating resources and establishing priorities using the home automation system. Although the researchers presented strategies for efficient energy management and optimal resource utilization, this was limited to a home automation system. However, authors in [4] describe a Smart City model that utilizes Multi-Agent Systems and IoT, giving cities intelligence through basic infrastructure. Their model is designed to be repeatable and portable, forming a blueprint for scalable urban development. The inability to apply smartness to entities in every country is a difficult task at hand. In the paper [5], the authors analysed, designed, and developed a software-based simulation tool using agents to replicate a Smart City's dynamic behaviour. While [6] outlined multiple MultiAgent System (MAS) models created to tackle diverse urban planning issues at various spatial and temporal scales, including city and regional growth over several decades, years-long regeneration and gentrification, and daily commuting habits. Paper [7] examines the advancements in Multi-Agent Systems (MAS) for energy consumption optimization. In this field, MAS has primarily been applied to demand response, simulation of human behaviour in smart buildings, and management of Wireless Sensor Networks for optimization decisions. These three areas have advanced as MAS has advanced and have been used to model related problems. However, work like [8] presents a Multi-Agent System designed to gather and manage information from potato crops, enabling the implementation of a precision irrigation system. The proposed MAS is built on the Cloud Computing framework and can collect data from Wireless Sensor Networks (WSNs) in potato fields for knowledge extraction and decision-making. Relating to the employment of MAS in the educational system, [9] propose a multi-agent system based on AI techniques capable of performing broader analyses of learning and teaching processes. While [10] outlines the design of a Multi-Agent System (MAS) architecture using JADE. An employer can utilize the MAS during an interview to assess whether a job candidate possesses the skills and expertise required in the job description. However, studies that addressed separated resource management systems didn't delve into smartly managing this separated resource and its effect on other resources or entities in the same city. Studies that addressed intelligent city management using multi-agent systems needed to show relations and interactions between agents in their proposed approach. Moreover, studies that involved using Petri nets were mainly simple multi-agent systems, and none of them-as far as we know- modelled smart city management systems. So, our contribution is mainly centred on the following:

i. Utilizing Multi-Agent System (MAS) concepts, we developed a robust model using coloured Petri net to achieve optimal resource allocation by effectively relating all entities in the system. Our study contributes to the advancement of resource allocation optimization in MAS by leveraging CPN tools software to simulate system behaviour.
ii. Our research focused on modelling the Energy Multi-Agent System (EMAS) in the urban environment using Petri nets, enabling us to thoroughly analyse and understand the dynamic behaviour and interaction flow between agents. By utilizing this approach, we make a significant contribution to the field of MAS, particularly in the area of urban energy management.
iii. Our study employed the CPN tools software to simulate the behaviour of the EMAS, resulting in insightful observations and findings. Through the utilization of Petri nets, we have contributed to the development of an efficient and accurate method for modelling MAS, offering researchers and practitioners a powerful tool to investigate complex systems.

III. PROBLEM STATEMENT AND OBJECTIVES

There are many sources available around the world. Still, the exponential increase in the number of people and the mismanagement of these sources is an obstacle to aligning the number of people in different cities with available sources. Proposed solutions using artificial intelligence to manage these sources assume that AI can be applied to all cities. Still, this solution can only be applied to some of the world's cities as some countries, especially third-world countries must be equipped for such a massive change. An intelligent management system of entities inside cities based on multi-agent systems is introduced. This proposed system will help enhance the allocation of resources to all residents in some cities by using software agents that continuously analyse received information and interact with each other for optimal behaviour and hence, optimal management of the whole city.

A. *Objectives of Research*

i. To study and explore the concepts of a multi-agent system.
ii. To design an approach for a city management system to ensure optimal resource allocation to all residents.
iii. To use CPN tools software for simulating and analysing the proposed system.

IV. PROPOSED SYSTEM MODEL

The main goal of our system is to achieve optimal city management. From this point, we divided our MAS into several subsystems based on the entity it stands for. Each of these subsystems contains several agents that cooperate to achieve optimal performance. However, the proposed system is considered complex because, in each entity, there are several agents. Besides communicating with each other, these agents are supposed to perform their tasks. This leads us to think about a challenge that involves analysing and understanding the behaviour and the flow of the events inside each subsystem. We propose a mechanism to handle this challenge using different methods. i.e., to realize the behaviour and the flow of the events inside each subsystem, we suggested modelling the system using Petri Nets. And to understand how the system might behave in an actual situation, we will use CPN tools simulation.

A. *PETRI NETS MODELING*

Petri Nets, named after Carl Adam Petri, are a fundamental model for representing parallel and distributed systems. The central concept is to depict changes in a system's state using transitions. A Petri Net consists of arcs that connect places and transitions, where places can contain tokens [11]:

i. Places symbolize states, conditions, or resources that must be available before action occurs.
ii. Transitions symbolize actions.
iii. A token in a place means that the corresponding condition is fulfilled or that a resource is available

Fig. 1 presents a simple net containing all elements in the Petri net:

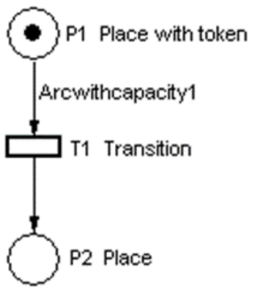

Figure 1: simple petri net diagram

From this, we can conclude that to represent a system using Petri net, we first have to define the states of the system, the actions, and what resembles tokens. Thus, when we use the Petri net to represent the multi-agent system, the places in the Petri net stand for the system states, the transitions in the Petri net stand for actions, tokens in the Petri net stand for the data that agents have in this system, and the flow of tokens through the Petri net represents the interactions between agents. Generally speaking, If we assume to have a finite set of states or conditions in our system and a finite set of actions and a group of agents that flow from one state to another, we can represent this mathematically by this equation:

$$N = \langle P, T, I, O, m_0 \rangle \qquad \text{Eq. 4.1}$$

The multi-agent system here is a tuple N, where:

i. P is the finite Set of system states
ii. T is the finite Set of actions
iii. I Is the Set of input arcs
iv. O is the Set of output arcs.
v. M_0: P → IN is the initial marking representing the initial distribution of agents

