

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет                      Комп'ютерної інженерії та управління  
Кафедра                        Комп'ютерних інтелектуальних технологій та систем

## **КВАЛІФІКАЦІЙНА РОБОТА**

### **Пояснювальна записка**

рівень вищої освіти                      другий (магістерський)

Мультиагентна система для забезпечення комфорту та безпеки у розумних  
будинках.

Виконав:

студент II курсу, групи КІТм-22-1

Олена ВАСИЛЕВСЬКА

Спеціальність 123 Комп'ютерна інженерія

Тип програми освітньо-професійна

Освітня програма Комп'ютерні

інтелектуальні технології

Керівник проф. каф. КІТС Наталія АКСАК

Допускається до захисту

Зав. кафедри

Олег РУДЕНКО

(підпис)

2023 р.

Харківський національний університет радіоелектроніки

|                     |   |
|---------------------|---|
| Факультет           | Комп'ютерної інженерії та управління              |
| Кафедра             | Комп'ютерних інтелектуальних технологій та систем |
| Рівень вищої освіти | другий (магістерський)                            |
| Спеціальність       | 123 Комп'ютерна інженерія                         |
| Тип програми        | освітньо-професійна                               |
| Освітня програма    | Комп'ютерні інтелектуальні технології             |

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_

(підпис)

« \_\_\_\_\_ » \_\_\_\_\_ 202\_ р.

## ЗАВДАННЯ

### НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові \_\_\_\_\_

Василевській Олені Олексіївні

(прізвище, ім'я, по батькові)

1. Тема роботи Мультиагентна система для забезпечення комфорту та безпеки у розумних будинках.

затверджена наказом по університету від " 03 " листопада 2023 р. № 1290Ст

2. Термін подання студентом роботи до екзаменаційної комісії 13.01.2024

3. Вхідні дані до роботи

Виявлення основних характеристик та особливостей мультиагентних систем,

Побудова тестової моделі системи.

Мультиагентна система

Мова програмування Python

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

1) Огляд предметної області

2) Аналіз об'єкту дослідження

3) Огляд існуючих рішень

4) Обґрунтування мети

5) Вирішення поставленої задачі

6) Програмна реалізація запропонованої моделі

7) Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням кафедри)

Слайд-презентація – 10 слайдів

6. Консультанти розділів роботи и (п.6 включається до завдання за наявністю консультантів згідно до наказу, зазначеному у п.1)

| Найменування розділу | Консультант (посада, прізвище, ім'я, по батькові) | Позначка консультанта про виконання розділу |      |
|----------------------|---|---|------|
|                      |   | підпис                                      | дата |
|                      |   |   |      |
|                      |   |   |      |

### КАЛЕНДАРНИЙ ПЛАН

| № | Назва етапів роботи                               | Термін виконання етапів роботи | Примітка |
|---|---|--------------------------------|----------|
| 1 | Огляд і аналіз сучасного стану поставленої задачі | 06.11.23 - 10.11.23            | Виконано |
| 2 | Постановка задачі кваліфікаційної роботи          | 11.11.23 - 13.11.23            | Виконано |
| 3 | Обґрунтування мети вирішення поставленої задачі   | 14.11.23 - 20.11.23            | Виконано |
| 4 | Вибір та розробка моделі взаємодії агентів        | 21.11.23 - 18.12.23            | Виконано |
| 5 | Оформлення матеріалів кваліфікаційної роботи      | 19.12.23 - 08.01.24            | Виконано |
| 6 | Розробка презентації                              | 09.01.24 - 13.01.24            | Виконано |
| 7 | Подання кваліфікаційної роботи на рецензування    | 16.01.2024                     | Виконано |
| 8 | Попередній захист                                 | 22.01.2024                     | Виконано |
| 9 | Захист роботи                                     | 26.01.2024                     | Виконано |

Дата видачі завдання 06 листопада 2023 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_  
(підпис) \_\_\_\_\_ (посада, ім'я, прізвище)

## РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 80 с., 16 рис., 1 табл., 2 дод., 64 джерела.

### ІоТ, ІНТЕРНЕТ РЕЧЕЙ, РОЗУМНИЙ БУДИНОК, МУЛЬТИАГЕНТНА СИСТЕМА, ХМАРНІ ОБЧИСЛЕННЯ

Метою даної роботи є розробка архітектури мультиагентної системи для розумного будинку та визначення ролі кожного агента для забезпечення комфорту та безпеки, зокрема визначивши їх функціональні обов'язки та взаємодію в системі.

У роботі пропонується підхід до інтеграції мультиагентних систем в розумні будинки та міста, який орієнтований на вдосконалення управління, безпеки та ефективності. Також увага акцентується на важливості синергії між різними технологіями ІоТ для створення більш інтелектуального та взаємопов'язаного життєвого середовища.

Об'єктом дослідження є мультиагентна система для забезпечення комфорту та безпеки в розумному будинку .

Предметом дослідження є процес аналізу та розробки мультиагентної системи для забезпечення комфорту та безпеки в розумному будинку.

## ABSTRACT

Master's thesis: 80 pages, 16 figures, 1 table, 2 appendices, 64 sources

IoT, INTERNET OF THINGS, SMART HOME, MULTI-AGENT SYSTEM,  
CLOUD COMPUTING

The purpose of the qualification work is developing the architecture of a multi-agent system for a smart home and defining the role of each agent to ensure comfort and safety, in particular by defining their functional responsibilities and interaction in the system.

The work proposes an approach to the integration of multi-agent systems in smart homes and cities, which is focused on improving management, security, and efficiency. Also, attention is focused on the importance of synergy between different IoT technologies to create a more intelligent and interconnected living environment.

The object of research is a multi-agent system for ensuring comfort and safety in a smart home.

The subject of the study is the process of analysis and development of a multi-agent system to ensure comfort and safety in a smart home.



## АНОТАЦІЯ

Василевська Олена Олексіївна. Мультиагентна система для забезпечення комфорту та безпеки у розумних будинках – Магістерська кваліфікаційна робота.

Актуальність теми дослідження.

У світі, де технології стрімко розвиваються, володіння розумним домом перестає бути розкішшю та перетворюється на необхідність для сучасного життя. Розумний дім відіграє ключову роль у підвищенні якості життя, забезпечуючи не тільки комфорт і зручність, але й безпеку, енергоефективність та екологічну сталість. У контексті глобальних викликів, таких як зміна клімату і необхідність зниження вуглецевого сліду, розумні дома стають важливим інструментом в досягненні цих цілей. Вони дозволяють інтелектуально керувати ресурсами, знижувати витрати та покращувати енергетичну ефективність. Крім того, враховуючи зростання чисельності населення та урбанізацію, розумні дома сприяють створенню більш ефективних та комфортних міських середовищ. Вони допомагають зменшити навантаження на міські інфраструктури та поліпшити загальну якість життя мешканців.

Окрім екологічного аспекту, розумний дім відіграє важливу роль у сфері охорони здоров'я та добробуту. Системи моніторингу якості повітря, води, а також здатність відстежувати і аналізувати поведінку мешканців, можуть сприяти підтримці здорового способу життя та навіть попереджати потенційні проблеми зі здоров'ям.

В контексті постійно змінюваного світу, де гнучкість та адаптивність стають ключовими, розробка мультиагентної системи для створення розумного дому надає невід'ємну перевагу. Мультиагентна система гнучка, автономна та ефективна: в ній управління розподіляється між кількома агентами, що підвищує надійність системи. Навіть якщо один агент виходить з ладу, інші можуть продовжувати виконувати свої функції. Такі системи легко

масштабуються, оскільки нові агенти можуть бути додані з мінімальними змінами в загальній структурі. Це робить мультиагентні системи ідеальними для розумних домів, де потреби та вимоги можуть змінюватися з часом. Агенти можуть обмінюватися інформацією та вчитися один від одного, покращуючи загальну продуктивність системи. В розумному домі це може означати більш ефективне управління енергією, безпекою та іншими системами.

Метою даної роботи є розробка архітектури мультиагентної системи для розумного будинку та визначення ролі кожного агента для забезпечення комфорту та безпеки, зокрема визначивши їх функціональні обов'язки та взаємодію в системі.

Об'єктом дослідження є мультиагентна система для забезпечення комфорту та безпеки в розумному будинку.

Предметом дослідження є процес аналізу та розробки мультиагентної системи для забезпечення комфорту та безпеки в розумному будинку.

Методи дослідження. Дослідження базується на системному аналізі результатів сучасних теоретичних і прикладних розробок вітчизняних і зарубіжних вчених в галузі ІТ. Для вирішення поставлених завдань використано: методи системного аналізу, мультиагентна технологія, методи побудови інтелектуальних обчислювальних систем, методи об'єктно-орієнтованого програмування, методи побудови програмних додатків з командним інтерфейсом

Практична цінність отриманих результатів. Практична цінність отриманих результатів у цій роботі полягає в застосуванні досліджених мультиагентних систем у реальних сценаріях розумних будинків. Результати можуть бути використані для створення більш ефективних, гнучких та інтелектуальних рішень у сфері автоматизації домогосподарства. Це може включати покращення систем управління енергією, оптимізацію систем опалення, освітлення, вентиляції, а також забезпечення вищого рівня безпеки. Крім того, отримані знання та розроблені методології можуть бути застосовані у проектах розумних міст, де потреба в інтеграції різних систем і управління

великими даними є ключовою. На практичному рівні, це дослідження може бути корисним для інженерів, дизайнерів і розробників, які працюють у галузі інтелектуальних технологій та автоматизації дому. Вони зможуть використовувати результати для покращення існуючих систем або створення нових інноваційних рішень, які будуть більш адаптивними та відповідатимуть специфічним потребам користувачів.

У першому розділі зроблено аналіз предметної області: розглянуто значення розумного дому як цілісної системи, яка інтегрує різні пристрої та системи для забезпечення комфорту та безпеки мешканців, його важливість у сучасному світі та економічна цінність цих систем. Розглянуто можливість створення мультиагентних систем для домашньої автоматизації, їх переваги та здатності інтегруватися з існуючими пристроями, внесок у безпеку та зручність користувачів. Розглядаються можливості MAC у контексті контролю доступу, виявлення пожеж, відеоспостереження, догляду за домашніми тваринами, інтеграції з іншими інформаційними технологіями, які сприяють більш точному виявленню потенційних загроз. Мультиагентні системи виявляються ефективними у створенні єдиної екосистеми, що об'єднує різні інтелектуальні пристрої і робить розумний будинок модульною, гнучкою системою, що дозволяє адаптувати системи до змінних потреб та умов.

У другому розділі були досліджені сучасні виклики та проблеми, пов'язані з розумними будинками, одним з яких є важливість забезпечення високого рівня користувацького досвіду і зручності в управлінні, що вимагає продуманого дизайну інтерфейсів та взаємодії з користувачем. У той час як автоматизація завдань у розумних будинках може бути досягнута за допомогою розумних пристроїв (наприклад, датчиків та ламп), застосунків і платформ, таких як IFTTT або SmartThings, розумних хабів, що об'єднують різні пристрої у єдину систему, та геолокації, яка автоматизує дії залежно від місцезнаходження користувача, існують проблеми у створенні розумних будинків, зокрема фрагментація і складність управління системами від різних виробників. Відсутність єдиної екосистеми або операційної системи для дому

ускладнює управління розумними будинками, часто вимагаючи від власників окремих паролів та ручного налаштування. За допомогою мультиагентних систем можна розробити архітектуру, яка підтримує ці вимоги, забезпечуючи гнучкість, адаптивність та інтелектуальну взаємодію між різними компонентами системи.

У третьому розділі пропонується динаміка мультиагентних систем у контексті розумних міст, включаючи аспекти інтеграції IoT у міському середовищі, виклики управління розподіленими системами та ефективне використання інтелектуальних пристроїв у контексті середовища. Технології розподіленої обробки даних та агентно-орієнтоване програмування вважаються ефективними для впровадження інтелектуальних систем у містах. Вони забезпечують самоадаптацію та самоорганізацію, сприяючи кращому збору та обробці даних. Ключовим аспектом є інформаційний обмін між агентами для координації та ефективності системи, що дозволяє адаптуватися до неочікуваних змін або збоїв, навчаючись на попередніх досвідах і постійно оновлюючи стратегії дій. Самоадаптивні мультиагентні системи безперервно аналізують зворотний зв'язок з внутрішніх і зовнішніх джерел, підтримуючи постійний обмін інформацією, через що мають змогу на ефективну реакцію на непередбачувані ситуації в середовищі. Однак обсяг зберігаємої інформації, кількість непостійних з'єднань та велика розмірність даних, які збираються через MAC, ставлять перед системою виклики у вигляді обмежень ресурсів для передачі, обробки та зберігання даних, у чому в нагоді стають методи подієвого відбору даних, розподіленої обробки, фільтрації інформації, абстрагування даних. Використання технік глибокого навчання, таких як LSTM та варіаційні автокодери, дозволяє забезпечити динамічну адаптацію MAC до змін у середовищі, а також впровадження ефективних заходів для виявлення та реагування на потенційні аномалії.

Четвертий розділ присвячений моделюванню мультиагентних систем у контексті розумних будинків. У цьому розділі акцент зроблено на розробці та аналізі різних моделей поведінки агентів, їх взаємодії з компонентами системи, а також на оцінці ефективності цих систем. Описано різні агенти, що

є частиною системи, кожен з яких має свої функції:

Агент повідомлень - забезпечує збір та розповсюдження інформації.

Агент керування пристроями - обробляє дані з сенсорів та керує підключеними пристроями.

Агент камери - спостерігає за середовищем та аналізує відео.

Агент сигналізації - активує систему безпеки при підтвердженні подій.

Агент годування - відповідає за автоматичне годування тварин.

Агент контролю клімату - регулює температуру, вологість та освітлення.

Агент медичного догляду - відстежує здоров'я користувачів та тварин.

Агенти взаємодіють між собою та з середовищем, яке включає сенсори, IoT пристрої, голосові та сенсорні інтерфейси, центральні сервери та системи безпеки. Використовуються моделі Крипке та темпоральна логіка для аналізу динаміки системи та взаємодії між агентами. Також зроблено програмну реалізацію взаємодії між агентами, включаючи розробку класів для кожного агента та визначення їх взаємодії через повідомлення та переходи станів.

## IoT, ІНТЕРНЕТ РЕЧЕЙ, РОЗУМНИЙ БУДИНОК, МУЛЬТИАГЕНТНА СИСТЕМА, ХМАРНІ ОБЧИСЛЕННЯ

Публікації здобувача за темою роботи:

1. Василевська О.О. Огляд можливостей мультиагентної системи для забезпечення комфорту та безпеки у розумних будинках. Науковий простір: аналіз, сучасний стан, тренди та перспективи: матеріали III Всеукраїнської студентської наукової конференції, м.Київ, 16 червня, 2023 рік /ГО «Молодіжна наукова ліга».— Вінниця: ГО «Європейська наукова платформа», 2023.с. 128 –131. ISBN 978-617-8126-55-1. DOI 10.36074/liga-ukr-16.06.2023.

2. Василевська О.О. Методи кодування зображень і відео для наднизької затримки. Науковий простір: аналіз, сучасний стан, тренди та перспективи: матеріали III Всеукраїнської студентської наукової конференції, м.Київ, 16 червня, 2023 рік /ГО «Молодіжна наукова ліга».— Вінниця: ГО

«Європейська наукова платформа», 2023.с. 125 –127. ISBN 978-617-8126-55-1.  
DOI 10.36074/liga-ukr-16.06.2023.

3. Василевська О.О. Архітектура мультиагентної системи для забезпечення комфорту та безпеки у розумних будинках. *VI International scientific and practical conference «The aspects of contemporary scientific research that encompass both theoretical and practical components»*(January 10-12, 2024) Venice, Italy, International Scientific Unity. 2024. С. 49 – 55.

Список праць наукового керівника за темою магістерської роботи:

1.N . Axak, M. Kushnaryov, A.Tatarnykov, The Agent-Based Learning Platform. ICST-2023: XI International Scientific and Practical Conference “Information Control Systems and Technologies”, September 21-23, 2023, Odessa, Ukraine, Vol-3513, pp. 263-275. ONLINE: <http://ceur-ws.org/Vol-3513/>, URN: urn:nbn:de:0074-3513-6.

2.N. Axak A. Tatarnykov, "The behavior model of the computer user," 2022 IEEE 17th international conference on computer sciences and information technologies (csit), 2022, pp. 458-461, doi: 10.1109/CSIT56902.2022.10000499.

3.Axak N., Korablyov M., Ushakov M. Cloud Architecture for Remote Medical Monitoring // IEEE Proceedings of the 15th International conference «Computer Sciences and Information Technologies», (CSIT-2020). – Zbarazh-Lviv, Ukraine, September 23-26, 2020. – vol. 1, pp. 344-347.

4.Natalia Axak, Mykola Korablyov, Matvii Ushakov. The Development of a Multi-Agent System for Controlling an Autonomous Robot // Proceedings of the 17th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Volume I, (ICTERI-2021). – Kherson, Ukraine, 2021. – Vol-3013, pp. 96-105. <http://ceur-ws.org>. ISSN 1613-0073

# ЗМІСТ

|   |    |
|---|----|
| Вступ   | 16 |
| 1 Аналіз предметної області та постановка задачі  | 17 |
| 1.1 Аналіз предметної області   | 17 |
| 1.2 Мета та завдання роботи   | 21 |
| 2 Концепція розумного будинку   | 22 |
| 2.1 Визначення «розумного дому»   | 22 |
| 2.2 Сучасні виклики та проблеми   | 22 |
| 2.3 Огляд існуючих рішень і систем  | 23 |
| 2.4 Автоматизація завдань у розумних будинках   | 24 |
| 2.5 Проблеми створення розумних будинків  | 25 |
| 3 Розумне місто   | 28 |
| 3.1 Динаміка мультиагентних систем  | 28 |
| 3.2 Навколишнє середовище   | 31 |
| 3.3 Інформаційний потік між агентами  | 33 |
| 3.4 Парадигми моніторингу   | 35 |
| 4 Застосування агентів у розумних будинках  | 41 |
| 4.1 Мультиагентні системи   | 41 |
| 4.2 Використання багатоагентних систем для розумного дому                                     | 42 |
| 4.3 Архітектура МАС для забезпечення комфорту та безпеки                                      | 45 |
| 4.3.1 Властивості агента повідомлень  | 53 |
| 4.3.2 Властивості агента керування пристроями   | 56 |
| 4.3.3 Властивості агента камери   | 58 |
| 4.3.4 Властивості агента сигналізації   | 60 |
| 4.3.5 Властивості агента годування  | 62 |
| 4.3.6 Властивості агента контролю клімату   | 64 |
| 4.3.7 Властивості агента медичного догляду  | 65 |
| 4.4 Приклад реалізації мультиагентної системи для забезпечення комфорту та безпеки у розумних | 67 |

|  |                                  |
|--|----------------------------------|
| Висновки                               | 71                               |
| Перелік джерел посилання               | 72                               |
| Додаток А                              | <b>Помилка!</b>                  |
| <b>Закладку не визначено.</b>          |                                  |
| Графічний матеріал                     | кваліфікаційної роботи           |
| <b>Помилка! Закладку не визначено.</b> |                                  |
| Додаток Б                              | Сертифікати учасниці конференцій |
| <b>Помилка! Закладку не визначено.</b> |                                  |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,  
СКОРОЧЕНЬ І ТЕРМІНІВ

AI – Artificial Intelligence

DSM – Demand-side management

FIPA – Foundation for Intelligent Physical Agents

GAE – Graph Autoencoders

GCN – Graph Convolutional Networks

GNN – Graph Neural Networks

GRNN – Graph Recurrent Neural Networks

GUI – Graphical User Interface

ML – Machine Learning

IoT – Internet of Things

IFTTT – If This, Then That

SL – Semantic Language

HVAC – Heating, ventilation, and air conditioning

MAC – Мультиагентна система

## ВСТУП

У сучасному світі розумні будинки є неабияким трендом у галузі технологій та інновацій. Завдяки постійному розвитку інтернету речей (IoT), автоматизації та штучному інтелекту, розумні будинки стають все більш доступними та популярними. Хоча задача створення ідеального розумного будинку, який би повністю задовольняв потреби мешканців, все ще залишається актуальною, вона може бути досягнута за допомогою мультиагентних систем. Використання мультиагентних систем є обнадійливим підходом для підвищення функціональності та ефективності розумних будинків. Ці системи, які складаються з незалежних агентів, що взаємодіють між собою, можуть відкрити нові можливості для ефективного керування різними аспектами побуту та забезпечення безпеки. Існує великий потенціал для подальшого вдосконалення цих систем з метою забезпечення максимального комфорту та безпеки для мешканців.

Ця робота присвячена дослідженню можливостей використання мультиагентної системи для досягнення цих цілей у розумних будинках. Мультиагентні системи - це інтелектуальні системи, що складаються з незалежних агентів, які взаємодіють між собою для досягнення спільних цілей. Акцентується увага на здатності таких систем до модульності, гнучкості та інтеграції з різними технологіями, що відкриває широкі перспективи для автоматизації та підвищення якості життя. Мультиагентні системи в розумних будинках можуть впроваджувати інноваційні рішення для керування різними аспектами побуту, від опалення та освітлення до безпеки та управління енергоресурсами. Використання мультиагентних систем у розумних будинках може відкрити нові можливості для ефективного керування різними аспектами побуту та забезпеченням безпеки, шляхом координації та співпраці агентів.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

## 1.1 Аналіз предметної області

Розумний дім є одним із прикладів Інтернету речей (IoT), але у той же час, IoT має більш широкий спектр використання, який не обмежується лише розумними будинками. IoT включає в себе різноманітні пристрої та системи, які підключені до Інтернету і можуть збирати, обмінюватися даними та взаємодіяти в широкому спектрі галузей, включаючи промисловість, медицину, транспорт, а також особисте використання в побуті, як-от у розумних будинках. Межа між IoT та розумним будинком полягає в масштабі та інтеграції систем. Чайник, який можна вмикати зі смартфона, є прикладом IoT, адже він підключений до Інтернету та має функції дистанційного управління. Проте, щоб вважатися частиною розумного будинку, цей чайник має бути інтегрований у ширшу систему домашнього автоматизування, яка може включати управління освітленням, температурою, безпекою та іншими побутовими системами. Розумний дім - це цілісне рішення, що охоплює кілька пристроїв та систем, об'єднаних спільною платформою керування. Розумний дім використовує підключені пристрої, які збирають інформацію з різних джерел, таких як датчики температури, вологості, руху та енергії, і аналізує ці дані для прийняття автоматичних рішень та керування різними системами будинку, надаючи користувачам зручність, ефективність та безпеку у їх повсякденному житті. Це сприяє створенню інтелектуального житлового середовища, де будинки стають не просто місцем проживання, але і активними учасниками мережі, співпрацюючи з іншими пристроями та системами, щоб забезпечувати комфорт, енергоефективність та безпеку для своїх мешканців.

З розвитком технологій і штучного інтелекту, мультиагентні системи (МАС) стають все більш популярними у сфері домашньої автоматизації. Ці системи представляють собою інноваційний підхід, що поєднує у собі розумні

алгоритми, датчики та комунікаційні протоколи, щоб забезпечити безпеку і зручність для користувачів. МАС інтегруються з існуючими розумними пристроями та системами, створюючи єдину екосистему, яка може працювати разом як для забезпечення безпеки у розумних будинках, так і для догляду за домашніми улюбленцями або іншими членами родини хто потребує догляду.

Мультиагентна система може бути використана для забезпечення комфорту та безпеки у розумних будинках, надаючи ряд можливостей і переваг. Незалежно від того, чи це контроль доступу, виявлення пожежі або небажаної активності в приміщенні, МАС працює в реальному часі для оперативного реагування на потенційні небезпеки. Їх здатність до автономного аналізу ситуацій та взаємодії з іншими пристроями дозволяє створювати більш динамічні та ефективні системи безпеки. Якщо одна з цих подій не буде помічена і не будуть вжиті відповідні механізми, організація або володарі будинку можуть зазнати серйозних збитків і фінансових втрат [1]. Також, мультиагентні системи можуть ефективно інтегруватися з іншими інформаційними технологіями, такими як хмарні обчислення та великі дані, для збору та аналізу інформації з різних джерел, що забезпечує більш точне та своєчасне виявлення потенційних загроз. Одним із ключових аспектів МАС є їх здатність до модульності та гнучкості. Це означає, що системи можуть бути легко налаштовані або розширені залежно від зміни потреб або умов, що дозволяє максимально адаптувати систему до специфіки кожного розумного будинку. Дослідження [2] повідомляє про зростання випадків фізичної безпеки в 2023 році порівняно з попереднім роком, і, згідно прогнозування, 91% опитуваних повідомили що кількість інцидентів продовжить зростати. Щодо пожежних небезпек, Всесвітній центр статистики пожеж зафіксував у середньому три з половиною мільйони пожеж і сорок тисяч смертей щороку в період з 1993 по 2020 рік [3]. Ці звіти підкреслюють потребу у кращому та більш надійному захисту. Традиційні рішення дають велику кількість помилкових спрацювань і зазвичай активують тривоги, які вимагають безпосередніх дій від персоналу. Відсутність можливостей автоматичного

підтвердження подій та впровадження протидій може призвести до швидкої ескалації ситуації. Отже, системи безпеки стають невід'ємною частиною розумних будівель [4].

Стандартні системи пожежної сигналізації зазвичай мають обмеження у вигляді можливості роботи лише у внутрішніх приміщеннях, мають багато помилкових спрацьовувань, не можуть надавати актуальну інформацію пожежникам [5] і, що найважливіше, мають значну затримку між початком пожежі та її виявленням [6].

Крім того, МАС можуть виявляти та вирішувати проблеми, пов'язані з комфортом, а саме доглядом за домашніми тваринами. Вони можуть встановлювати розклад годування та години виходу, контролювати рівень води, її чистоту та температуру в акваріумі, а також здійснювати візуальний контроль за тваринами за допомогою вбудованих камер. Оскільки багато людей живуть зі своїми домашніми тваринами та не завжди можуть мати можливість знаходитись зі своїми улюбленцями протягом всього дня, тому використання МАС може дозволити власникам тварин спокійно перебувати поза домом, знаючи, що їхні улюбленці отримують необхідний догляд і захист.

Використовуючи МАС, власники домашніх тварин можуть віддалено відстежувати активність і місцезнаходження своїх улюбленців, стежити за станом їх здоров'я або навіть взаємодіяти з ними [7]. Розумні речі для догляду за домашніми тваринами відіграють незамінну роль у повсякденному житті власників домашніх тварин. Саме для облегшення догляду можна застосувати технологію IoT для впровадження інтегрованої системи, яка включатиме годівницю для корму, диспенсер для води та заміна наповнювача туалету домашньої тварини, які є трьома найважливішими елементами, про які власники домашніх тварин будуть турбуватися [8]. МАС здатна за допомогою різних датчиків може контролювати і підтримувати оптимальні умови для домашніх тварин: регулювати температуру, вологість та освітлення у відповідності до потреб конкретного виду тварини. Наприклад, система може

увімкнути або вимкнути опалення в залежності від температури приміщення або вологості для забезпечення комфорту тварини [9].

Також МАС може надати власникам можливість візуального спостереження за своїми тваринами. Завдяки вбудованим камерам і зв'язку з мобільними пристроями, власники можуть віддалено спостерігати за тваринами, перевіряти їхню активність та стан здоров'я. Це дозволяє вчасно виявляти будь-які незвичайні або небезпечні ситуації та приймати відповідні заходи [10]. МАС здатна вирішувати потреби різних видів домашніх тварин, починаючи від собак і котів до екзотичних птахів, рептилій чи акваріумних рибок.

Потенційна цінність Інтернету речей велика і зростає. За оцінками [11], до 2030 року в усьому світі вона може досягти 12,5 трильйонів доларів. Це включає цінність, отриману споживачами та замовниками продуктів і послуг IoT. Потенційна економічна цінність IoT залежить від налаштувань і способів використання, причому заводські налаштування та програми для здоров'я людини становлять величезну частку цієї суми. Заводські налаштування можуть принести від 1,4 трлн до 3,3 трлн дол. США до 2030 року, або трохи більше чверті від загальної потенційної вартості. Економічний вплив IoT на здоров'я людини може досягти приблизно 14 відсотків від загальної оціночної вартості. Прогнозується, що між 2023 і 2027 роками світовий дохід на ринку розумних будинків постійно зростатиме на 83 600 мільйонів доларів США (+60,01 відсотка). Після четвертого року поспіль зростання доходу, за оцінками, досягне 222 901,45 мільйонів доларів США, а отже, нового піку в 2027 році. Примітно, що дохід ринку розумних будинків постійно зростає протягом останніх років.

Цей аналіз дозволяє отримати глибше розуміння розумних будинків та потенціалу мультиагентних систем для забезпечення комфорту та безпеки.

Розумні будинки та мультиагентні системи стикаються з деякими викликами, такими як складність інтеграції різних пристроїв та систем, захист

приватності та безпеки даних, а також необхідність ефективного управління конфліктами між агентами.

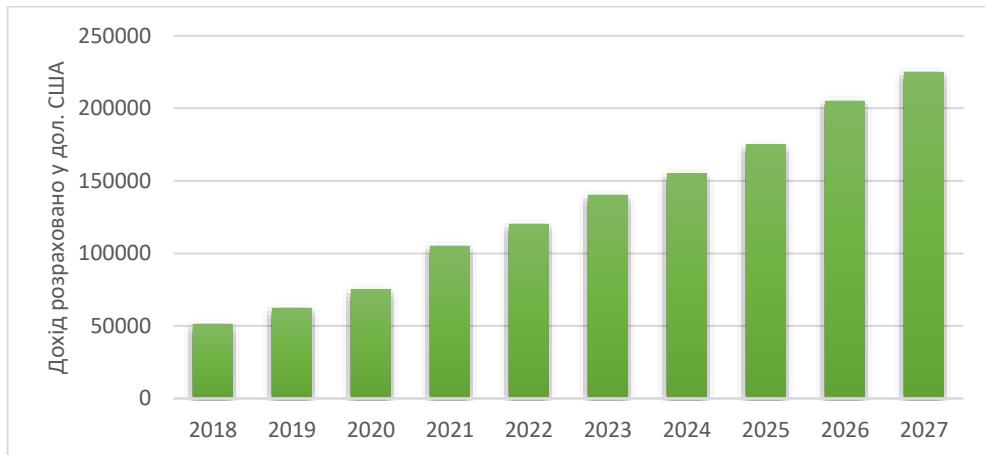


Рисунок 1.1 – Дохід ринку розумних будинків у всьому світі 2018 -2027 [12]

Проте, використання машинного навчання, аналізу даних та розширених інтерфейсів може покращити функціональність та ефективність мультиагентних систем у розумних будинках.

## 1.2 Мета та завдання роботи

Розробити архітектуру мультиагентної системи для розумного будинку та визначити ролі кожного агента з метою забезпечення комфорту та безпеки, зокрема визначивши їх функціональні обов'язки та взаємодію в системі.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- проаналізувати існуючі підходи для побудови розумного будинку з використанням мультиагентних технологій;
- обґрунтувати мову програмування для вирішення поставленої задачі;
- розробити прототип мультиагентної системи для розумного будинку.

Такий підхід дозволяє сфокусуватися на проектуванні архітектури та конкретних ролях агентів, забезпечуючи чітку орієнтацію на створення ефективної та легко розширюваної мультиагентної системи для розумного будинку.

## 2 КОНЦЕПЦІЯ РОЗУМНОГО БУДИНКУ

### 2.1 Визначення «розумного дому»

Для деяких розумний будинок переповнений гаджетами, які автоматизують або контролюють майже всі мислимі частини будинку. Для інших «розумний будинок» — це надзвичайна простота з найменшими технологічними втручаннями, яких вимагає необхідність. Сутність цих інноваційних будівельних проектів полягає у використанні технологій, які дозволяють автоматизувати різноманітні завдання і оптимізувати повсякденне життя мешканців. Від керування освітленням і клімат-контролем до безпеки і розваг — автоматизація є головним компонентом розумного будинку. Сучасні автоматизовані будинки мають цілу екосистему підключених пристроїв, які можуть допомогти користувачам мати найкращу безпеку та зручність у повсякденному житті. Багато власників розумних будинків також інтегрують робототехніку у своє повсякденне життя, щоб автоматизувати повсякденні завдання. Використання розумних домів стає звичайністю через зростаючу популярність та доступність технологій IoT, а також через бажання людей підвищити зручність, безпеку та енергоефективність своїх будинків. Постійний розвиток технологій та підвищення усвідомленості про екологічні та енергетичні переваги сприяють популяризації концепції розумних домів.

### 2.2 Сучасні виклики та проблеми

Як і перед будь якою іншою розробкою, існують деякі проблеми, пов'язані з використанням мультиагентних систем у розумних будинках. Однією з головних проблем є інтеграція різноманітних систем та пристроїв в єдину ефективну мережу. Це включає в себе виклики, пов'язані зі сумісністю та стандартизацією. Інша проблема полягає у забезпеченні безпеки та конфіденційності даних у таких системах. Ще одним аспектом є питання

енергоефективності та розумного управління ресурсами, яке є ключовим для сталого розвитку розумних будинків. Крім того, існують технічні виклики, пов'язані з розвитком та підтримкою складного програмного забезпечення, яке використовується в мультиагентних системах. Важливо також забезпечити високий рівень користувацького досвіду і зручності в управлінні розумними будинками, що вимагає продуманого дизайну інтерфейсів та взаємодії з користувачем. Усі ці аспекти вимагають уваги та розвитку, щоб мультиагентні системи могли вдосконалюватись та ефективно виконувати свої функції у розумних будинках.

### 2.3 Огляд існуючих рішень і систем

Як складна парадигма, МАС має онтології, які дозволяють нам формально визначати типи, властивості та зв'язки між сутностями, які фактично або фундаментально існують для домену зокрема. Потрібна чітко визначена онтологія на додаток до визначення, зробленого FIPA-Directory Facilitator (FIPA-DL). Кілька специфікацій з'явилися в онтологіях; однак жоден не отримав широкого визнання. Мова веб-онтології (OWL) є одним із найпоширеніших стандартів у МАС.

Необхідність використання онтологій зумовлена властивою складністю додатків, розроблених у контексті багатоагентних систем, що викликає наступні труднощі: велика кількість комунікацій між агентами; сумісність систем і платформ і семантичні проблеми.

Щоб агенти могли спілкуватися один з одним, вони повинні використовувати однакову мову, словниковий запас і протоколи. Дотримуючись рекомендацій стандарту FIPA, JADE забезпечує певний ступінь збігу у використанні комунікаційних актів FIPA та мови їх вмісту SL (семантичної мови), яка визначає спосіб обміну повідомленнями між агентами. Однак необхідно визначити конкретні онтології з власним словником і семантикою змісту повідомлень, якими обмінюються агенти.

Існувала велика проблема щодо включення методів навчання, методів аналізу великих даних тощо в МАС. Однак найважливішим завданням є розробка архітектури МАС, яка б дозволила інтегрувати її практично в будь-який ІВ.