There are various types of Petri nets, this includes simple Petri nets, coloured Petri nets, timed nets, and stochastic nets. The simple Petri net is like the definition: a digraph containing places and transitions connected using arcs. The coloured Petri net, on the other hand, is an extended Petri net where we can differentiate tokens by colours. Whereas in timed Petri nets, the transitions are attached to delay to allow them to model time. Unlike stochastic Petri nets, we link delays to places, not tokens. Since our multi-agent system consists of agents holding information, the most suitable Petri net to use is the coloured Petri net (CPN). Before modelling the system using CPN, a sufficient description of the system architecture is needed to understand the system clearly

V. SYSTEM ARCHITECTURE

Because our system includes multiple entities, which in turn have multiple agents, we used a concept map to identify better the system, agents, and agents' behaviour. This concept map demonstrates all entities in the city, the subsystems in each entity, the agents in each subsystem, and their possible actions. This concept map can be found in the paper appendix. However, due to time and software limitations, we only worked with one entity: the energy entity, its subsystems, agents, and their behaviours. Working on the other entities will be completed in the near future.

A. *ENERGY ENTITY*

In each city, an energy entity is responsible for the fair distribution of energy among all countries in this city. Moreover, this entity is in charge of figuring out the amount of energy consumed by each country to monitor whether any country has excessive energy consumption. This entity has to handle all issues related to the misuse of energy, which might lead to permanent damage. Thus, we will clarify the basic subsystems in this entity and agents and their behaviours in each subsystem.

1. *The basic subsystems in the energy entity*

Fig. 2 shows the entities in general and the energy entity's subsystems.

1) *First Subsystem: Energy supply for all buildings*

This subsystem consists of agents with different roles. However, the data they collect and their communication will guarantee that each building will have a fair amount of energy, even those in remote countries.

a) *Counting agents*

As the name suggests, this agent's task is to count the number of buildings and people inside each country and create a list of the buildings' types. The number of counting agents inside each country will be according to the country's area thus:

i. Assign one counting agent for each 5 Km area.
N_{bC} = country's area/5
ii. This agent will communicate with the sorting agents to give them the list of buildings it created.
iii. Types of counting agents: these agents can be software agents that receive data from sensors installed inside the buildings.
iv. Time of working: in addition to counting buildings,
v. These agents count people inside the building, so

they initially count the number of people and building when first installed and then update their status when a new building is constructed or when new people enter the building.

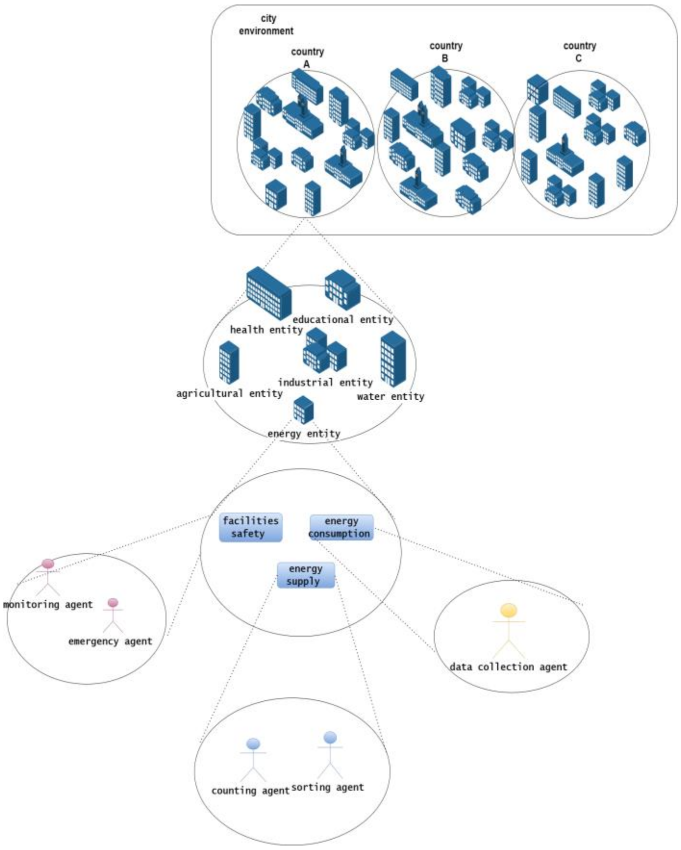

Figure 2:main entities in the system

b) *Sorting agents*

i. Sorting agents receive the list of building inside a specific country and sort the building into several categories. Such as: health, educational, industrial, and residential categories.

ii. The number of sorting agents will be compatible with the number of counting agents such that for each counting agent, there will be a sorting agent which receives the list and categorizes them

iii. Types of sorting agents: these agents can also be software that categorizes buildings according to an algorithm.

iv. In addition to communication with counting agents, sorting agents will post the categories they created to a blackboard so the data collection agents can use it.