На даний момент існує три відкритих протоколи для зв'язку з системами HVAC, які були швидко створені за допомогою стандартних протоколів для зв'язку з HVAC: ModBus, LonWorks і BACnet. Ці протоколи мають набір характеристик, які роблять їх придатними для різних цілей і застосувань [17]. Відповідно до опитування, проведеного на веб-сайті Building Operating Management у 2011 році [18], 30% респондентів мали принаймні одну програму з використанням Modbus, 40% для Lon-Works і 62% для BACNet.

Так як мультиагентні системи розроблені для рішення складних проблем, створення їх архітектури є непростим завданням. Існують різні методи, які спрощують це завдання. Традиційні об'єктно-орієнтовані підходи можуть бути підходящими для цієї мети. Агенти використовують деякі механізми та філософію, яка використовується для об'єктів. МАС можуть публікувати опис послуг, які вони надають, у реєстрі сторінок (наприклад, Directory Facilitator у фреймворку JADE) і спілкуватися за допомогою спільних повідомлень на основі онтології. Також можлива мобільність, яка дозволяє агентам мігрувати на іншій пристрій, зберігаючи свій стан виконання. МАС спеціалізуються на об'єктно-орієнтованій методології програмування. Архітектура агентів легка у розробці та реалізації, коли необхідно працювати з комп'ютерно-інтенсивними системами, які потребують масштабування для впорядкування великої кількості завдань. Найбільш використовувані методології в головних дослідженнях МАС включають GAIA, MaSe і Tropos. GAIA - це універсальна методологія, зосереджена лише на аналізі та дизайні, в той час як MaSE і Tropos фокусуються на цілях та плануванні агентів.

## 2.4 Автоматизація завдань у розумних будинках

У розумних будинках автоматизація завдань може бути реалізована за допомогою різних технологій та пристроїв. Автоматизацію в розумному будинку можна здійснити так:

- Використання розумних пристроїв, таких як розумні датчики, розумні лампи, термостати тощо, дозволяє створити автоматичні сценарії. Наприклад, ввімкнення світла при вході в кімнату або автоматичне вимкнення пристроїв, коли ви покидаєте будинок.

- Використання розумних застосунків або платформ, таких як IFTTT (If This, Then That) або SmartThings, дозволяє вам створювати складніші автоматичні сценарії, включаючи реакцію на певні події або умови. Наприклад, ви можете налаштувати автоматичне включення системи опалення, якщо температура падає нижче певного рівня.

- Використання розумних хабів, таких як Samsung SmartThings або Hubitat, дозволяє об'єднувати різні розумні пристрої в єдину систему керування. Це дозволяє створювати складні автоматичні сценарії, що включають взаємодію між різними пристроями.

- Використання геолокації на вашому смартфоні або інших пристроях дозволяє автоматично виконувати дії, коли ви підходите або відходите від будинку. Наприклад, автоматичне відкривання гаражних воріт, коли ви під'їжджаєте до будинку.

## 2.5 Проблеми створення розумних будинків

Поширеність розумних пристроїв та додатків для дому призводить до фрагментації і складності управління такими системами, що становить значну проблему для власників будинків.

В той час як окремі компанії, такі як Amazon, Google, Apple та інші, пропонують власні рішення для розумного будинку, вони часто акцентуються на контролі з допомогою смартфона, а не на організованому підході до інтеграції всіх пристроїв. Ця відсутність інтеграції та єдиного підходу

ускладнює завдання для власників будинків, які нерідко зустрічаються з викликами в управлінні такими розсипаними технологіями для розумного будинку [13].

Зрозуміло, що вибір певної екосистеми, такої як Amazon, Google або Apple, і її використання для створення розумного будинку спрощує ситуацію. Однак, багато власників будинків використовують різні пристрої від різних виробників, що ставить перед ними виклик у питанні інтеграції та управління цими різноплановими системами. Потрібна єдина платформа або операційна система, яка забезпечує зручне управління всіма пристроями та додатками розумного будинку, незалежно від виробника.

Таблиця 2.1 – Огляд найпопулярніших систем для систем розумних будинків.

| Назва системи       | Переваги   | Недоліки   |
|---------------------|--|--|
| Amazon Alexa        | широкий список підтримуваних пристроїв, потужні рутини           | не найбільш зручний інтерфейс, обмеження у керуванні окремими пристроями через додаток |
| Google Home         | простий інтерфейс, широкий список підтримуваних пристроїв.       | обмеження у керуванні деякими пристроями через додаток.                                |
| Samsung SmartThings | зовнішній вигляд, зручна настройка, підтримка живого стрімінгу   | мало підтримуваних сервісів  |
| Olisto              | автоматизація різних дій, створення спеціальних кнопок           | складна настройка, обмежена підтримка сервісів   |
| iHaus               | легкість налаштування, автоматичне виявлення пристроїв           | обмежена підтримка пристроїв, незатребуваний інтерфейс                                 |
| IFTTT               | автоматизація через широкий список партнерів, наявність віджетів | необхідність вручну налаштовувати більшість макросів                                   |
| Home Remote         | інтуїтивний інтерфейс, можливість гнучкого налаштування          | менша підтримка пристроїв порівняно з Alexa та Google Home                             |

Основна проблема полягає в тому, що розумний будинок має бути дійсно розумним і забезпечувати простоту в управлінні та взаємодії між різними пристроями. Замість того, щоб витрачати час та зусилля на налагодження та синхронізацію великої кількості окремих додатків та пристроїв, власникам будинків потрібна єдина інтуїтивно зрозуміла платформа, яка забезпечує їх потреби щодо комфорту, безпеки та енергоефективності.

Така єдина платформа для управління розумним будинком може бути заснована на мультиагентній системі, де різні агенти взаємодіють між собою для спільної оптимізації функцій будинку. Це дозволить автоматизувати різні аспекти будинку, від освітлення та опалення, безпеки та контролю енергоспоживання до годування домашніх улюбленців та відстежування їх стану. Мультиагентна система може забезпечити гнучкість, адаптивність та інтелектуальну взаємодію, що дозволить розумному будинку стати справді універсальним середовищем, яке задовольняє потреби мешканців та їх домашніх тварин у комфорті та безпеці.

## 3 РОЗУМНЕ МІСТО

### 3.1 Динаміка мультиагентних систем

Зростаюча концентрація населення в міських районах і подальші виклики, що виникають, підкреслили потребу в розумних способах полегшення життя громадян, надання послуг і пом'якшення наслідків стихійних лих. Одним з рішень було впровадження нових міських районів, обладнаних системою моніторингу з розвинутою інфраструктурою обліку та "розумними" об'єктами з можливістю постійного доступу до інформації та вбудованим інтелектом. Кожен з розумних об'єктів збирає дані зі свого оточення, взаємодіє з іншими об'єктами, обробляє інформацію і, в деяких випадках, автономно реагує на динамічні внутрішні та зовнішні зміни. Прикладами таких інтелектуальних пристроїв є бездротові сенсорні мережі (WSN - Wireless sensor networks), радіочастотна ідентифікація (RFID - radio frequency identification), комунікації ближнього поля (NFC - near-field communications), унікальні/універсальні/повсюдні ідентифікатори (UID – unique/universal/ubiquitous identifiers), приводи, смартфони та розумні пристрої. Пристрої з'єднані через платформу під назвою «Інтернет речей» (IoT), яка дозволяє технологіям отримувати доступ і обмінюватися даними через бездротові мережі та мережі Інтернет-провайдерів.

Незважаючи на те, що впровадження Інтернету речей відкриває нові можливості інтеграції різноманітних пристроїв з різними функціональними можливостями, обчислювальними можливостями та потоками даних, це є надзвичайно складною задачею, враховуючи високодинамічний і розподілений характер цих мережевих систем [19]. Кожен сенсорний пристрій та інші кінцеві користувачі можуть у будь-який час приєднатися до системи, вийти з неї або змінити своє розташування, що робить топологію системи невизначеною та схильною до неочікуваних змін. Механізми керування з

фіксованими конфігураціями неефективні для неоднорідної та динамічної природи систем IoT. Розподілений характер IoT також робить його більш вразливим до можливих кібератак і збоїв. Збої в частині системи, особливо в центральному лідері, можуть каскадно переходити до інших підключених вузлів і в кінцевому підсумку призвести до збою всієї системи. Ці системи мають бути розроблені та керовані більш надійним та стійким способом, сприяючи ефективності системи в наданні послуг та досягненні заздалегідь визначених цілей і цілей.

Розподілена обробка та керування багатоагентними системами (МАС) або агентно-орієнтоване програмування (АОР - agent-oriented programming) є одними з основних технологічних парадигм для ефективного розгортання інтелектуальних пристроїв і послуг у розумному місті [20]. Ці методи вважаються найкращими абстрактними підходами для моделювання операцій і функціональних можливостей систем IoT на всіх рівнях сприйняття, мережі та застосунку. [21].

Також була доведена їх ефективність у підтримці автономних мереж об'єктів в IoT з властивостями самоадаптації та самоорганізації [22].

Існує велика кількість миттєвих комунікацій між пристроями в IoT. У МАС кожен пристрій підключений до агента із заздалегідь визначеним набором функцій і можливостей. Це забезпечує відповідну високопродуктивну інфраструктуру для тестування та впровадження великомасштабного збору даних і пропонує масштабовану платформу для розподіленого обчислення отриманих даних [23].

Крім того, симуляція об'єктів як інтелектуально-реактивних агентів дозволяє в режимі реального часу налаштовувати параметри керування в складних взаємопов'язаних мережах, таких як IoT. Автономні агенти в МАС можуть бути запрограмовані та працювати без втручання людини. Вони здатні самостійно взаємодіяти з іншими агентами та обробляти модульні функції та координацію.

Самоадаптивність є однією з вимог у розробці поширених

взаємопов'язаних систем, таких як IoT та розумні міста [24]. Кооперативний характер систем на основі MAC та їх здатність до навчання дозволяють розробляти надійні рішення, які можуть адаптувати свою конфігурацію та роботу в разі неочікуваних змін і збоїв. Щоб досягти ефективності у вирішенні неочікуваних ситуацій, необхідно вчитися на попередніх подіях, таких як різкі зміни навколишнього середовища та внутрішні збої, і постійно адаптувати контрольні параметри та стратегії. Самоадаптивна MAC постійно отримує зворотний зв'язок від внутрішніх блоків і зовнішнього середовища. Отримані дані відстежуються та аналізуються, а сприйняття передаються в модуль планування для прийняття рішення про наступну дію та реакцію системи. Команди управління визначаються з урахуванням головної мети системи та ролей і відповідальності підрозділів у досягненні бажаного рівня. Потім дії надсилаються назад до виконавчих блоків, а результати збираються для оновлення наступних дій і, зрештою, оцінки та модифікації системи.

Характеристики адаптивної MAC можна ідентифікувати на основі її сутностей, операцій та дій, потоку інформації та зовнішнього середовища, в якому вона функціонує. Зведення впливаючих факторів при визначенні MAC, представлено на рисунку 3.1.

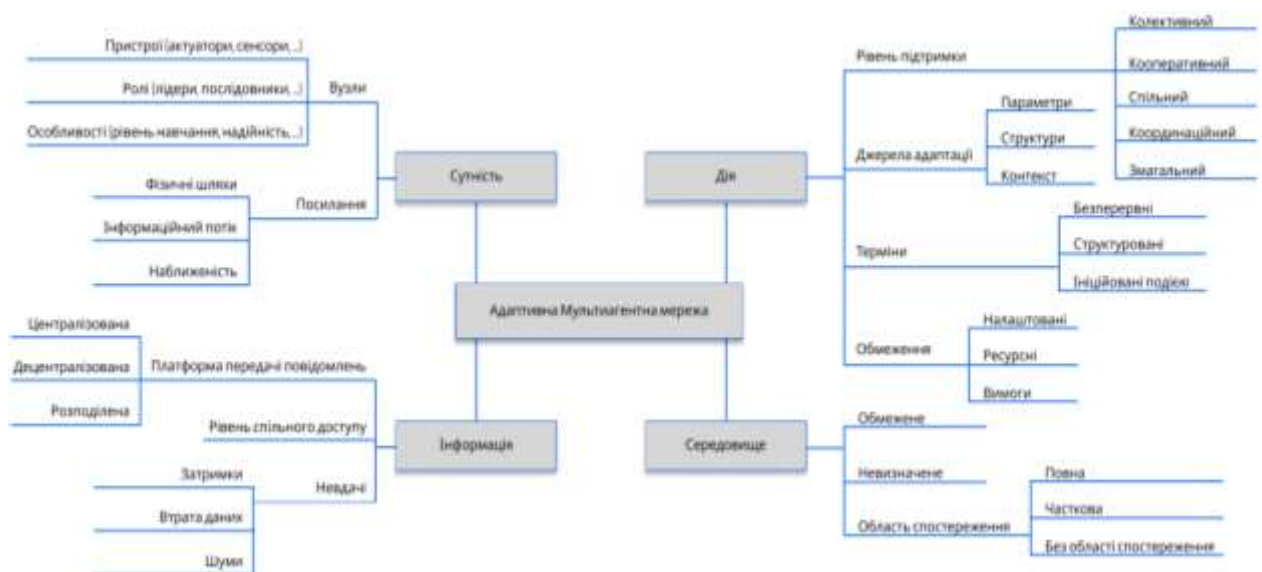


Рисунок 3.1. – Фактори, що сприяють визначенню адаптивних MAC.

### 3.2 Навколишнє середовище

Агенти взаємодіють і впливають на навколишнє середовище під час свого функціонування. Середовище визначається на основі його меж, рівнів невизначеності та спостережливості. Межі середовища в основному залежать від застосування МАС і в цілому включають спільну фізичну, комунікаційну, соціальну структуру та простір агентів, а також ресурси та сервіси, що визначають обмеження, правила взаємодії та відносини між ними [25]. Фактично, ця абстракція розглядається як зовнішній світ, що забезпечує середовище для координації, зберігаючи незалежність процесів від дій операційних агентів. Межі середовищ також визначаються на основі взаємодії між ними та механізмів управління ресурсами і контекстом. Інші зовнішні суб'єкти, які взаємодіють з агентами або контролюються ними, також вважаються частиною середовища. У мультиагентному програмуванні середовище представляє собою моделі, засновані на таких аспектах, як модель дій, модель сприйняття агентів, обчислювальна модель для внутрішніх і зовнішніх функціональних можливостей, модель обміну даними між агентами і середовищем, а також модель розподілу агентів [26].

У більшості реальних МАС, особливо в області IoT, агенти працюють у невизначеному середовищі. Тому необхідно досліджувати рівень невизначеності середовища та приховану динаміку, що лежить в основі. Системи можуть працювати в детермінізованому середовищі або піддаватися різноманітним невизначеностям, які впливають на керованість МАС. Невизначеність може бути викликана невідомим значенням функції або обмеженням системи. Наприклад, розташування інших агентів у плануванні шляху, поведінка агентів і налаштування параметрів у гетерогенному управлінні можна розглядати як джерело невизначеності в середовищі агента. Невизначеність у комунікаціях - ще одне явище, яке ставить під загрозу правильного функціонування мереж МАС, зокрема, в системах з декількома роботами [27]. Невідомі збої можуть бути одним із джерел невизначеності в

середовищі MACs. Іноді нестабільність є результатом непередбачуваної реакції між агентами та навколишнім середовищем. Середовище може бути невизначеним внаслідок того, що цілі з'являються у будь-який момент часу. Джерелом виникнення неоднозначності також може бути один з параметрів середовища. Невизначеність середовища може проявлятися у вигляді шумів, що вносяться в систему і впливають на стан агентів. Основним джерелом появи невизначеності є дії інших агентів, які роблять середовище непередбачуваним. Така динаміка робить середовище непостійним через одночасне навчання агентів найкращій політиці, що постійно змінює загальний стан середовища. Невизначеність у середовищі зазвичай моделюється стохастичністю в рівняннях процесу, рівняннях виходу/вимірювання або каналах зв'язку між агентами [28].

Теоретично, середовище може бути змодельоване на основі однієї з категорій: повністю спостережуване, частково спостережуване або не спостережуване середовище. У повністю спостережуваних системах агенти здатні сприймати повну інформацію про стан навколишнього середовища та його елементів, таких як ресурси та інші стани агентів. Агенти в частково спостережуваному середовищі, такому як частково спостережуваний MDP (POMDP), здатні вивчати лише часткову інформацію про стани та їх розподіли ймовірностей замість абсолютних значень. У цих системах зв'язок між діями та винагородою не є чітким і потребує оцінки. У більш екстремальному випадку агенти можуть бути не в змозі сприймати нічого про навколишнє середовище і сліпо виконувати свої завдання та обов'язки. Більшість розроблених методів базуються на нереалістичному припущенні щодо спостережуваності середовища, і агенти завжди перебувають під обмеженнями, що змушує їх вивчати абстракції середовища замість деталей. Одним із головних припущень щодо спостережуваності середовища для дій агентів та їхніх винагород і покарань є середовище Маркова [29]. У цьому середовищі майбутній стан агентів визначається їхнім поточним станом. У немарківських середовищах динаміка стану є більш складною і може існувати

сильна залежність від початкових станів або змін, які є епізодичними з часом. Спостережливість також може бути обмежена інформацією агентів, отриманою від інших агентів. Наприклад, вони можуть не мати постійного доступу до мережі зв'язку та бути відключеними на певний період часу. Агенти можуть мати негативні спостереження, що впливають на їхнє сприйняття з боку навколишнього середовища та інших агентів [30].

### 3.3 Інформаційний потік між агентами

Для розробки ефективного механізму управління МАС необхідно звернути увагу на інформаційний потік між агентами, який може бути визначений у вигляді платформи передачі повідомлень, рівня обміну інформацією та збоїв зв'язку в часі. Інформаційний потік в адаптивній мультиагентній мережі реалізується або для збору інформації про стан та результати роботи агентів та надсилання їх контролерам, або для передачі команд від контролерів до взаємодіючих агентів.

Платформа передачі повідомлень відображає рівень доступу агентів до інформації високого рівня про стан системи, цілі та обмеження. Три основні платформи для обміну інформацією між агентами та контролерами включають централізовані, багаторівневі або децентралізовані та розподілені структури (рис. 3.2). У централізованому підході один агент має доступ до інформації агента та приймає рішення за всю систему. Масштабованість, вразливість до збоїв в роботі агента-контролера та потреба у великих комунікаційних ресурсах для агентів є загальними проблемами цього підходу. Щоб подолати цю проблему, деякі дослідники користувалися такими варіантами, як параметрична схожість агентів, щоб наблизити всі налаштування лише до однієї унікальної політики [31]. Отримавши загальну політику в цій системі, агенти можуть локально досліджувати свої політики для мінімізації власних втрат. Загальна політика оновлюється інтерактивно за допомогою зворотного зв'язку, отриманого з траєкторій усіх агентів. У децентралізованих або

шаруватих підходах систему контролює більше одного контролера на кожному шарі або спільноті [32]. Цей метод вважається більш надійним, ніж попередній, але він вимагає точного визначення правил комунікації та співпраці між контролерами. Розподілена форма вважається найпоширенішою технікою управління МАС, де кожен агент відповідає за прийняття рішень і координацію з іншими агентами для досягнення головної мети системи. У цьому методі агенти мають повну автономію у виборі власних дій [33]. Цей метод є більш надійним і масштабованим.

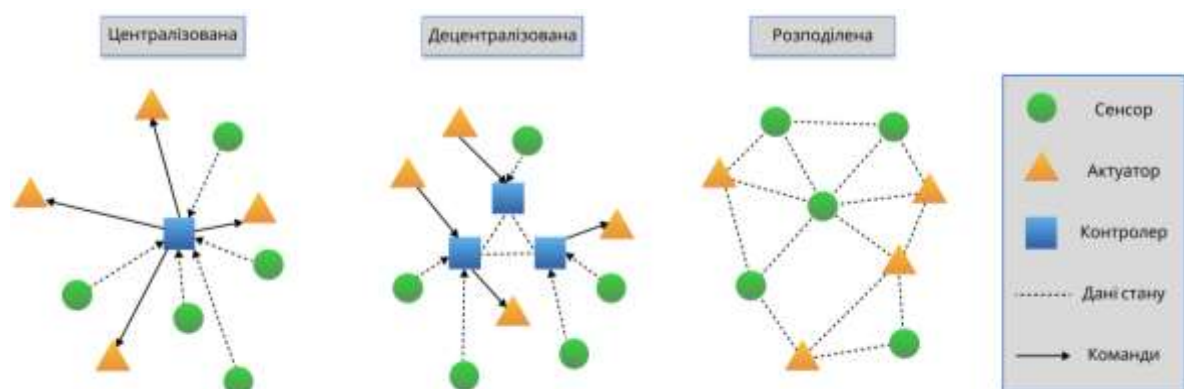


Рисунок 3.2. - Платформи потоку інформації МАС

Крім того, необхідно враховувати рівень обміну знаннями та інформацією між агентами. Відомо, що ступінь обміну інформацією безпосередньо впливає на процес навчання агентів та ефективність системи [34]. Деякі з агентів мають мінімальний обмін інформацією для отримання очікуваної винагороди, конфігурації агента або стратегії адаптації. Хоча у більшості існуючої літератури припускається, що агенти можуть спілкуватися лише зі своїми сусідніми агентами, деякі дослідження були спрямовані на інші типи комунікацій між агентами, такі як розсилка або комунікації через посередніх агентів [35]. Проміжне програмне забезпечення може виступати в ролі "посередника", який підбирає потрібну інформацію для потрібного агента, або брокера, який фільтрує або переформулює передану інформацію, а потім поширює її серед відповідних агентів. Агенти можуть використовувати менеджер трасування, який отримує дані і надсилає їх

підписаним агентам на основі їхніх інтересів і зареєстрованих запитів. Існує і більш гнучкий обмін інформацією, за допомогою якого агенти можуть порівнювати та вибирати власні комунікаційні платформи зі списку декількох джерел сигналу. Архітектура цільових комунікацій допомагає агентам обирати зміст і одержувачів своїх повідомлень у кооперативному, конкурентному або змішаному середовищі.

Ця комунікаційна та інформаційна динаміка піддається збоям і проблемам, таким як: затримки, втрати даних, помилки квантування, шуми, невідома динаміка, згасання каналів, конкуренція за доступ до вузлів та інтервали вибірки. У затримках зв'язку інформація про стан сусідніх агентів отримується із затримками через обмеження пропускної здатності каналів зв'язку та перевантаження мережі. У деяких випадках затримки не є фіксованими і змінюються через відмінності у спостережуваності стану в часі [36]. Це відрізняється від вхідних затримок, які відображають час обробки та з'єднання вхідного потоку даних у МАС. Іншим впливовим фактором є врахування втрати даних при визначенні зв'язку МАС. Інформаційна динаміка розроблена на основі швидкості відсутньої інформації під час спілкування та корисності зібраної інформації для процесу прийняття рішень. Шуми в статусах і вимірюваннях агентів є ще однією проблемою, що впливає на якість потоку інформації в мультиагентних мережах.

### 3.4 Парадигми моніторингу

Щоб вижити в динамічному середовищі, адаптивна мультиагентна система потребує постійного моніторингу свого внутрішнього та зовнішнього стану. Дані, зібрані агентами розподіленого зондування, обробляються для дослідження середовища, оцінки продуктивності системи та визначення наступних оптимальних дій. Аналіз даних може бути реалізований в центральному процесорі або в розподілених локальних процесорах. Централізована єдина обробка даних буде дуже складною, враховуючи

складні відносини між агентами та високий рівень мультиколінеарності між змінними. Традиційні методи інтелектуального аналізу даних для плоского векторного аналізу даних неефективні для обробки великомасштабних даних із реляційними залежностями, вагами, напрямками країв і неоднорідністю між елементами системи [37]. Іншою проблемою для обробки даних, зібраних через МАС, є обмеження ресурсів для передачі, обробки та зберігання великих обсягів даних великої розмірності, зібраних агентами з часом. У літературі було запропоновано широкий спектр технік для полегшення збору даних, і одним з найбільш прийнятних підходів є використання подієвого відбору даних. У цих техніках алгоритми глибокого навчання, такі як графові нейронні мережі (GNN), графові згорткові мережі (GCN), графові автокодери (GAE), графові рекурентні нейронні мережі (GRNN) або навчання з підкріпленням графів, широко застосовувалися для обробки даних, зібраних із взаємопов'язаних елементів або агентів системи [38]. Ці підходи застосовувалися для одного з завдань з видобутку даних, таких як зменшення розмірності і прогнозування, а також для пошуку шаблонів, кластеризації та виявлення аномалій у масштабних мережах, наприклад, у системах з декількома агентами.

Деякими початковими методами для абстрагування МАС та їхніх зв'язків було використання теорії графів і власних векторів матриці суміжності мережі. Цих методів недостатньо для складних систем з динамічними багатовимірними агентами, тому щоб вирішити цю проблему, методи кодування, такі як GAE, були застосовані для кодування або декодування графів у вектори. Методи кодування називаються механізмами мережевого вбудовування та можуть кодувати топологічні характеристики вузлів та їхню інформацію про близькість першого та другого порядку, атрибути агентів, динаміку МАС і їхній розвиток з часом, а також дифузю інформації або динамічний розвиток ролей. Методи генеративних мережевих автоматів також застосовуються для одночасного представлення переходів між станами та трансформацій топології мережі на основі концепцій

перезапису графів. Інші глибокі нейронні методи, які можуть бути застосовані для абстракції MAC, включають рекурентні нейронні мережі довгочасної пам'яті (LSTM) і варіаційні автокодера. Деякі з цих методів можуть бути використані для моделювання агентів-супротивників на основі локальної інформації головних агентів [39]. Деякі методи дистиляції знань, такі як обрізання та декомпозиція низького рангу, також можуть бути застосовані для оригінальних систем з декількома агентами, щоб видалити надлишкову інформацію, зібрану від розподілених агентів. Традиційні методи зменшення розмірності, такі як аналіз головних компонентів, також були успішно модифіковані та застосовані для мультиагентних мереж, таких як WSN [40]. Єдиним недоліком цих методів є довгий час обчислення для великомасштабних систем і втрата зразка сусідства агентів. Це можна вирішити, зосередившись більше на структурних ролях вузлів та збільшивши гнучкість навчання представлень вузлів. Навчання на основі спільного комунікативного підходу через автокодування було ще одним рішенням для зменшення часу обчислень та підвищення загальної продуктивності системи [41].

Віддалений доступ різних пристроїв і розподілений характер обчислень в IoT роблять його вразливим до різноманітних атак. Аномалії - це стани системи, які значно відрізняються від нормальної роботи системи. Існують методи, які можуть бути застосовані для зловмисних взаємодій або атак, які можуть видаляти або маніпулювати даними, пов'язаними зі структурою мережі та статусами вузлів або зв'язків. Аномалії можуть бути трьох різних рівнів: точкові, контекстні та колективні аномалії. У точкових аномаліях нерегулярність відбувається в одному агенті, яку можна спостерігати без будь-якої причини. Аномалії контексту включають більший спектр аномалій агента з часом. Наприклад, комунікаційні шаблони можуть змінюватись в частині MAC. У колективних аномаліях одні агенти можуть здаватися цілком нормальними, але сукупність даних, зібраних від агентів, демонструє незвичайні моделі. При виявленні аномалії на рівні вузла ідентифікуються

агенти, які значно відрізняються від інших агентів. На платформі IoTs це може виглядати як аномальні користувачі або мережеві зловмисники, які вводять фальшиву інформацію в систему [42]. Ці вузли також можуть представляти агентів, продуктивність яких значно відрізняється від решти агентів. При виявленні аномалій на рівні краю виявляються незвичайні та несподівані зв'язки та відносини між агентами. Виявлення аномалій на рівні підграфа зосереджено на кількох агентах, які разом демонструють аномальну поведінку. Ідентифікація цієї групи агентів буде дуже складною, і зазвичай для ідентифікації щільних блоків у цих мережах використовуються двосторонні графи. Іноді аномалії ідентифікуються на рівні графів і на певних знімках часової системи, помітні її незвичайні закономірності та особливості. Для цієї мети широко застосовувалися глибокі нейронні мережі, такі як LSTM [43]. Мультиагентні системи представлені як ефективна платформа для методів виявлення аномалій для систем IoT. Інші методи, такі як кластеризація Kmean, довели свою ефективність для виявлення незвичайних моделей зібраних даних у розподілених мережах.

Прогностичні моделі можуть бути застосовані на рівні системи або об'єкта для прогнозування певних особливостей і характеристик з часом. Наприклад, знання та передбачення майбутніх топологічних змін, таких як видалення та додавання агентів і нових шаблонів зв'язку, може мати істотний вплив на розробку протоколів керування. Глибокі генеративні моделі графів широко використовуються для цієї мети для моделювання мережевої структури без знання її структурної інформації, вони можуть використовувати рекурентні нейронні мережі для прогнозування майбутніх з'єднань або ребер у графі на основі процедур упорядкування вузлів. Існують й інші методи, які використовують ефективні стратегії вибірки для вилучення шаблонів у вхідних даних і вивчення їх динаміки для створення прогнозованої тимчасової мережі. Окрім структурних моделей системи, оцінка майбутніх станів інших агентів може допомогти агентам оптимізувати власні дії більш ефективно. Було доведено, що згорткові нейронні мережі успішно фіксують просторово-

часові характеристики мережевих об'єктів і передбачають їхні майбутні характеристики (вузли) і комунікації (краї) з сусідніми агентами [44]. Як приклад, прогнозування майбутніх траєкторій агентів є однією з поширених проблем у плануванні робіт, автономному водінні та прогнозуванні руху [45]. Інше застосування прогностичних моделей полягає в ранньому виявленні подій у техніках, що ініціюються подіями, або виявленні гарячих точок із незвичайною щільністю певних подій. Методи класифікації для мережі підключених агентів можуть бути застосовані для моніторингу ефективності системи та ініціювання коригувальних дій, якщо вона класифікована як аномальний граф. Однією з поширених платформ для цієї мети є DeepSphere, яка застосовувалася для вивчення моделей розвитку системи з часом і виявлення аномальних знімків мережі з часом. Розподілена оцінка певних характеристик системи або середовища є іншим прикладом прогностичних моделей на рівні системи. Ці техніки можуть працювати одночасно або потребувати консенсусу на двох етапах вибірки та використовувати широкий спектр підходів, включаючи адаптивний спостерігач, фільтр Калмана, спостерігач Люенбергера, Байєсівський фільтр, інференцію розповсюдження віри та  $H_{\infty}$ filter [46].

Моніторинг динаміки MAS може бути спрямований на кластеризацію агентів на основі їх продуктивності або прихованих інтересів. Це головним чином тому, що групування агентів може допомогти виявити неспівпрацюючих агентів або розробити більш спеціалізовані контрольні фреймворки для кожної групи агентів [47]. Іноді кластеризація корисна, коли гетерогенні агенти проявляють різну динаміку у досягненні консенсусу, що вимагає більш індивідуалізованих контрольних протоколів. Подібність між групою агентів або їхніми траєкторіями можна виміряти за допомогою динамічних кореляцій між агентами або за допомогою проектів аналізу головних компонентів (РСА). Агенти кластеризації можуть бути досліджені шляхом застосування традиційних методів розбиття графів і виявлення спільнот, таких як спектральна кластеризація, ієрархічна кластеризація, моделі

Маркова та методи максимізації модульності. Глибоке навчання вважається ефективним інструментом для виявлення спільноти у багатовимірних графоструктурованих даних із кількома мітками [48]. Кластерна динаміка досліджується в управлінні роєм агентів у формуванні та їх агрегації, а також моделі розщеплення та інші кількісні характеристики, такі як розмір і відстань кластера один від одного. Іноді кластеризація є динамічним процесом, у якому агенти з подібною поведінкою знаходять один одного та об'єднуються в просторі станів - це схоже на динаміку методів групового консенсусу.

Щоб розробити більш надійні та стійкі МАС, вони повинні мати можливість вивчати поведінку інших агентів і динаміку середовища, щоб покращити свою продуктивність з часом. Інтелектуальний аналіз зразків вважається одним із основних методів для відкриття знань і ідентифікації причинно-наслідкових структур і асоціацій. Наприклад, мотиви трендів майнінгу допомагають ідентифікувати групу вузлів або агентів, які демонструють подібні тенденції зростання або зменшення з часом. Більш складні тенденції, такі як повторювані тенденції, ідентифікуються в наборі вузлів протягом послідовності часових інтервалів за допомогою алгоритмів, таких як RPMiner [49]. Ідентифіковані шаблони в атрибутах і станах можуть спричинити за собою зміни в топологічних структурах мережі та зв'язку між агрегатами, які відомі як тригерні шаблони. Доказано, що в аналізі шаблонів багатоагентних мереж покладається на частоту появи іноді вводять в оману, і нові показники, такі як швидкість віртуального зростання послідовності, необхідні для ідентифікації висококорельованих шаблонів зі значною послідовністю тенденцій на графіку. Дослідження норм — це ще одна категорія, яка досліджує події, що викликають винагороди та штрафи, і визначає норми в змінному середовищі. Це допоможе агентам вижити й адаптуватися до середовища, продовжуючи виконувати свою роботу.

## 4 ЗАСТОСУВАННЯ АГЕНТІВ У РОЗУМНИХ БУДИНКАХ

### 4.1 Мультиагентні системи

Мультиагентна система - це система, в якій кілька агентів (програмних або фізичних сутностей) працюють разом, взаємодіючи та співпрацюючи для досягнення спільної мети. Кожен агент має свою внутрішню структуру та може мати власні цілі, знання та можливості дії. У контексті розумних будинків, мультиагентні системи використовуються для автоматизації та керування різними пристроями та системами у будинку. Вони забезпечують співробітництво та координацію між різними розумними пристроями, такими як освітлення, опалення, системи безпеки, побутова техніка та інші.

Переваги мультиагентної системи перед окремими застосунками в контексті розумних будинків можуть бути наступними:

– Синергія та співробітництво: Мультиагентна система дозволяє різним розумним пристроям та застосункам взаємодіяти між собою, обмінюватися інформацією та координувати свої дії. Це дозволяє створювати складні сценарії автоматизації, де різні пристрої можуть взаємодіяти та виконувати завдання спільно, підвищуючи ефективність та зручність для користувачів.

– Гнучкість та розширюваність: Мультиагентні системи можуть бути легко розширені шляхом додавання нових агентів або розумних пристроїв до системи. Це дозволяє розширювати функціональні можливості розумного будинку без потреби в значних змінах або перепрограмуванні всієї системи.

– Розумний аналіз даних: Мультиагентні системи можуть здійснювати аналіз та обробку даних з різних джерел, включаючи сенсори, вимірювальні пристрої, зовнішні системи тощо. Це дозволяє системі отримувати більш повне та точне розуміння контексту та потреб користувачів, що сприяє більш ефективній автоматизації завдань.