2) Second subsystem: Energy consumption information

This subsystem's purpose is to calculate energy consumption. Two assigned agents work together to complete this task. The agents are:

1) Data collection agent

i. The data collection agent calculates the amount of energy consumed by each building through build-in sensors that send data about each building's consumption

ii. It communicates with the counting agent and gets the number of buildings and people in each country.

iii. From previous calculations, It has an initial energy consumption value for each building. When it gets the number of people in this country, it calculates the amount of energy consumption with this equation:

$$\text{Amount of consumed energy} = \sum_{1}^{n} (\text{initial consumption value})$$

where k is the number of buildings

i. It then stores the amount of consumption in a database.

ii. Types of data collection agent can be software that continuously receives the number of buildings and calculate the consumption

2) City energy consumption agents

i. Similar to the job done by the data collection agents, but instead of calculating the amount of energy consumption, these agents calculate the energy consumption of the whole city using this equation:

$$\text{Amount of consumed energy} = \sum_{1}^{n} (\text{initial consumption value})$$

ii. Where n is the number of countries in this city They then store the amount of consumption in a database for later comparison with other cities

iii. There is a threshold amount for energy consumption. This threshold depends on the amount of other cities' consumption with a similar number of residents and factories.

iv. When this agent detects excess energy usage, it performs some improvements, including either legalization of power consumption, if possible, or using an alternative power supply

3) Third subsystem: Citizens' and facilities' safety

This subsystem helps keep different facilities safe from any danger that might happen due to some electrical faults. The agents that are working together to make this happen are:

a) Monitoring agents

i. Monitoring agents continuously monitor the electrical circumstance of the area they are responsible for.

ii. When they detect a possible electrical danger, they immediately alarm the interested parties so they can take immediate action if needed.

iii. Different types of sensors can be used in each building to send data to the monitoring agents

b) Emergency agents

i. It acts immediately When it receives an alarm from the monitoring agent with the specified location

ii. Types of Agents that communicate with emergency agents are communication agents in the health entities that will take immediate action if there are any injuries or communication agents in the fire department when there is a fire in the building.

VI. THE SYSTEM MATHEMATICAL REPRESENTATION

After we define the agents in the system, their behaviours, and the actions they perform, in this section, we will present the mathematical model of the system.

A few steps should be followed to represent the system using Petri nets. First, we will present the structure analysis of the system. This includes identifying the places, the transitions, the input arcs, the output arcs, and the markings. For more clarification, we have designed a simple Petri net that contains possible states of the system when some agent receives data or takes action. However, the next section will thoroughly explain all agents' states and actions. Fig. 3 demonstrates this simple Petri net.

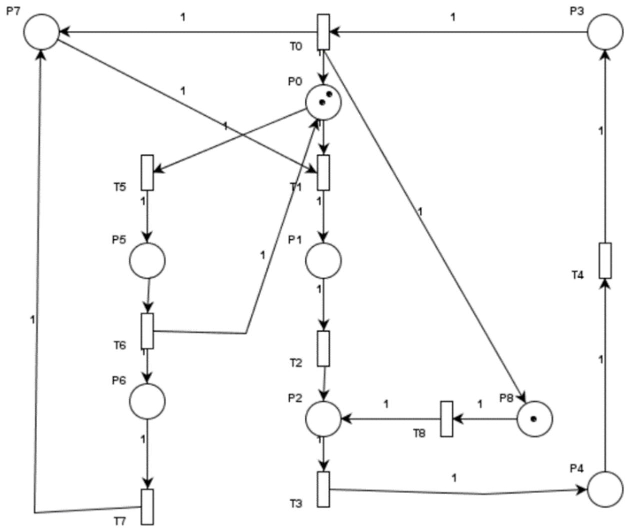

Figure 3:Energy system simple timed Petri net

From the figure, we can identify the following:

1. P (set of places in the system) = {P0, P1, P2, P3, P4, P5, P6, P7}, while:
   - P0: state of the system when the counting agent is ready to send data to the sorting agent and data collection agent
   - P1: state of the system when the sorting agent receives the building list from the counting agent and categorizes the list
   - P2: state of the system when the monitoring agent receives the sorted list from the sorting agent
   - P3: state of the system when the emergency agent receives an alarm from the monitoring agent
   - P4: state of the system when the external agents receive messages from the emergency agent.
   - P5: state of the system when the data collection agent receives the number of buildings from the counting agent and calculates the consumed energy of all buildings.
   - P6: state of the system when the city agent receives the amount of consumed energy from the data collection agent and calculates the consumed energy for the whole city.
   - P7: state of the system when the database is updated.
   - P8: state when having an emergency or alarm.
2. T (set of system transition) = {T0, T1, T2, T3, T4, T5, T6, T7, T8}, while:
   - T0: the action of updating the database
   - T1; the action of sending a list of the building from the counting agent to the sorting agent
   - T2: the action of sorting building into different categories and sending these categories under request to other agents
   - T8: the action of detecting danger from the surrounding and alarming the monitoring agent
   - T3: the action of evaluating the situation and send to the emergency agent when the danger is classified as a high risk
   - T4: requesting intervention from an external agent in a high-risk situation.
   - T5: sending the number of buildings to the data collection agent from the counting agent.
   - T6: the action of calculating the consumed energy and sending the calculated amount to the city agent
   - T7: calculating the energy consumed by the whole city, identifying the excessive consumption, and creating history.
3. I (set of input arcs) and O (Set of output arcs). Table 1 contains the I and O of each transition:

Transition	Input arc	Output arc
T0	P3	P0, P7
T1	P0	P1
T2	P1	P2
T3	P2	P4
T4	P4	P3, P0
T5	P0	P5
T6	P5	P0, P6
T7	P6	P7
T8	P8	P2

Table 1: Transitions and arcs of the system

4. Marking of the Petri net after each transition:

Marking is the number of tokens in each place(state) after each transition. Markings of the system can be represented using vectors where numbers in this vector represent the number of tokens at each place after the specified transition. We first introduce a graph demonstrating the places containing tokens after each transition to understand the marking vectors. This graph is presented in fig.4.