– Адаптивність: Мультиагентні системи можуть бути програмовані для виявлення та адаптації до змінних умов та потреб користувачів. Вони можуть використовувати навчання на основі даних для постійного вдосконалення своєї роботи та адаптації до змін у середовищі.

#### 4.2 Використання багатоагентних систем для розумного дому

Мультиагентна система може бути використана для забезпечення комфорту та безпеки у розумних будинках, надаючи ряд можливостей і переваг. Незалежно від того, чи це контроль доступу, виявлення пожежі або небажаної активності в приміщенні, МАС працює в реальному часі для оперативного реагування на потенційні небезпеки.

Стандартні системи пожежної сигналізації зазвичай мають обмеження у вигляді можливості роботи лише у внутрішніх приміщеннях, мають багато помилкових спрацьовувань, не можуть надавати актуальну інформацію пожежникам [14] і, що найважливіше, мають значну затримку між початком пожежі та її виявленням [15].

Крім того, МАС можуть виявляти та вирішувати проблеми, пов'язані з комфортом, а саме доглядом за домашніми тваринами. Вони можуть встановлювати розклад годування та години виходу, контролювати рівень води, її чистоту та температуру в акваріумі, а також здійснювати візуальний контроль за тваринами за допомогою вбудованих камер. Оскільки багато людей живуть зі своїми домашніми тваринами та не завжди можуть мати можливість знаходитись зі своїми улюбленцями протягом всього дня, тому використання МАС може дозволити власникам тварин спокійно перебувати поза домом, знаючи, що їхні улюбленці отримують необхідний догляд і захист.

Також МАС може надати власникам можливість візуального спостереження за своїми тваринами. Завдяки вбудованим камерам і зв'язку з мобільними пристроями, власники можуть віддалено спостерігати за тваринами, перевіряти їхню активність та стан здоров'я. Це дозволяє вчасно

виявляти будь-які незвичайні або небезпечні ситуації та приймати відповідні заходи [16]. МАС здатна вирішувати потреби різних видів домашніх тварин, починаючи від собак і котів до екзотичних птахів, рептилій чи акваріумних рибок. Така система потребує використання спеціалізованих агентів, які можуть розпізнавати своє оточення і розподіляти різні завдання у просторовому і часовому аспектах, де деякі агенти можуть вживати заходів негайно, тоді як іншим потрібно чекати на підтвердження та зворотний зв'язок від своїх сусідів.

Однак, використання агентів у розумному будинку може стикатися з деякими проблемами:

Одна з основних проблем полягає у забезпеченні сумісності між різними агентами та пристроями у розумному будинку. Різні виробники можуть використовувати різні протоколи комунікації та стандарти, що ускладнює їх взаємодію та інтеграцію.

Розумний будинок, який використовує агентів, може стати цільовою точкою для кібератак. Якщо агенти не мають достатнього рівня безпеки, зловмисники можуть зламати їх та отримати доступ до системи управління будинком, що може мати серйозні наслідки для приватності та безпеки мешканців. Також розумний будинок може бути вразливим до третіх осіб через незахищені мережеві з'єднання, вразливості в програмному забезпеченні, недостатнє шифрування даних, слабкі паролі або відсутність двофакторної автентифікації.

Проблеми з приватністю у розумних будинках можуть виникати через збір і передачу великої кількості даних про особисте життя та поведінку мешканців. Ці дані можуть включати інформацію про їхні звички, час місцезнаходження в будинку, використання побутових приладів тощо. Якщо ці дані потраплять до рук третіх осіб без належного захисту, існує ризик витоку конфіденційної інформації. Також проблеми можуть виникнути, якщо виробники чи провайдери розумних технологій зловживають доступом до цих даних або якщо системи не належним чином захищені від кібератак.

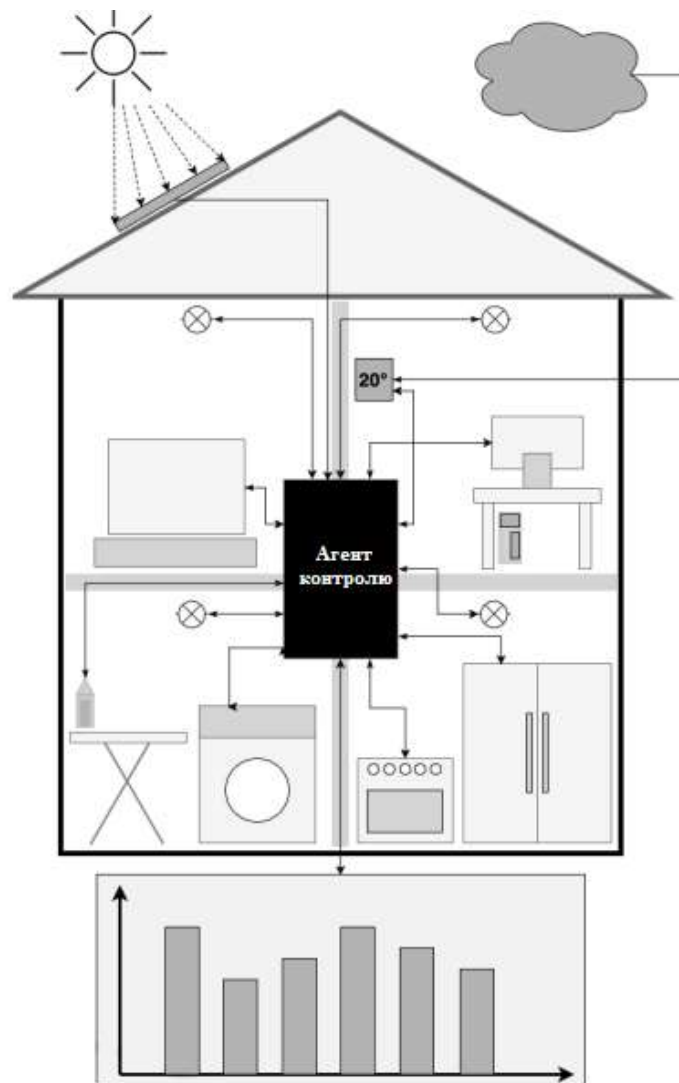


Рисунок 4.1 – Ілюстрація агента контролю у середовищі

Іноді налаштування та конфігурація агентів у розумному будинку може бути складною для непрофесійних користувачів. Це може потребувати глибоких знань з певних технологій та вимагати значного зусилля для налагодження оптимального функціонування системи.

Багато розумних систем та агентів потребують постійного з'єднання з Інтернетом для правильної роботи. В разі відсутності Інтернету або збоїв в мережі це може призвести до неправильної роботи або недоступності певних функцій системи, такі як неможливість дистанційного керування або моніторингу за безпекою та надсиланням важливих сповіщень.

Впровадження розумного будинку з агентами може бути досить витратним. Вартість придбання агентів, датчиків, пристроїв з підтримкою

розумного дому та їх налаштування можуть бути значними. Крім того, може знадобитися постійне оновлення технологій та програмного забезпечення, що також призводить до додаткових витрат.

З використанням багатьох агентів та пристроїв, управління розумним будинком може стати складним завданням. Користувачам може бути потрібно опанувати різні інтерфейси та програми для керування різними аспектами будинку. Це може створити певну плутанину та неефективність у використанні системи.

Якщо агенти у розумному будинку не працюють належним чином або виникають проблеми з їх зв'язком, це може призвести до несправностей в автоматизованих функціях будинку. Наприклад, неправильне включення або вимкнення освітлення, незаплановане відкриття дверей або проблеми з опаленням.

У деяких випадках вже наявні системи у будинку, які не підтримують розумний функціонал, тому інтеграція агентів може бути викликана складнощами. Наприклад, якщо в будинку встановлена застаріла електрична проводка, може знадобитися додаткова робота для забезпечення сумісності з розумними пристроями та агентами.

Враховуючи ці проблеми, важливо ретельно планувати та обирати відповідні агенти та системи для розумного будинку, забезпечувати безпеку майна та користувачів цієї системи.

### 4.3 Архітектура МАС для забезпечення комфорту та безпеки

Створення моделі мультиагентної системи для забезпечення комфорту та безпеки у розумних будинках може бути виконано за допомогою розробки ролей та взаємодії між різними агентами.

Архітектура МАС [52] складається з таких компонентів (рис. 4.2.):

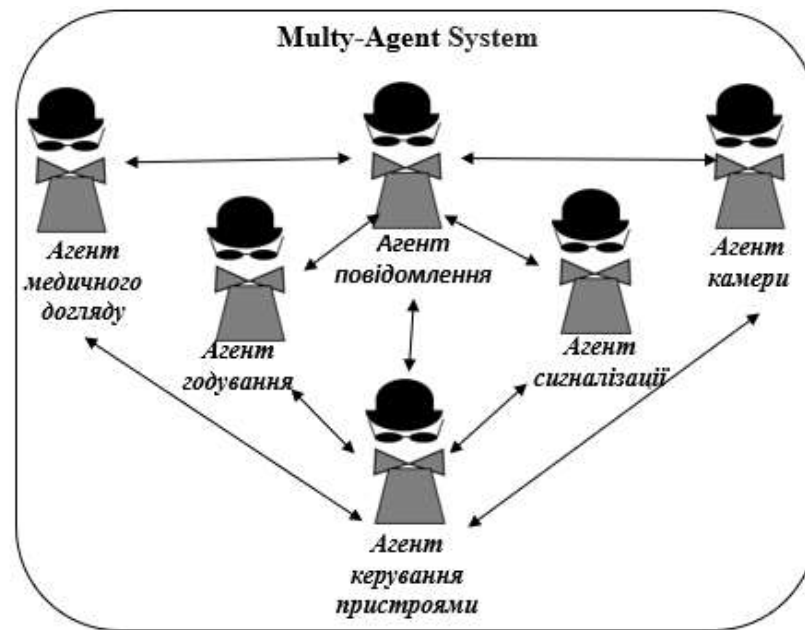


Рисунок 4.2 – Основні актори мультиагентної системи

1. Агенти. Агенти представляють окремі підсистеми або пристрої. Кожен агент має власну функціональність і завдання:

- *Агент повідомлення* (*MessagingAgent – Ag<sub>MS</sub>*), який у реальному часі надає інформацію про систему та її агентів [53]. В його функції (*A<sub>MS</sub>* – алгоритми розповсюдження повідомлень) входить отримання та аналіз інформації від інших агентів та сенсорів в будинку та розповсюдження повідомлень та сповіщень користувачам

- *Агент керування пристроями* (*DeviceControlAgent – Ag<sub>DC</sub>*) обробляє показники з групи датчиків та може вживати спеціальних заходів відповідно до подій. В його функції (*A<sub>DC</sub>* – алгоритми автоматизації) входить керування підключеними пристроями в будинку (освітлення, температурний режим, смарт-пристрої тощо) та виконання команд користувача або автоматична адаптація до змін у середовищі

- *Агент камери* (*CameraAgent – Ag<sub>CM</sub>*) спостерігає за фізичним середовищем та аналізує потокове відео для надання візуального підтвердження подій. В його функції (*A<sub>CM</sub>* – алгоритми виявлення руху) входить виявлення руху та аналіз відеоданих для забезпечення безпеки та генерація сповіщень у випадку виявлення підозрілої діяльності.

– *Агент сигналізації (AlarmAgent – Ag<sub>Al</sub>)* переводить систему в режим сигналізації та активує відповідних агентів, коли подія підтверджується. В його функції ( $A_{Al}$  – алгоритми реакції на події) входить керування системою сигналізації та безпеки в будинку та виявлення потенційно небезпечних ситуацій та висилання тривожних сигналів.

– *Агент годування (FeedingAgent – Ag<sub>Fd</sub>)* відповідає за автоматичне годування тварин згідно з попередньо налаштованим графіком або за запитом власника. В його функції ( $A_{Fd}$  – алгоритми регулювання харчування) входить моніторинг та автоматизація системи харчування в будинку та забезпечення оптимальних умов для годування та регулювання дієти користувачів.

– *Агент контролю клімату (ClimateAgent – Ag<sub>Cl</sub>)* відповідальний за контроль температури, вологості, освітлення у приміщенні ( $A_{Cl}$  – алгоритми клімат контролю);

– *Агент медичного догляду (MedicalCareAgent – Ag<sub>MC</sub>)* включає систему виявлення незвичайної активності або змін у поведінці тварини, що можуть свідчити про проблеми зі здоров'ям та сповіщати власника про потенційні проблеми або забезпечувати нагадування про потребу у відвідуванні до ветеринара. В його функції ( $A_{MC}$  – алгоритми аналізу стану здоров'я) входить відстеження медичних параметрів користувачів (пульс, тиск, рівень цукру тощо) та генерація тривожних сигналів у випадку виявлення небезпеки для здоров'я.

Агенти взаємодіють між собою для обміну інформацією, координації дій та прийняття рішень. Вони можуть використовувати протоколи комунікації для передачі повідомлень та даних між собою. Це дозволяє їм працювати в команді для досягнення загальних цілей, наприклад, забезпечення безпеки та догляду за тваринами.

2. Середовище: Це фізичне або віртуальне оточення, в якому діють агенти. В розумних будинках різні приміщення, простори та пристрої, які використовуються для догляду за тваринами та забезпечення безпеки. Середовище включає:

– Сенсори та пристрої вимірювання, такі як рухові датчики, температурні сенсори, камери відеоспостереження та інші пристрої вимірювання. Ці сенсори надають агентам інформацію про стан оточуючого середовища та дії користувачів.

– Мережа підключених пристроїв (IoT), що дозволяє агентам взаємодіяти з ними та керувати ними. Приміром можуть бути розумні термостати, освітлення, домашні асистенти та інші сучасні пристрої.

– Голосові та сенсорні інтерфейси, для взаємодії користувачів з системою. Голосові команди, дотики екрану чи жести використовуються для введення команд та взаємодії з системою.

– Центральний сервер або хмарні ресурси використовуються для обробки великої кількості даних та координації роботи агентів. Для цього використовується штучний інтелект для аналізу та прийняття рішень на основі зібраних даних.

– Системи безпеки та захисту, такі як система відеоспостереження, система виявлення вторгнень та сигналізація. Агенти сигналізації та камери співпрацюють для забезпечення безпеки приміщення.

– Інтеграція з медичними пристроями. Якщо користувач має медичні пристрої, такі як монітори пульсу чи термометри, вони також підключені до системи. Агент медичного догляду може використовувати ці дані для відстеження стану здоров'я та вчасного реагування на проблеми.

Таке середовище створює інтелектуальний розумний будинок, в якому агенти співпрацюють, обмінюються інформацією та автоматизують різні аспекти повсякденного життя для забезпечення комфорту, безпеки та ефективності.

Опис мультиагентної системи розумного будинку з використанням теорії множин виконаний шляхом визначення множин агентів та їхніх характеристик. [54]:

$$MAS = \{Ag, A, E\}, \quad (4.1)$$

де  $Ag = \{Ag_{Ms}, Ag_{DC}, Ag_{Cm}, Ag_{Al}, Ag_{Fd}, Ag_{Cl}, Ag_{MC}\}$ , – набір агентів, що діють у середовищі  $E$ ;  $A = \{A_{Ms}, A_{DC}, A_{Cm}, A_{Al}, A_{Fd}, A_{Cl}, A_{MC}\}$ , – набір дій агента ( $A_i = \{a_1^i, a_2^i, \dots, a_n^i\}, i = \left( \begin{array}{l} \text{MessagingAgent, DeviceControlAgent, CameraAgent, AlarmAgent,} \\ \text{FeedingAgent, ClimateAgent,} \\ \text{MedicalCareAgent} \end{array} \right)$ );

$E = \{E_{\text{physical}}, E_{\text{information}}, E_{\text{user}}, E_{\text{medical}}, E_{\text{security}}\}$  – набір станів фізичного середовища – будинок з розміщеними сенсорами та пристроями ( $E_{\text{physical}}$ ), інформаційної інфраструктури – центральний сервер або хмарні ресурси ( $E_{\text{information}}$ ), інфраструктури інтерфейсу користувача – голосові та сенсорні інтерфейси ( $E_{\text{user}}$ ), інфраструктури медичних пристроїв – інтеграція з медичними пристроями ( $E_{\text{medical}}$ ), інфраструктури підсистеми безпеки та захисту – встановлення систем безпеки  $E_{\text{security}}$ , ( $E_j = \{e_1^j, e_2^j, \dots, e_m^j\}, j = \overline{\text{physical, information, user, medical, security}}$ )).

Опис взаємодії агентів і середовища з використанням темпоральної логіки виконаємо за допомогою формул темпоральної логіки LTL (Linear Temporal Logic) для опису властивостей в часі. Темпоральна логіка дозволяє виразити різноманітні аспекти динаміки системи та стосунків між подіями у просторі часу.

Для відображення формул темпоральної логіки на набір станів, можна використовувати модель Крипке. Модель Крипке — це математична структура, яка використовується в логіці та теорії мов програмування для моделювання динамічних систем. Вона названа на честь логіка Стефана Крипке, який розробив цю концепцію в 1963 році. Модель Крипке може бути використана для формального визначення та аналізу систем, де стани та їхні переходи грають важливу роль.

Основні компоненти моделі Крипке включають:

- Множину можливих станів системи  $S$ , де кожен елемент  $s \in S$  представляє конкретний стан.
- Відношення переходів  $R \subseteq S \times S$ . Для кожної пари станів  $(s, t)$ , яка належить  $R$ , говориться, що можливий перехід від стану  $s$  до стану  $t$ .

– Модель відношення, яка використовується в контексті логіки для визначення, які стани "доступні" з даного середовища  $W \subseteq S \times 2^S$ . Зазвичай визначається як  $W = \{(s, \{t \in S \mid (s, t) \in R\}) \mid s \in S\}$ .

– Властивості станів  $\Phi$ . Кожний стан може мати свої властивості, які визначають його характеристики.

– Семантику моделі. Визначає, які властивості та формули виконуються в кожному світі. Властивості та формули визначаються в термінах відношення переходів і моделі відношення.

Тепер визначимо деякі властивості взаємодії агентів та їхнього середовища.

*Властивості агента повідомлення  $Ag_{MS}$ :*

- *M*: Агент отримує та розповсюджує повідомлення.
- *G*: Чому щось у майбутньому (завжди).

Формула темпоральної логіки: *MLGM*

*Властивості агента керування пристроями  $Ag_{DC}$ :*

- *C*: Агент керує підключеними пристроями.
- *F*: В майбутньому (хоча б раз).

Формула темпоральної логіки: *CFC*

*Властивості агента камери  $Ag_{CM}$ :*

- *V*: Агент виявляє рух на відео.
- *X*: У наступному кроці часу.

Формула темпоральної логіки: *VXV*

*Властивості агента сигналізації  $Ag_{AI}$ :*

- *A*: Агент активує систему сигналізації.
- *G*: Чому щось у майбутньому (завжди).

Формула темпоральної логіки: *AGA*

*Властивості агента годування  $A_{Fd}$ :*

- *F*: Агент моніторить та автоматизує систему харчування.
- *X*: У наступному кроці часу.

Формула темпоральної логіки: *FXF*

*Властивості агента медичного догляду  $A_{MS}$ :*

- *H*: Агент відстежує медичні дані та визначає стан здоров'я.
- *U*: До того моменту, коли стан здоров'я стабілізується.

Формула темпоральної логіки:  $HUN$

*Властивості середовища  $E$ :*

*I*: Середовище інтегровано з медичними пристроями.

*G*: Чому щось у майбутньому (завжди).

Формула темпоральної логіки:  $IGI$

Тоді перехід від стану  $s_0$  до стану  $s_1$  може бути записаний формулою  $MLGM$  або "Агент повідомлення отримує та розповсюджує повідомлення завжди":

$$s_0 \rightarrow s_1, \quad (4.2)$$

де  $s_0$  – стан, в якому агент повідомлення отримав інформацію;

$s_1$  – стан, в якому агент повідомлення розповсюджує повідомлення.

Формула  $M$  виконується в стані  $s_0$ , оскільки агент отримав інформацію.

Формула  $GM$  виконується в усіх станах, оскільки агент завжди розповсюджує інформацію.

Тоді ми можемо визначити стан системи  $S$  як кортеж, що включає стани кожного агента та ресурсу:

$$s_i = (Ag_{DC}, Ag_{MS}, E_{information}). \quad (4.3)$$

Далі, визначимо відношення переходу між станами в системі. Наприклад, якщо агент  $Ag_{DC}$  відправляє повідомлення про недостатнє освітлення і агент  $Ag_{MS}$  отримує це повідомлення та пересилає його для обробки на хмарні ресурси  $E_{information}$ , то можна визначити відношення переходу:

$$(Ag_{DC}, Ag_{MS}, E_{information}) \rightarrow (Ag'_{DC}, Ag'_{MS}, E'_{information})$$

де  $Ag'_{DC}, Ag'_{MS}$  та  $E'_{information}$  представляють оновлені стани агентів та ресурсу після вказаних подій.

Така модель дозволяє формально визначити та відслідковувати зміни станів системи в результаті подій та взаємодії агентів.

Програмна реалізація взаємодії агента керування пристроями ( $Ag_{DC}$ ), який відправляє повідомлення про недостатнє освітлення, з агентом повідомлення ( $Ag_{MS}$ ), що отримує це повідомлення та пересилає його для обробки на хмарні ресурси ( $E_{information}$ ) приведена в лістингу 4.1:

#### Лістинг 4.1

```
class KripkeModel:
    def __init__(self, states, transitions):
        self.states = states
        self.transitions = transitions

class DeviceControlAgent:
    def __init__(self, name):
        self.name = name

    def send_insufficient_light_message(self):
        return f"{self.name} sent a message about insufficient
light."

class MessagingAgent:
    def __init__(self, name):
        self.name = name

    def receive_message(self, message):
        print(f"{self.name} received a message: {message}")
        return message

    def forward_message_to_cloud(self, message, cloud_agent):
        print(f"{self.name} forwarded the message to cloud.")
        cloud_agent.process_message(message)

class CloudAgent:
    def __init__(self, name):
        self.name = name

    def process_message(self, message):
        print(f"{self.name} processed the message: {message}")

# Створення моделі Крипке з двома станами
states = {'State0', 'State1'}
transitions = {'State0': ['State1'], 'State1': []}
model = KripkeModel(states, transitions)

# Створення агентів
device_control_agent = DeviceControlAgent('AgDC')
```

```

messaging_agent = MessagingAgent('AgMs')
cloud_agent = CloudAgent('E_information')

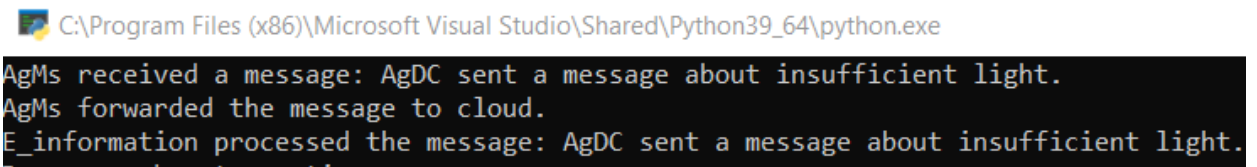
# Перехід від одного стану до іншого та виклик функцій агентів
for state in model.states:
    # Агент керування пристроями відправляє повідомлення
    message =
device_control_agent.send_insufficient_light_message()

    # Агент повідомлення отримує повідомлення та пересилає на
хмарні ресурси
    forwarded_message = messaging_agent.receive_message(message)
    messaging_agent.forward_message_to_cloud(forwarded_message,
cloud_agent)

    next_states = model.transitions.get(state, [])
    if not next_states:
        break

```

Таким чином, створюється взаємодія між агентами.



```

C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python39_64\python.exe
AgMs received a message: AgDC sent a message about insufficient light.
AgMs forwarded the message to cloud.
E_information processed the message: AgDC sent a message about insufficient light.

```

Рисунок 4.4 – Передача згенерованого повідомлення на обробку

#### 4.3.1 Властивості агента повідомлень

Поведінка агента повідомлень визначається таким чином: агент повідомлень  $Ag_{MS}$  отримує повідомлення від інших агентів або датчиків у системі; визначає тип повідомлення та його важливості для подальшого визначення реакції; пересилає важливі повідомлення іншим агентам для активного реагування на ситуації та надсилає повідомлення користувачам, включаючи інструкції та рекомендації. Загальний опис діаграми діяльності агента  $Ag_{MS}$  наведена на рисунку 4.5.

$A_{MS}$ . Алгоритм розповсюдження повідомлень для агента  $Ag_{MS}$  в мультиагентній системі для забезпечення комфорту та безпеки в розумних будинках спроектований для ефективної комунікації між агентами та

оперативного реагування на події (Рис 4.6).

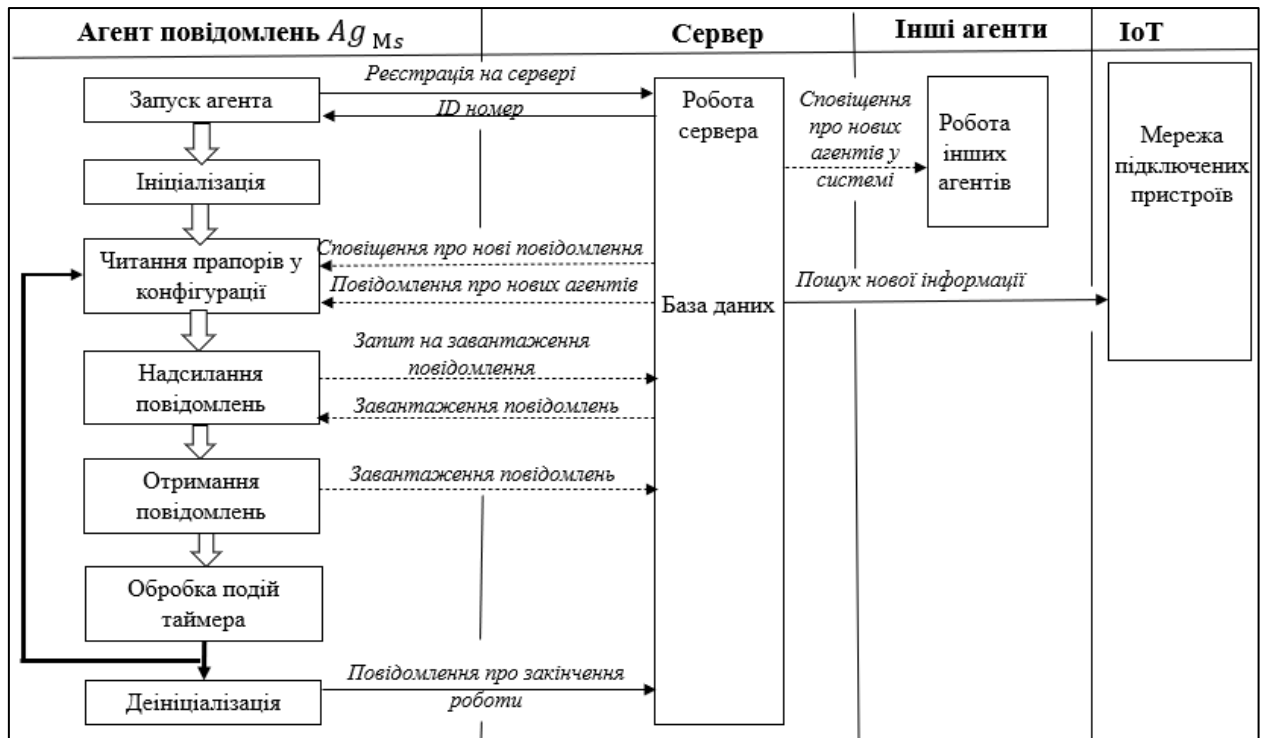


Рисунок 4.5 – Діаграма діяльності агента  $Ag_{Ms}$

Це загальний алгоритм, в якому на першому етапі агент повідомлення  $Ag_{Ms}$  починає свою роботу з визначення свого початкового стану, включаючи його готовність приймати та розповсюджувати повідомлення.

На другому етапі агент  $Ag_{Ms}$  постійно спостерігає за змінами у середовищі та власних внутрішніх параметрах, які можуть вплинути на комфорт чи безпеку.

В разі виявлення події, яка потребує уваги (наприклад, зміна температури, рух на об'єкті, виявлення витoku газу), на третьому етапі агент  $Ag_{Ms}$  генерує повідомлення про цю подію.

Перед відправленням на четвертому етапі агент  $Ag_{Ms}$  форматує повідомлення, включаючи інформацію про подію, її важливість та контекст.

На п'ятому етапі агент  $Ag_{Ms}$  визначає цілі розповсюдження повідомлення. Ці цілі можуть включати інших агентів безпеки, агентів керування пристроями або хмарні ресурси для додаткового аналізу.



Рисунок 4.6 – Алгоритм розповсюдження повідомлень для агента  $Ag_{Ms}$

На шостому етапі агент  $Ag_{Ms}$  використовує механізми розповсюдження, такі як мережеві протоколи, щоб ефективно відправити повідомлення до визначених цілей.

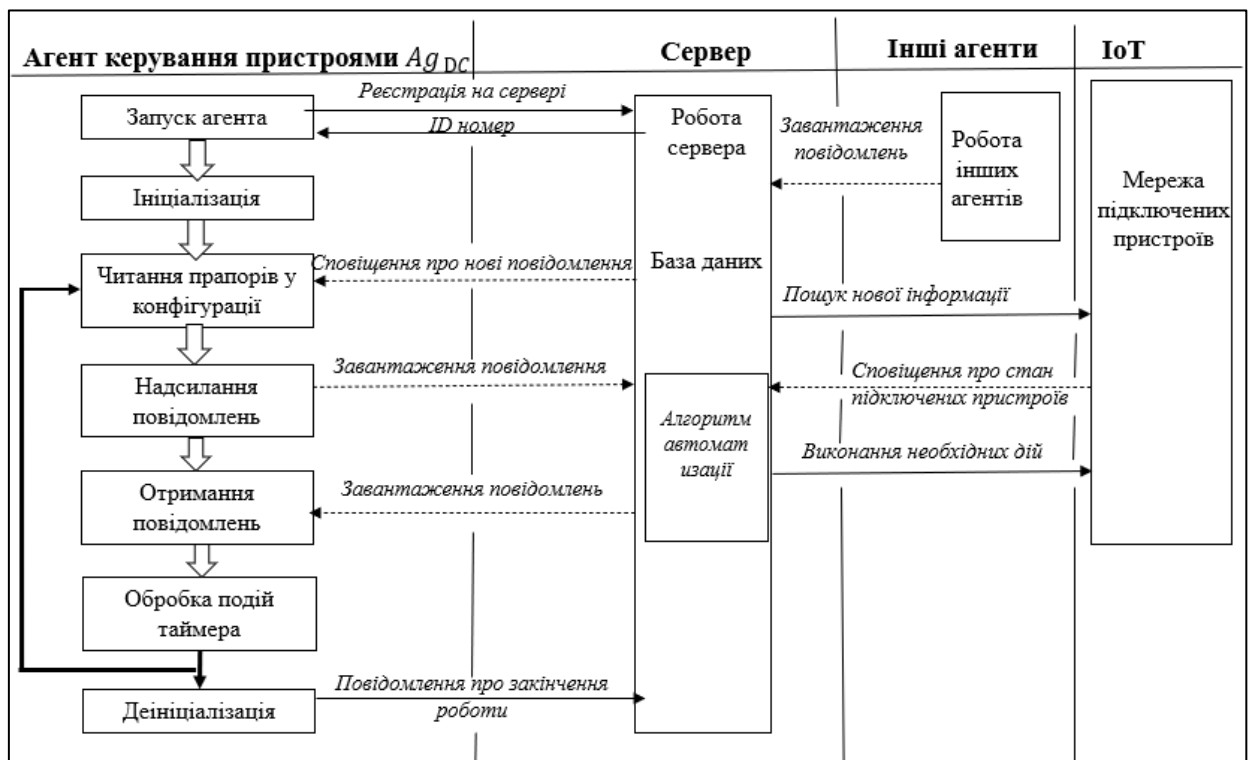
При отриманні повідомлення на сьомому етапі цільові агенти підтверджують його отримання та вживають заходів для подальшої обробки події або реагування на неї.

На восьмому етапі агент  $Ag_{Ms}$  оновлює свій внутрішній стан на основі отриманої інформації та готується до подальших можливих подій.

Розробка ефективного алгоритму розповсюдження повідомлень є ключовою частиною створення мультиагентної системи для забезпечення комфорту та безпеки в розумних будинках.

### 4.3.2 Властивості агента керування пристроями

Поведінка агента керування пристроями  $Ag_{DC}$  визначається таким чином: агент отримує запити та команди від користувачів або інших агентів; визначає призначення запитань та команд для подальшого виконання; взаємодіє з пристроями (освітлення, опалення, системи безпеки тощо) для виконання необхідних дій; відправляє повідомлення про виконання команди або стан пристроїв. Загальний опис діаграми діяльності агента  $Ag_{DC}$  наведена на рисунку 4.6.



Рисунк 4.7 – Діаграма діяльності агента  $Ag_{DC}$

$A_{DC}$ . Алгоритм автоматизації для агента керування пристроями  $Ag_{DC}$  в мультиагентній системі для розумного будинку спроектований для ефективного управління підключеними пристроями та оптимізації їхньої роботи для забезпечення комфорту та безпеки (Рис. 4.8).

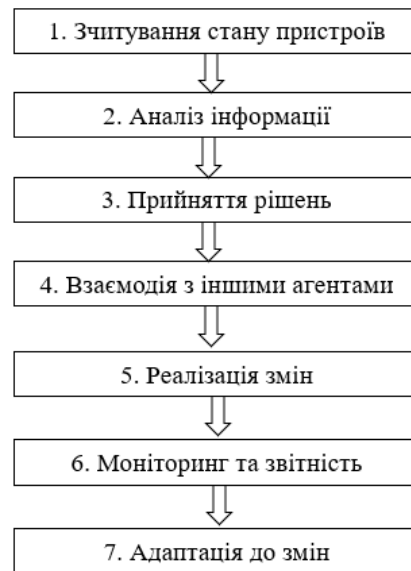


Рисунок 4.8 – Алгоритм для агента керування пристроями  $Ag_{DC}$

Етап 1. Агент керування  $Ag_{DC}$  пристроями регулярно зчитує стан підключених пристроїв та отримує дані про їхню роботу, ефективність та стан.

Етап 2. Агент  $Ag_{DC}$  аналізує зчитану інформацію та враховує фактори, такі як поточні умови середовища, графік користувачів, попередній досвід управління та важливість кожного пристрою.

Етап 3. На основі аналізу агент  $Ag_{DC}$  приймає рішення про те, як краще оптимізувати роботу підключених пристроїв. Це може включати в себе зміну параметрів пристроїв, включення/виключення, регулювання режимів роботи тощо

Етап 4. У випадку необхідності агент  $Ag_{DC}$  може обмінюватися інформацією та координувати свої дії з іншими агентами системи, такими як агент повідомлення чи агент безпеки

Етап 5. Агент  $Ag_{DC}$  виконує необхідні команди для зміни режимів роботи пристроїв відповідно до прийнятих рішень. Це може включати виклик API пристроїв, надсилання команд через мережу тощо

Етап 6. Агент  $Ag_{DC}$  продовжує моніторити реакцію пристроїв на внесені зміни та генерує звіти про їх ефективність та стан для подальшого аналізу.