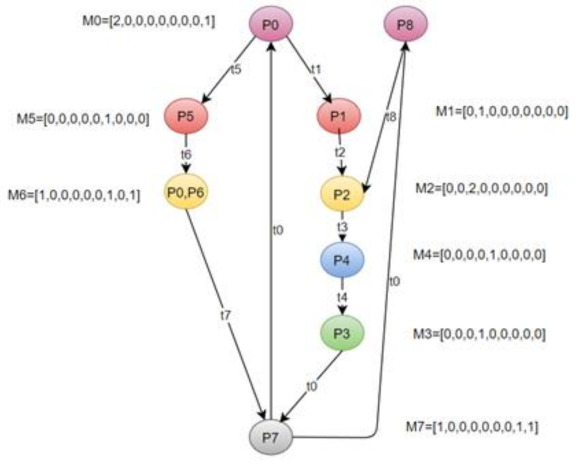

Figure 4: Reachability graph of the energy MAS

The graph illustrated in Fig.4 represents a reachability graph in Petri net. The reachability graph of a Petri net is a directed graph that shows all the possible states that the Petri net can reach from its initial marking.

The incidence matrix of the energy MAS is:

	T0	T1	T2	T3	T4	T5	T6	T7	T8
P0	1	-1	0	0	0	-1	1	0	0
P1	0	1	-1	0	0	0	0	0	0
P2	0	0	1	-1	0	0	0	0	1
P3	-1	0	0	0	1	0	0	0	0
P4	0	0	0	1	-1	0	0	0	0
P5	0	0	0	0	0	1	-1	0	0
P6	0	0	0	0	0	0	1	-1	0
P7	0	0	0	0	0	0	0	1	0
P8	1	0	0	0	0	0	0	0	-1

Column T represents how each transition's firing affects the system's marking.

From here, we need to prove that our Petri net is reachable. This means all possible states can be reached from the initial state by firing its transitions.

In the context of a multi-agent system, all agents can achieve their goals under the specified constraints and by satisfying dependencies on other agents.

For this, let us start now by writing the initial marking as a column vector:

$$M_0 = \begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Similarly, we can represent firing transitions as column vectors with one entry for each transition. For example, if we assume that each of the transitions t0, t1, t2, t3, t4,t5, t6, t7, and t8 is fired once:

Enabled transitions

T0	T1	T2	T3	T4	T5	T6	T7	T8
no	yes	no	no	no	yes	no	no	yes

We can express this using a column vector. We will call this vector y. And hence, the resulting vector is:

$$y = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

The result of firing these transitions can be computed from this equation:

$$M^{`} = M_0 + I.y \qquad \text{Eq. 5.1}$$

Where:

- M` is the resulting matrix
- M_0 is the initial matrix
- I is the incidence matrix
- y is the vector representing what enables the transition

By representing the equation with matrices:

$$\begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} . \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Generally speaking, Let N be a Petri net with incidence matrix I, and let M`, M0 be two markings of N,

The following implication holds:

- If M` ∈ reach(M), then there exists a vector y such that:
- $M^{`} = M + I \cdot y$

Such that all entries in y are natural numbers.

On the other hand, we can use the above equation to show that some marking is not reachable. We mean that if $M^{`} = M + I \cdot y$ has no solution for y; then we can say that M` is unreachable.

In the context of a multi-agent system, checking for unreachability is essential because it helps ensure that the system functions correctly and that no unwanted behavior can occur. In other words, if some state in Petri net is unreachable, this means that this state cannot be reached from the initial state. This leads to situations where agents cannot achieve their goals. Therefore, checking unreachability is essential in verifying and validating a multi-agent system.

Thus, we need to find y for each marking in the above equation to ensure that our system is reachable and functions correctly.

1) Proving the reachability of the Petri Net

- We have already shown the enabled transition after M0; now we will move to find y for all other markings
  - $M1 + I.y1 = M2$

$$\begin{bmatrix}0\\1\\0\\0\\0\\0\\0\\0\\0\end{bmatrix} + \begin{bmatrix}1 & -1 & 0 & 0 & 0 & -1 & 1 & 0 & 0\\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1\\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0\\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1\end{bmatrix} . y = \begin{bmatrix}0\\0\\2\\0\\0\\0\\0\\0\\0\end{bmatrix}$$

Solving for *y2 :*

$$y2 = \begin{bmatrix}0\\0\\1\\0\\0\\0\\0\\0\\0\end{bmatrix}$$

This means that only t2 is enabled.

- $M2 + I.y3 = M3$

$$\begin{bmatrix}0\\0\\2\\0\\0\\0\\0\\0\\0\end{bmatrix} + \begin{bmatrix}1 & -1 & 0 & 0 & 0 & -1 & 1 & 0 & 0\\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1\\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0\\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1\end{bmatrix} . y3 = \begin{bmatrix}0\\0\\0\\0\\1\\0\\0\\0\\0\end{bmatrix}$$

Solving for y3 :

$$y3 = \begin{bmatrix}0\\0\\0\\1\\0\\0\\0\\0\\0\end{bmatrix}$$

This means that only t3 is enabled. Continuing with the same procedure, we get the following:

$$y4 = \begin{bmatrix}0\\0\\0\\0\\1\\0\\0\\0\\0\end{bmatrix}, y5 = \begin{bmatrix}0\\0\\0\\0\\1\\0\\0\\0\\0\end{bmatrix}, y6 = \begin{bmatrix}0\\0\\0\\0\\0\\0\\1\\0\\0\end{bmatrix}, y7 = \begin{bmatrix}1\\0\\0\\0\\0\\0\\0\\1\\0\end{bmatrix}, y8 = \begin{bmatrix}1\\0\\0\\0\\0\\0\\0\\0\\0\end{bmatrix}$$

Finding these vectors proved that our system is reachable. And hence, it functions correctly to achieve its goals.