Етап 7. Враховуючи зміни у середовищі, розкладі користувачів або в стані пристроїв, агент  $Ag_{DC}$  автоматично адаптує свої стратегії та рішення.

Взаємодія агента повідомлень з агентом керування пристроями складається з таких кроків:

Крок 1. Агент повідомлення отримує повідомлення від інших агентів або датчиків.

Крок 2. Агент повідомлення робить аналіз повідомлення, тобто визначає тип повідомлення та його важливість для подальшого визначення реакції.

Крок 3. Агент повідомлення передає інформацію агенту керування пристроями про отримане повідомлення.

Крок 4. Агент керування пристроями визначає необхідні дії для керування пристроями на основі отриманого повідомлення.

Крок 5. Агент керування пристроями виконує визначені дії для керування пристроями.

Крок 6. Агент керування пристроями повертає результат агенту повідомлення, відправляючи звіт про виконані дії.

Крок 7. Агент керування пристроями повертається до агента повідомлення, і, можливо, сповіщає про успішну взаємодію.

#### 4.3.3 Властивості агента камери

Агент камери здійснює постійний моніторинг визначених об'єктів або зон за допомогою камер; визначає рух в зоні відображення камери та виклик події у разі виявлення та зберігає знімки або відео в разі спрацювання сенсорів чи за іншими визначеними умовами.

$A_{Cm}$ . Алгоритми виявлення руху для агента камери  $Ag_{Cm}$  в мультиагентній системі для розумного будинку дозволяють вчасно реагувати на події та важливі зміни в середовищі для забезпечення безпеки та комфорту (Рис. 4.8).

Етап 1. Агент камери  $Ag_{cm}$  захоплює відеопотік зі своєї області моніторингу. Ініціалізація параметрів та налаштувань, таких як чутливість до руху, розмір області моніторингу, пороги виявлення та інші.

Етап 2. Відеопотік конвертується в градації сірого для спрощення обробки та зменшення обсягу даних

Етап 3. Послідовна обробка кадрів відеопотоку для виявлення змін в області моніторингу. За допомогою алгоритму визначення різниці (наприклад, методу абсолютної різниці), агент  $Ag_{cm}$  визначає, де відбулися зміни між поточним та попереднім кадрами

Етап 4. Застосування методів фільтрації та видалення шуму для отримання чіткіших результатів

Етап 5. Використання бінаризації для виділення областей інтересу та знаходження контурів виявлених об'єктів.

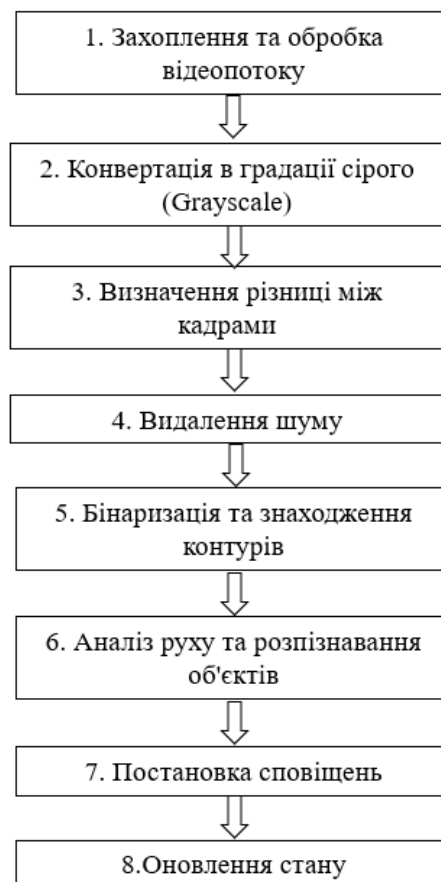


Рисунок 4.9 – Алгоритми виявлення руху для агента камери  $Ag_{cm}$

Етап 6. Визначення характеристик руху (швидкість, напрямок) та можлива класифікація об'єктів (людини, тварини, інше). Використання алгоритмів компарації кадрів, як отримання різниці або використання оптичного потоку для визначення областей, де відбувається рух.

Етап 7. Визначення та відстеження об'єктів, що спричиняють рух, на протязі кількох кадрів для стабільності та точності виявлення. В залежності від результатів аналізу руху, агент  $Ag_{cm}$  може генерувати сповіщення для інших агентів системи (наприклад, агента повідомлення або агента безпеки).

Етап 8. Агент  $Ag_{cm}$  оновлює свій внутрішній стан на основі виявлених об'єктів та руху, готуючись до подальших подій.

#### 4.3.4 Властивості агента сигналізації

Агент сигналізації здійснює моніторинг параметрів безпеки, таких як дим, витоки газу, проникнення, за допомогою датчиків; виявляє потенційно небезпечних ситуацій на основі отриманих даних; активує сирени, робить сповіщення користувачів та інших агентів у випадку аварійної ситуації.

$A_{AI}$ . *Алгоритми реакції на події агента сигналізації  $Ag_{AI}$  в мультиагентній системі для розумного будинку спроектовані для ефективного виявлення потенційних небезпек та негайного реагування на них. (Рис. 4.10).*

Етап 1. Агент сигналізації  $Ag_{AI}$  постійно моніторить оточуюче середовище та отримує інформацію від інших агентів системи.

Етап 2. Застосування алгоритмів для виявлення потенційно небезпечних ситуацій, таких як вторгнення, дим, витоки газу чи виявлення нестандартної активності.

Етап 3. Оцінка рівня небезпеки на основі характеристик виявленої події та контексту оточуючого середовища.

Етап 4. У випадку виявлення серйозної небезпеки, агент  $Ag_{AI}$  генерує сигнал аварії, який може включати сирени, візуальні сигнали та повідомлення для користувачів.

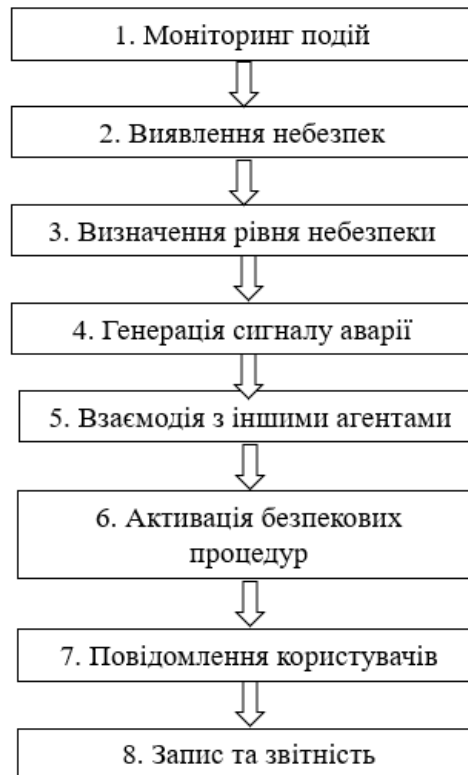


Рисунок 4.10 – Алгоритми реакції на події агента сигналізації  $Ag_{A1}$

Етап 1. Агент сигналізації  $Ag_{A1}$  постійно моніторить оточуюче середовище та отримує інформацію від інших агентів системи.

Етап 2. Застосування алгоритмів для виявлення потенційно небезпечних ситуацій, таких як вторгнення, дим, витоки газу чи виявлення нестандартної активності.

Етап 3. Оцінка рівня небезпеки на основі характеристик виявленої події та контексту оточуючого середовища.

Етап 4. У випадку виявлення серйозної небезпеки, агент  $Ag_{A1}$  генерує сигнал аварії, який може включати сирени, візуальні сигнали та повідомлення для користувачів.

Етап 5. Обмін інформацією та координування дій з іншими агентами системи, такими як агенти безпеки чи агенти керування пристроями.

Етап 6. Запуск безпекових процедур та автоматичне виконання дій для забезпечення безпеки, таких як блокування дверей, виклик служб екстреної допомоги, перегляд відео тощо.

Етап 7. Інформування користувачів про виниклу небезпеку, рекомендації та інструкції щодо безпечних дій.

Етап 8. Запис подій та реакцій для подальшого аналізу, вдосконалення алгоритмів та створення звітів для користувачів чи служб безпеки.

Головною метою цього алгоритму є ефективна реакція на події, що можуть вплинути на безпеку та комфорт в розумному будинку.

#### 4.3.5 Властивості агента годування

Агент годування здійснює спостереження за процесом приготування їжі та споживанням користувачів; аналізує дієтичні обмеження, алергії та інші фактори, що впливають на харчові вимоги користувачів та забезпечує оптимальний харчовий режим відповідно до специфікацій та звичок користувачів

$A_{Fd}$  Алгоритми регулювання харчування агента годування  $Ag_{Fd}$  в мультиагентній системі для розумного будинку спроектовані для оптимізації раціону та забезпечення комфорту та безпеки в житті користувачів. (Рис. 4.11).

Етап 1. Агент годування  $Ag_{Fd}$  отримує інформацію про харчування користувачів, включаючи їхні вподобання, алергії, медичні обмеження та інші фактори, які впливають на харчові потреби.

Етап 2. Враховування звичок, графіка та індивідуальних харчових потреб користувачів для персоналізованого підходу до регулювання харчування.

Етап 3. Розробка алгоритму для визначення оптимального та комфортного раціону, який враховує дієтичні потреби, енергетичний баланс та інші фактори.

Етап 4. Взаємодія з іншими агентами системи, такими як агенти медичного догляду, агенти керування пристроями та інші, для оптимізації харчового режиму.



Рисунок 4.11 – Алгоритми регулювання харчування агента годування  $Ag_{Fd}$

Етап 5. Моніторинг харчування користувачів та здійснення контролю за споживанням у відповідності до рекомендацій та медичних обмежень.

Етап 6. Генерація рекомендацій для користувачів щодо оптимального харчування, включаючи порції, розподіл поживних речовин, час прийому їжі та інше.

Етап 7. Автоматичне внесення корекцій до раціону в залежності від змін у фізичній активності, стані здоров'я та інших факторів.

Етап 8. Систематичне відстеження харчового режиму та генерація звітів про прогрес для користувачів та інших агентів системи.

Головною метою є забезпечення здорового та комфортного харчування у розумному будинку.

#### 4.3.6 Властивості агента контролю клімату

Агент контролю клімату здійснює моніторинг температури, вологості, рівня CO<sub>2</sub> та інших параметрів; визначає комфортні параметри та виявлення необхідних корекцій та керує системами опалення та кондиціонування для досягнення комфортних умов.

*A<sub>cl</sub>* Алгоритми клімат контролю агента контролю клімату *Ag<sub>cl</sub>* в мультиагентній системі для розумного будинку спроектовані для оптимального управління температурою, вологістю та іншими параметрами середовища в приміщенні (Рис. 4.12).

Етап 1. Агент контролю клімату отримує дані про температуру, вологість, рівень CO<sub>2</sub> та інші параметри середовища в реальному часі.

Етап 2. Врахування встановлених стандартів та особливостей користувачів для визначення оптимального комфортного клімату в приміщенні.

Етап 3. Розробка режимів роботи, які враховують різні часи доби, ділові події, звички користувачів та інші фактори.

Етап 4. Автоматичне регулювання систем опалення та кондиціонування повітря в залежності від змін зовнішніх та внутрішніх умов.

Етап 5. Врахування показників витрат енергії та ефективності систем опалення та кондиціонування для оптимізації витрат та зменшення екологічного впливу.

Етап 6. Координація дій з іншими агентами системи, такими як агенти керування пристроями, для оптимального використання ресурсів та забезпечення комфортного клімату.

Етап 7. Моніторинг та аналіз зручності кліматичних умов, а також забезпечення безпеки в разі можливих проблем (наприклад, перегрів чи переохолодження).

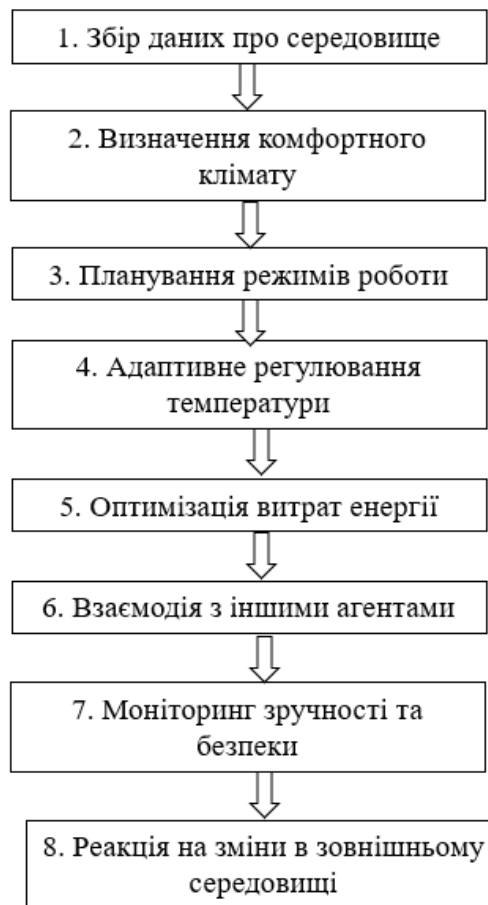


Рисунок 4.12 – Алгоритми клімат контролю агента контролю клімату  $Ag_{cl}$

Етап 8. Автоматична адаптація до змін в зовнішньому середовищі, таких як зміни температури, погодні умови тощо. Збір даних та генерація звітів про стан клімату та рекомендацій для оптимізації та підвищення комфортності.

Головною метою цього алгоритму є надання комфортного та ефективного регулювання клімату в розумному будинку.

#### 4.3.7 Властивості агента медичного догляду

Агент медичного догляду здійснює отримання медичних даних від різних датчиків та засобів моніторингу; виявляє та аналізує показники здоров'я для діагностики стану користувача та надає рекомендації щодо прийому ліків, фізичної активності, змін у режимі дня тощо.

$Ag_{MC}$  Алгоритми аналізу стану здоров'я агента медичного догляду  $Ag_{MC}$  в мультиагентній системі для розумного будинку спроектовані для моніторингу фізичного та психічного стану користувачів. (Рис. 4.13).



Рисунок 4.13 – Алгоритми аналізу стану здоров'я агента медичного догляду  $Ag_{MC}$

Етап 1. Агент медичного догляду отримує дані про здоров'я користувачів, такі як вимірювання температури, пульсу, тиску, рівня цукру в крові, дані з датчиків фізичної активності та інші параметри.

Етап 2. Аналіз змін у фізіологічних показниках, таких як пульс, тиск, температура, щоб вчасно виявити можливі аномалії та тривожні симптоми.

Етап 3. Використання даних з датчиків для аналізу емоційного стану користувача та виявлення ознак стресу або депресії.

Етап 4. Комунікація та обмін інформацією з іншими агентами, такими як агенти контролю клімату та безпеки, для врахування здоров'я в загальній

системі управління будинком.

Етап 5. Відстеження графіку прийому ліків та моніторинг реакцій на лікування.

Етап 6. Використання аналітики для прогнозування можливих проблем здоров'я та надання рекомендацій для їх попередження.

Етап 7. Негайне виявлення та реагування на критичні стани здоров'я користувачів, включаючи виклик служб екстреної допомоги

Етап 8. Забезпечення користувачів інформацією про медичні ресурси та можливості консультацій з лікарем.

#### 4.4 Приклад реалізації мультиагентної системи для забезпечення комфорту та безпеки у розумних

Забезпечення комфорту в розумному будинку за допомогою мультиагентної системи для забезпечення комфорту та безпеки у розумних при падінні температури до 15 градусів може включати в себе різні аспекти, від автоматичного керування опалюванням до сповіщення користувачів та координації із суміжними системами (Рис. 4.14).

Алгоритм складається з таких кроків (Лістинг 4.1):

Крок 1. Виявлення зниження температури: Датчики температури в системі виявляють падіння температури до 15 градусів.

Крок 2. Система керування опаленням: Агент контролю клімату аналізує дані про падіння температури. Запускає опалювальні системи для підвищення температури до комфортного рівня.

Крок 3. Сповіщення користувача: Агент повідомлення відправляє повідомлення користувачу про падіння температури та дії, які були вжиті.

Крок 4. Координація з іншими системами: Система може координуватися з іншими агентами, такими як агент керування пристроями, для оптимізації енергоспоживання та комфорту.

Крок 5. Оптимізація роботи системи: Агент керування пристроями може

автоматично взаємодіяти з опалювальними пристроями, забезпечуючи оптимальний режим опалення.

Крок 6. Відновлення комфорту: Система продовжує контролювати температуру та вживає заходів для її підтримки на комфортному рівні.

Крок 7. Моніторинг: Система веде моніторинг стану та може генерувати звіти для подальшого аналізу та вдосконалення роботи.