2) Boundedness of Petri net

Another property of Petri net we need to check in our system is whether the system is bounded or unbounded. Boundedness in Petri net means that the number of tokens in each place is finite. If a Petri net is unbounded, at least one sequence of transitions can lead to infinite tokens in one or more places. This can cause problems during the simulation or execution of the Petri net, as it may not be possible to allocate infinite resources to the system.

Therefore, proving the boundedness of a Petri net is essential to ensure the stability and feasibility of the system it models [12].

- **Lemma 5.1 (Finiteness of a set of firing domains) Set of firing domains of a time Petri net is finite [13]**

This Lemma implies that the number of transitions that can fire is finite. It is essential because this implies that the system has a possible number of finite states and ensures it will reach a finite state.

The proof of this Lemma relies on the observation that a time Petri net can only fire a bounded number of transitions in a finite amount of time. Since the time intervals between transitions are finite and bounded, the number of possible firing sequences must also be finite.

Since our system is time, thus, it can fire a finite number of transitions in a finite amount of time, which means that our system is bounded.

However, we can also use the reachability graph to prove the boundedness of the system. This can be done by determining if the maximum number of tokens in each reachable marking is finite. And since we have already drawn the reachability graph and demonstrated the number of tokens in each marking, which was finite, our system is bounded. Proving the boundness of the Petri net in the context of multi-agent systems helps to ensure the stability and efficiency of the system. It can help to identify potential issues or limitations that may need to be addressed in the design or implementation of the system.

3) Liveness of Petri net

The Petri net with initial marking M0 is live if it is possible to eventually fire any transition by progressing through some different firing sequence regardless of what marking is reached from M0. In other words, we can say that the Petri net is live if its transitions are live. A live transition is a transition that is always enabled from any reachable marking [14]. Again, we can use the reachability graph to prove the liveness of the systems' transitions. So, from fig. 4, which represents the reachability graph, we notice that all transitions are live since they are permanently enabled from the reachable markings. A live Petri net guarantees deadlock-free operation, regardless of the chosen firing sequence.

In a MAS, liveness analysis determines whether the system will eventually reach a desired state or configuration, assuming all agents behave as expected.

In summary, Reachability, Liveness, and boundedness are vital properties that can be analyzed in a Multi-Agent System or a Petri Net to determine the system's behavior and ensure it behaves as desired.

Agent	State 1	Action	State 2	Action 2
Counting agent (c-agent)	1. Have an initial building list 2. Have the initial number of buildings & resident	1. Exchange building list with sorting agent 2. Exchange the number of building with the counting agent	1. Have old list of existing building 2. Have old number of buildings and people	Collect additional information from sensors
Sorting agent (s-agent)	Initial condition of this agent: does not have an initial categorized list of the existing building in a country	1. Request the existing list from the counting agent 2. Classify the buildings in different categories	Have a new list of the existing building in some country	Categorize list
Data collection agent (d-agent)	Initial condition: without the number of building in the specified country	1. Request the number of building from the counting agent 2. Calculate the amount of energy consumption of that country	Calculated amount of energy consumption in some country	Set the state of the country to calculated
City consumption agent (city-agent)	Initial condition: does not have the amount of consumption in a city	1. Request the amount of consumed energy from the data collection agents in all countries in this city 2. Calculate the amount of energy consumed in this city 3. Categorize the over-consumed energy 4. Send notification to interested parties	Calculated amount of energy of the whole city Have an energy consumption history of the city	1. Decrease the amount of energy for a specific country 2. Require options for alternative energy for excessive but necessary energy usage
Monitoring agent (m-agent)	Initial state: no electrical faults detected	1. Receive information from sensors 2. identify the building with electrical faults 3. Cutoff electricity 4. Classify type of faults 5. Exchange with emergency agent	1. Electrical faults detected in a specific building 2. Red flag/ electrical faults history	1. Continuous observation 2. Detection of building with history
Emergency agent (e-agent)	Initial state: no electrical faults need intervention	Receive information from monitoring agent	Emergency detection require interruption	1. Specify type of interruption 2. Send alarm to interested emergency agents from other entities

Table 2: States and actions of the system

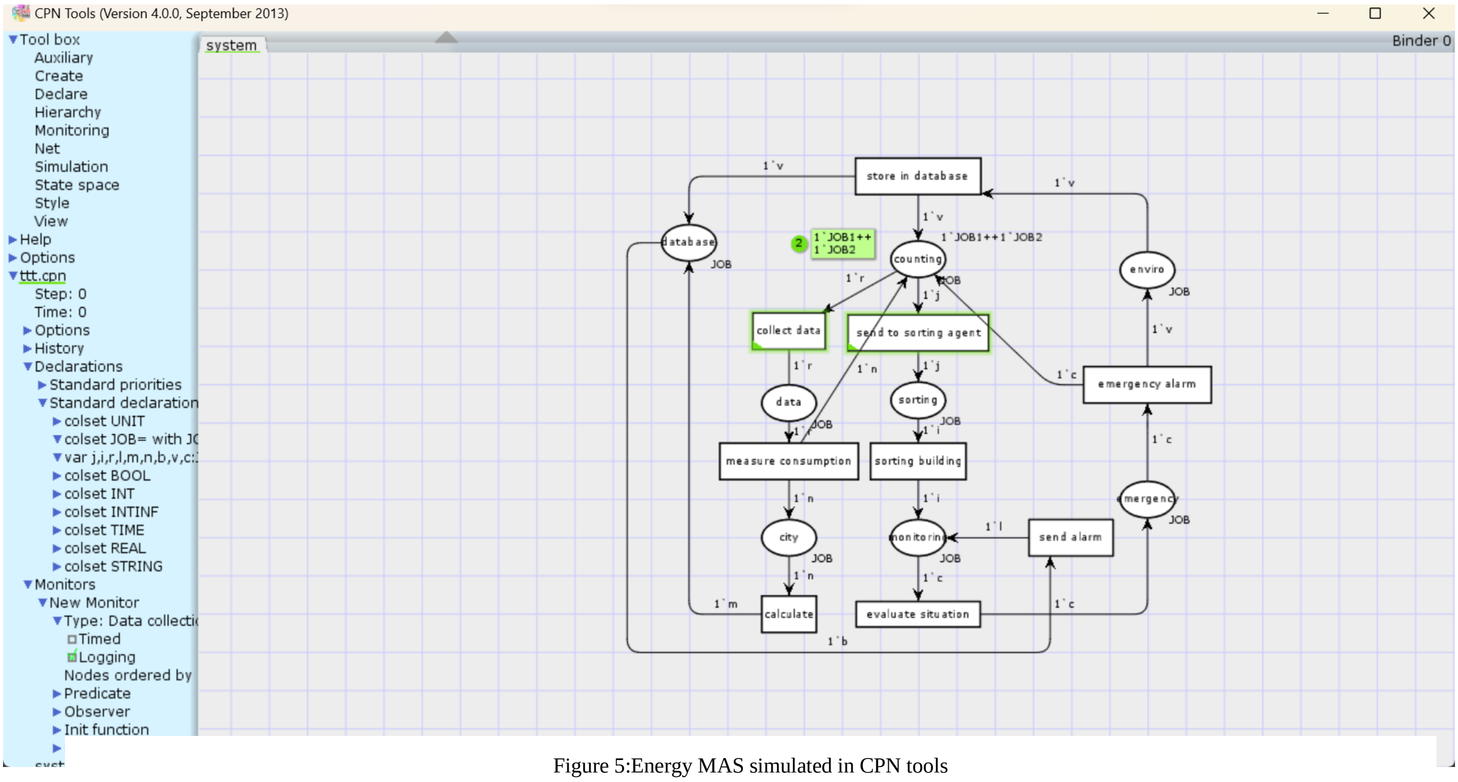

Figure 5:Energy MAS simulated in CPN tools

In this section, we have mathematically proved that our system functions correctly by illustrating the system's reachability, liveness and boundedness proprieties. In the next section, we will demonstrate the simulation results, which almost match the results from this section.

VII. SIMULATION USING CPN TOOLS SOFTWARE

Since we used Petri nets to model our system mathematically, we decided that the best way to simulate the system was by using a software called CPN tools. It is a software dedicated to simulating colored Petri nets. Given that the agents in our MAS have different types and values of data, it was appropriate to model the system using a colored Petri net. First, we would like to identify the agents states and actions in a table to make it easier to understand the simulation diagram.

Table 2 demonstrates the states of each agent and the actions taken by these agents at each state. This table also shows what possible interactions between different agents might occur. The simulation of the energy MAS in CPN tools follows the table. It is important to notice that the simulated Petri net is not the complete system model since we mentioned earlier that the whole city management system is very complex.
Hence, we only simulated the energy MAS.
The simulation results that we got from the software are in fig. 6:

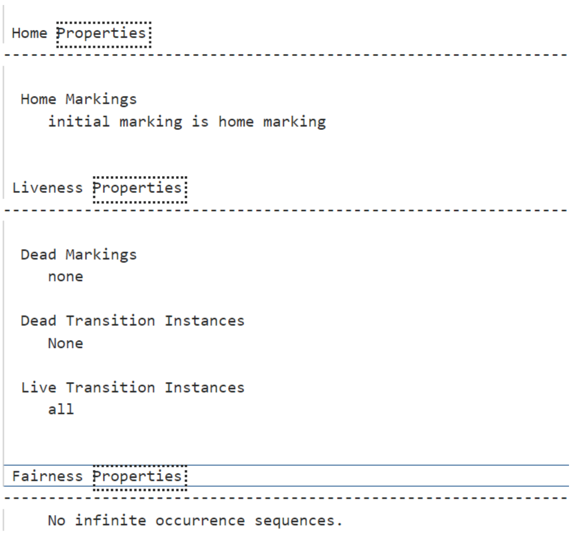

Figure 6: Simulation results

A) Results and discussion

After performing the simulation multiple times in CPN tools (about 2000 iterations), we got the simulation results demonstrated in fig. 6. These results ensure what we got in the mathematical modelling section. i.e., that the result clearly shows that:

- Our system is reachable, which means that all markings can be reached.
- There are no dead transitions, which means that all transitions are enabled (they are all live)
- Our system has no deadlocks, meaning all agents can make further progress as long as the resources are available.
- This means that our system is live
- In the context of a multi-agent system, the simulated system can achieve its goals and behave as it desires.

c) Results in the context of multi-agent system and energy distribution

In the preceding section, we demonstrated that our simulation results confirmed the effectiveness of our multi-agent system in achieving its goals. However, in this subsection, we aim to showcase the significant impact of our system on the overall objective of creating a fair energy distribution system. This can be succinctly summarized by the following key points:

i. Our system enables continuous collection of data regarding the growth of buildings in each country. This translates into a scalable energy distribution that is aligned with the increasing number of individuals and buildings. Additionally, energy consumption is carefully monitored to prevent excessive usage and investigate any unexplained consumption.
ii. Our system features sorting facilities that categorize different energy consumers, thus mitigating the risk of energy cut-off for crucial facilities like hospitals and schools.
iii. Our system provides constant monitoring of electrical faults, thereby avoiding critical situations and expediting emergency response, which ultimately leads to reduced human and material losses.

Overall, our multi-agent system for fair energy distribution proves to be a robust and dynamic solution that addresses the complexities and challenges of the modern energy landscape

VIII. COMMUNICATION BETWEEN AGENTS

In previous sections, we identified the agents in our energy MAS, their behaviours, and their actions under various conditions. The simulation shows how these agents interact with each other to achieve a goal or perform an action. However, in this section, we will present the agents' communications and negotiations with each other based on FIPA ACL language.
In energy MAS, agents communicate with each other to achieve a particular goal. Some of them communicate with other agents from other entities. Ultimately, this will help achieve the common goal of the whole system. Each interaction between any two agents that use FIPA ACL standards requires declaring the interaction protocol, the communicative act, and the content language inside each message. Thus, this section will present some examples of messaging between different agents.
Fig. 7 illustrates a message between two agents according to FIPA specifications; the main components are shown in the figure:

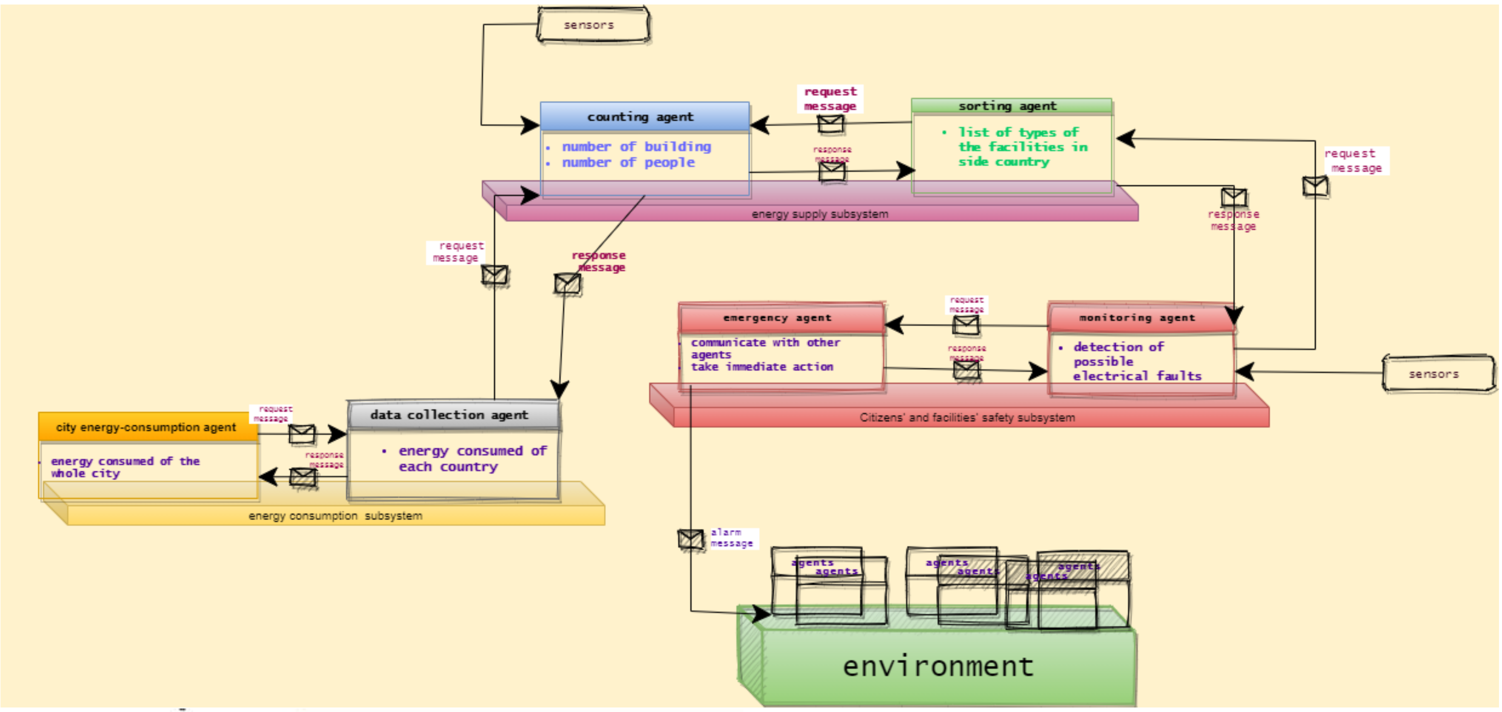

Figure 8: Paradigm of exchanging messages between agents

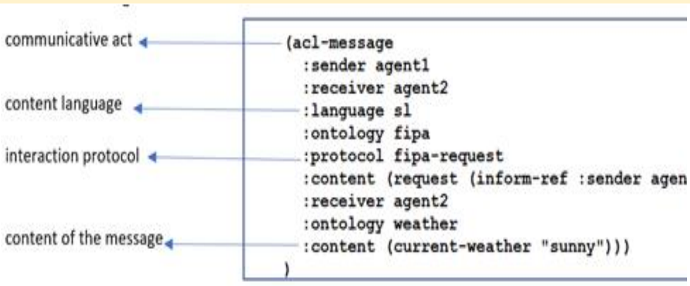

Figure 7:Components of a message between agents according to FIPA specification

In this context, we have constructed examples of messages between three agents in the Energy MAS. Table 3 presents some of these messages.

Additionally, Fig.8 shows a paradigm of exchanging messages between agents in the energy MAS.

The message code in FIPA-ACL	Description of the message
(acl-message :sender agent1 :receiver agent2 :language sl :ontology facility :protocol fipa-request :content (request (action (get-list-of-buildings "country1")) (receiver-goal (sort-facilities))))	The message is using Semantic Language (SL) as the content language and the "facility" ontology, which specifies the common terms and definitions used to represent and interpret the content of the message. The message is using the FIPA-request protocol, which specifies the steps and procedures for making and responding to requests for services or information. The content of the message includes a request action that specifies the action of getting a list of buildings in "country1" and a receiver goal that specifies the purpose of the request, which is to sort the facilities according to their types. Agent2 will receive the message and use the FIPA-request protocol to process and respond to the request based on its capabilities and goals.

(acl-message :sender agent2 :receiver agent1 :language sl :ontology facility :protocol fipa-request :content (agree (request (action (get-list-of-buildings "country1")) (receiver-goal (sort-facilities)))))	The message includes an agree action that indicates that agent2 is accepting the request for a list of buildings in "country1" and agreeing to provide the list to agent1. The content also includes the original request action and receiver goal from the request message, which helps to clarify the context and purpose of the agreement
(acl-message :sender agent1 :receiver agent2 :language ccl :ontology facility :protocol fipa-request :content (request (action (get-list-of-buildings "country1")) (receiver-goal (sort-facilities)) (constraint (number-of-facilities ?n > 10))))	The message is using Constraint Choice Language (CCL) as the content language because we have constraints on the number of facilities that should be on the list. the message includes a request action that specifies the action of getting a list of buildings in "country1" and a receiver goal that specifies the purpose of the request, which is to sort the facilities according to their types. The content also includes a constraint that specifies a minimum number of facilities in the list (greater than 10).

Table 3: Examples of messages between agents

IX. CONCLUSION

The multi-agent system has many applications in every aspect of our life, from AI and robotics to agriculture and ecosystems. In this project, we employed the multi-agent system to design competent city resources management. We devised a system containing multiple entities where each entity has several agents who communicate with each other to achieve the system's common goal, which is to achieve the best resource management in any city. Due to software and time limitations, we could only simulate the energy entity MAS of this complete system. We defined the agents of the energy MAS, their possible behaviour, actions, and the possible communications between them. We used Petri net to model the system mathematically and CPN tools software to simulate the system behaviour in real life. The mathematical model and simulation results proved that the system works correctly to achieve its goals. In the future, we intend to build a whole city management system that compromises all essential entities of all cities.

REFERENCES

[1] Sadigov, R. (2022). Rapid growth of the world population and its socioeconomic results. *The Scientific World Journal*, *2022*, 1–8. https://doi.org/10.1155/2022/8110229

[2] Cruz-Cunha, M. M., Miranda, I. M., & Gonçalves, P. (2013). *Handbook of Research on ICTs and Management Systems for Improving Efficiency in Healthcare and Social Care* (1st ed.). IGI Global.

[3] Conte, G., Morganti, G., Perdon, A. M., & Scaradozzi, D. (2009). Multi-agent system theory for resource management in home automation systems. *Journal of Physical Agents (JoPha)*, *3*(2), 15–19. https://doi.org/10.14198/jopha.2009.3.2.04

[4] Roscia, M., Longo, M., & Lazaroiu, G. C. (2013). Smart City by multi-agent systems. *2013 International Conference on Renewable Energy Research and Applications (ICRERA)*. https://doi.org/10.1109/icrera.2013.6749783

[5] Albino, V., Berardi, U., & Dangelico, R. M. (2015). Smart Cities: Definitions, Dimensions, Performance, and Initiatives. *Journal of Urban Technology*, *22*(1), 3–21. https://doi.org/10.1080/10630732.2014.942092

[6] Crooks, A. T., Patel, A., & Wise, S. (2014). Multi-Agent Systems for Urban Planning. *Advances in Civil and Industrial Engineering*, 29–56. https://doi.org/10.4018/978-1-4666-4349-9.ch003

[7] González-Briones, A., De La Prieta, F., Mohamad, M., Omatu, S., & Corchado, J. (2018). Multi-Agent Systems Applications in Energy Optimization Problems: A State-of-the-Art Review. *Energies*, *11*(8), 1928. https://doi.org/10.3390/en11081928

[8] González-Briones, A., Mezquita, Y., Castellanos-Garzón, J. A., Prieto, J., & Corchado, J. M. (2019). Intelligent multi-agent system for water reduction in automotive irrigation processes. *Procedia Computer Science*, *151*, 971–976. https://doi.org/10.1016/j.procs.2019.04.136

[9] Hamal, O., El Faddouli, N. E., & Alaoui Harouni, M. H. (2021). Design and implementation of the multi-agent system in education. World Journal on Educational Technology: Current Issues, 13(4), 775–793. https://doi.org/10.18844/wjet.v13i4.6264

[10] Sandita, A. V., & Popirlan, C. I. (2015). Developing A Multi-Agent System in JADE for Information Management in Educational Competence Domains. Procedia Economics and Finance, 23, 478–486. https://doi.org/10.1016/s2212-5671(15)00404-9

[11] Petri Nets. (n.d.). *https://www.techfak.uni-bielefeld.de/%7Emchen/BioPNML/Intro/pnfaq.html*

[12] The Petri Net Method By Dr Chris Ling School of Computer Science & Software Engineering Monash University - ppt télécharger. (n.d.). Slideplayer.fr. Retrieved February 21, 2023, from *https://slideplayer.fr/slide/18047033/*

[13] Diaz, M. (2013). *Petri Nets*. John Wiley & Sons.

[14] J.-M. Proth and X. Xie, *Petri nets: a tool for design and management of manufacturing systems*, John Wiley & Sons, 1996

APPENDIX


City management system
Energy supply
Provide energy for all citizens in all countries
Monitor energy consumption
Save from electrical faults
Collect data about the whole city consumption
Water supply
Provide water for all citizens in all countries
Monitor water consumption
Prevent water cutoff
Utilization of alternative water sources
Health sector
Provide health care for all citizen
Ensure that all children had their vaccines
Provide necessary medicine for all citizens
Health facilities
Education sector
Provide education for all citizens in all countries
Monitor the performance of the students
Provide educational supplies for poor students
Educational facilities
Agricultural sector
Provide water for all agricultural lands
Collect crops
Fight against agricultural pest
Improve performance of agricultural lands
Manufacturing sector
Improve all kind of products
Ensure safety of all workers