#### Лістинг 4.2:

```
import tkinter as tk
from datetime import datetime

# Клас агента контролю клімату
class ClimateControlAgent:
    def send_Message(sender, message, current_temperature,
target_temperature, log_field):
        formatted_current_temperature = (
            f"{datetime.now().strftime('%Y-%m-%d %H:%M:%S')} -
{sender} to All: {message}\n"
            f"Current Temperature: {current_temperature}"
        )
    def __init__(self, agent_name, log_field):
        self.agent_name = agent_name
        self.target_temperature = 20
        self.current_temperature = 18
        self.log_field = log_field

    def adjust_temperature(self, new_temperature):
        self.current_temperature = new_temperature

    def check_temperature(self):
        if self.current_temperature < 15:
            self.adjust_temperature(self.target_temperature)
            MessageAgent.send_Message(self.agent_name, "The
temperature in the house is lowered. I turn on the heating.",
self.current_temperature, self.target_temperature,
self.log_field)

# Клас агента повідомлення
class MessageAgent:
    @staticmethod
    def send_Message(sender, message, current_temperature,
target_temperature, log_field):
        formatted_current_temperature = (
            f"{datetime.now().strftime('%Y-%m-%d %H:%M:%S')} -
{sender} to All: {message}\n"
```

```

        f"Current Temperature: {current_temperature}"
    )

    formatted_target_temperature = (
        f"{datetime.now().strftime('%Y-%m-%d %H:%M:%S')} -
{sender} to All: {message}\n"
        f"Target Temperature: {target_temperature}"
    )

    log_message =
f"{formatted_current_temperature}\n{formatted_target_temperature
}"

    log_field.config(state=tk.NORMAL)
    log_field.insert(tk.END, log_message + "\n")
    log_field.config(state=tk.DISABLED)
    log_field.see(tk.END)

# Клас агента керування пристроями
class DeviceControlAgent:
    @staticmethod
    def control_heating_system(sender, action, log_field):
        formatted_message = f"{datetime.now().strftime('%Y-%m-%d
%H:%M:%S')} - {sender} to Heating System: {action}"
        log_message = f"{formatted_message}"
        log_field.config(state=tk.NORMAL)
        log_field.insert(tk.END, log_message + "\n")
        log_field.config(state=tk.DISABLED)
        log_field.see(tk.END)

# Симуляція зміни температури
def simulate_temperature_change():
    climate_agent_label = tk.Label(window, text="i ")
    climate_agent_label.pack()
    climate_agent_log = tk.Text(window, height=5, width=50,
wrap=tk.WORD, font=('Helvetica', 10), state=tk.DISABLED)
    climate_agent_log.pack(pady=5)
    climate_agent = ClimateControlAgent("Current temperature",
climate_agent_log)

    Message_agent_label = tk.Label(window, text="Message Agent")
    Message_agent_label.pack()
    Message_agent_log = tk.Text(window, height=5, width=50,
wrap=tk.WORD, font=('Helvetica', 10), state=tk.DISABLED)
    Message_agent_log.pack(pady=5)
    MessageAgent.send_Message("Message Agent", "Test Message",
15, 22, Message_agent_log)

    device_agent_label = tk.Label(window, text="Device Agent")
    device_agent_label.pack()
    device_agent_log = tk.Text(window, height=5, width=50,
wrap=tk.WORD, font=('Helvetica', 10), state=tk.DISABLED)
    device_agent_log.pack(pady=5)

```

```

DeviceControlAgent.control_heating_system("Device Agent",
"Test Action", device_agent_log)

# Створення головного вікна
window = tk.Tk()
window.title("Smart Home Simulation")

# Симуляція при натисканні кнопки
simulate_button = tk.Button(window, text="Simulate",
command=simulate_temperature_change)
simulate_button.pack(pady=10)

# Запуск головного циклу
window.mainloop()

```



Рисунок 4.14 – Вікно для взаємодії агентів

Цей підхід дозволяє мультиагентній системі ефективно взаємодіяти та реагувати на зміни у середовищі з метою забезпечення комфорту та оптимального управління системами будинку.

## ВИСНОВКИ

У цій магістерській роботі було проведено глибокий аналіз можливостей та викликів, пов'язаних з інтеграцією мультиагентних систем у контексті розумних будинків. Було визначено, що мультиагентні системи пропонують значний потенціал для оптимізації процесів управління в розумних будинках, покращення енергоефективності, безпеки та зручності для мешканців. Основним внеском роботи є розробка моделей взаємодії між агентами, які дозволяють ефективно координувати роботу різних систем і пристроїв у розумному будинку.

Окрім того, були досліджені різні стратегії та алгоритми, які дозволяють агентам реагувати на змінні умови та потреби мешканців. Це дозволило створити гнучкі та адаптивні системи, які можуть ефективно вирішувати задачі автоматизації та контролю в домашньому середовищі. На основі проведених досліджень було розроблено прототип мультиагентної системи для забезпечення комфорту та безпеки для розумного будинку.

Результати дослідження підкреслюють важливість подальшого розвитку та впровадження мультиагентних систем у розумних будинках, що може мати значний вплив на покращення якості життя людей та ефективності управління домогосподарствами. Таким чином, ця робота вносить цінний вклад у розвиток смарт-технологій та їх застосування в повсякденному житті.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Jouini, M., Rabai, L. B. A., & Aissa, A. B. (2014). Classification of security threats in information systems. *Procedia Computer Science*, 32, 489-496.
2. Fonseca, T., Dias, T., Vitorino, J., Ferreira, L. L., & Praça, I. (2022). A Low-Cost Multi-Agent System for Physical Security in Smart Buildings. *arXiv preprint arXiv:2209.00741*.
3. Brushlinsky, N. N., Ahrens, M., Sokolov, S. V., & Wagner, P. World of Fire Statistics= Мировая пожарная статистика= Die Feuerwehrstatistik der Welt: Report= отчет= Bericht [Electronic resource]/Center of Fire Statistics of CTIF.[S. l.]. 2016. N, 21, 61.
4. Hassan, W. H. (2019). Current research on Internet of Things (IoT) security: A survey. *Computer networks*, 148, 283-294.
5. Berry, D., Usmani, A., Torero, J. L., Tate, A., McLaughlin, S., Potter, S., ... & Atkinson, M. (2005, September). FireGrid: Integrated emergency response and fire safety engineering for the future built environment. UK e-Science Programme All Hands Meeting.”
6. Çetin, A. E., Dimitropoulos, K., Gouverneur, B., Grammalidis, N., Günay, O., Habiboğlu, Y. H., ... & Verstockt, S. (2013). Video fire detection–review. *Digital Signal Processing*, 23(6), 1827-1843.
7. Harshika, G., Haani, U., Bhuvaneshwari, P., & Venkatesh, K. R. (2022). Smart Pet Insights System Based on IoT and ML. In *Innovative Data Communication Technologies and Application: Proceedings of ICIDCA 2021* (pp. 725-738). Singapore: Springer Nature Singapore.
8. Chen, Y., & Elshakankiri, M. (2020, April). Implementation of an IoT based pet care system. In *2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC)* (pp. 256-262). IEEE.

9. Qiao, B., Liu, K., & Guy, C. (2006, December). A multi-agent system for building control. In *2006 IEEE/WIC/ACM International Conference on Intelligent Agent Technology* (pp. 653-659). IEEE.
10. Kim, S. (2016). Smart pet care system using internet of things. *International Journal of Smart Home*, 10(3), 211-218.
11. McKinsey Digital blue. IoT value set to accelerate through <https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/iot-value-set-to-accelerate-through-2030-where-and-how-to-capture-it> (дата звернення 29.12.23).
12. Revenue of the smart home market worldwide from 2019 to 2028. <https://www.statista.com/forecasts/887554/revenue-in-the-smart-home-market-in-the-world> (дата звернення 29.12.23).
13. Smart From The Start. 'Smart Home' Is Becoming 'Sustainable Home' <https://www.forbes.com/sites/johnkoetsier/2021/11/23/smart-from-the-start-smart-home-is-becoming-sustainable-home>. (дата звернення 03.01.24).
14. Brushlinsky, N. N., Ahrens, M., Sokolov, S. V., & Wagner, P. World of Fire Statistics Die Feuerwehrstatistik der Welt: Report Bericht [Electronic resource]/Center of Fire Statistics of CTIF.[S. l.]. 2016. N, 21, 61.
15. Harshika, G., Haani, U., Bhuvaneshwari, P., & Venkatesh, K. R. (2022). Smart Pet Insights System Based on IoT and ML. In *Innovative Data Communication Technologies and Application: Proceedings of ICIDCA 2021* (pp. 725-738). Singapore: Springer Nature Singapore.
16. Hassan, W. H. (2019). Current research on Internet of Things (IoT) security: A survey. *Computer networks*, 148, 283-294.
17. Samad, T., & Frank, B. (2007, July). Leveraging the web: A universal framework for building automation. In *2007 American Control Conference* (pp. 4382-4387). IEEE.
18. Dodier, R. H., Henze, G. P., Tiller, D. K., & Guo, X. (2006). Building occupancy detection through sensor belief networks. *Energy and buildings*, 38(9), 1033-1043.

19. Mukhopadhyay, S. C., & Suryadevara, N. K. (2014). Internet of things: Challenges and opportunities (pp. 1-17). Springer International Publishing.
20. Roscia, M., Longo, M., & Lazaroiu, G. C. (2013, October). Smart City by multi-agent systems. In 2013 International Conference on Renewable Energy Research and Applications (ICRERA) (pp. 371-376). IEEE.
21. Dorri, A., Kanhere, S. S., & Jurdak, R. (2018). Multi-agent systems: A survey. *Ieee Access*, 6, 28573-28593.
22. do Nascimento, N. M., & de Lucena, C. J. P. (2017). Fiot: An agent-based framework for self-adaptive and self-organizing applications based on the internet of things. *Information Sciences*, 378, 161-176.
23. Forestiero, A. (2017, May). Multi-agent recommendation system in Internet of Things. In 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID) (pp. 772-775). IEEE.
24. Krupitzer, C., Roth, F. M., VanSyckel, S., Schiele, G., & Becker, C. (2015). A survey on engineering approaches for self-adaptive systems. *Pervasive and Mobile Computing*, 17, 184-206.
25. Weyns, D., Omicini, A., & Odell, J. (2007). Environment as a first class abstraction in multiagent systems. *Autonomous agents and multi-agent systems*, 14, 5-30.
26. Ricci, A., Piunti, M., & Viroli, M. (2011). Environment programming in multi-agent systems: an artifact-based perspective. *Autonomous Agents and Multi-Agent Systems*, 23, 158-192.
27. Amato, C. (2018, July). Decision-Making Under Uncertainty in Multi-Agent and Multi-Robot Systems: Planning and Learning. In IJCAI (pp. 5662-5666).
28. Ma, L., Wang, Z., Han, Q. L., & Liu, Y. (2017). Consensus control of stochastic multi-agent systems: a survey. *Science China Information Sciences*, 60, 1-15.
29. Li, W., Xie, L., & Zhang, J. F. (2015). Containment control of leader-following multi-agent systems with Markovian switching network topologies and measurement noises. *Automatica*, 51, 263-267.

30. Mansour, A. M. (2021). Cooperative Multi-Agent Vehicle-to-Vehicle Wireless Network in a Noisy Environment. *International Journal of Circuits, Systems and Signal Processing*, 15.
31. Khan, A., Zhang, C., Lee, D. D., Kumar, V., & Ribeiro, A. (2018). Scalable centralized deep multi-agent reinforcement learning via policy gradients. *arXiv preprint arXiv:1805.08776*.
32. Huang, D., Jiang, H., Yu, Z., Hu, C., & Fan, X. (2019). Cluster-delay consensus in MASs with layered intermittent communication: A multi-tracking approach. *Nonlinear Dynamics*, 95, 1713-1730.
33. Sayed, A. H. (2014). Adaptive networks. *Proceedings of the IEEE*, 102(4), 460-497.
34. Zhang, C., Lesser, V. R., & Abdallah, S. (2010, May). Self-organization for coordinating decentralized reinforcement learning. In *AAMAS* (Vol. 10, pp. 739-746).
35. Búrdalo, L., Terrasa, A., Julián, V., & García-Fornes, A. (2018). The information flow problem in multi-agent systems. *Engineering Applications of Artificial Intelligence*, 70, 130-141.
36. Jiang, X., Xia, G., & Feng, Z. (2019). Output consensus of high-order linear multi-agent systems with time-varying delays. *IET Control Theory & Applications*, 13(8), 1084-1094.
37. Bacciu, D., Micheli, A., & Podda, M. (2020). Edge-based sequential graph generation with recurrent neural networks. *Neurocomputing*, 416, 177-189.
38. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Philip, S. Y. (2020). A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1), 4-24.
39. Papoudakis, G., & Albrecht, S. V. (2020). Variational autoencoders for opponent modeling in multi-agent systems. *arXiv preprint arXiv:2001.10829*.
40. Rassam, M. A., Zainal, A., & Maarof, M. A. (2013). An adaptive and efficient dimension reduction model for multivariate wireless sensor networks applications. *Applied Soft Computing*, 13(4), 1978-1996.

41. Lin, T., Huh, J., Stauffer, C., Lim, S. N., & Isola, P. (2021). Learning to ground multi-agent communication with autoencoders. *Advances in Neural Information Processing Systems*, 34, 15230-15242.
42. Ding, K., Li, J., & Liu, H. (2019, January). Interactive anomaly detection on attributed networks. In *Proceedings of the twelfth ACM international conference on web search and data mining* (pp. 357-365).
43. Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S. A., Binder, A., ... & Kloft, M. (2018, July). Deep one-class classification. In *International conference on machine learning* (pp. 4393-4402). PMLR.
44. Peng, H., Wang, H., Du, B., Bhuiyan, M. Z. A., Ma, H., Liu, J., ... & Philip, S. Y. (2020). Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting. *Information Sciences*, 521, 277-290.
45. Li, L., Yao, J., Wenliang, L., He, T., Xiao, T., Yan, J., ... & Zhang, Z. (2021). Grin: Generative relation and intention network for multi-agent trajectory prediction. *Advances in Neural Information Processing Systems*, 34, 27107-27118.
46. Ierardi, C., Orihuela, L., & Jurado, I. (2019). Distributed estimation techniques for cyber-physical systems: a systematic review. *Sensors*, 19(21), 4720.
47. Jing, G., Zheng, Y., & Wang, L. (2014). Flocking of multi-agent systems with multiple groups. *International Journal of Control*, 87(12), 2573-2582.
48. Liu, F., Xue, S., Wu, J., Zhou, C., Hu, W., Paris, C., ... & Yu, P. S. (2020). Deep learning for community detection: progress, challenges and opportunities. *arXiv preprint arXiv:2005.08225*.
49. Cheng, Z., Flouvat, F., & Selmaoui-Folcher, N. (2017). Mining recurrent patterns in a dynamic attributed graph. In *Advances in Knowledge Discovery and Data Mining: 21st Pacific-Asia Conference, PAKDD 2017, Jeju, South Korea, May 23-26, 2017, Proceedings, Part II 21* (pp. 631-643). Springer International Publishing.
50. Berry, D., Usmani, A., Torero, J. L., Tate, A., McLaughlin, S., Potter, S., ... & Atkinson, M. (2005, September). FireGrid: Integrated emergency response

and fire safety engineering for the future built environment. UK e-Science Programme All Hands Meeting.”

51. Jouini, M., Rabai, L. B. A., & Aissa, A. B. (2014). Classification of security threats in information systems. *Procedia Computer Science*, 32, 489-496.

52. Fonseca, T., Dias, T., Vitorino, J., Ferreira, L. L., & Praça, I. (2022). A Low-Cost Multi-Agent System for Physical Security in Smart Buildings. *arXiv preprint arXiv:2209.00741*.

53. Çetin, A. E., Dimitropoulos, K., Gouverneur, B., Grammalidis, N., Günay, O., Habiboğlu, Y. H., ... & Verstockt, S. (2013). Video fire detection—review. *Digital Signal Processing*, 23(6), 1827-1843.

54. Chen, Y., & Elshakankiri, M. (2020, April). Implementation of an IoT based pet care system. In *2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC)* (pp. 256-262). IEEE.

55. Qiao, B., Liu, K., & Guy, C. (2006, December). A multi-agent system for building control. In *2006 IEEE/WIC/ACM International Conference on Intelligent Agent Technology* (pp. 653-659). IEEE.

56. Kim, S. (2016). Smart pet care system using internet of things. *International Journal of Smart Home*, 10(3), 211-218.

57. Abras, S., Pesty, S., Ploix, S., & Jacomino, M. (2008, April). An anticipation mechanism for power management in a smart home using multi-agent systems. In *2008 3rd International Conference on Information and Communication Technologies: From Theory to Applications* (pp. 1-6). IEEE.

58. Alam, M. R., Reaz, M. B. I., & Ali, M. A. M. (2012). A review of smart homes—Past, present, and future. *IEEE transactions on systems, man, and cybernetics, part C (applications and reviews)*, 42(6), 1190-1203.

59. González-Briones A, De La Prieta F, Mohamad MS, Omatu S, Corchado JM. Multi-Agent Systems Applications in Energy Optimization Problems: A State-of-the-Art Review. *Energies*. 2018; 11(8):1928.

60. N . Axak, M. Kushnaryov, A.Tatarnykov, The Agent-Based Learning Platform. ICST-2023: XI International Scientific and Practical Conference

“Information Control Systems and Technologies”, September 21-23, 2023, Odessa, Ukraine, Vol-3513, pp. 263-275. ONLINE: <http://ceur-ws.org/Vol-3513/>, URN: urn:nbn:de:0074-3513-6.

61. Василевська О.О. Огляд можливостей мультиагентної системи для забезпечення комфорту та безпеки у розумних будинках. Науковий простір: аналіз, сучасний стан, тренди та перспективи: матеріали III Всеукраїнської студентської наукової конференції, м.Київ, 16 червня, 2023 рік /ГО «Молодіжна наукова ліга».— Вінниця: ГО «Європейська наукова платформа», 2023.с. 128 –131. ISBN 978-617-8126-55-1. DOI10.36074/liga-ukr-16.06.2023.

62. N. Axak and A. Tatarnykov, "The Behavior Model of the Computer User," 2022 IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT), 2022, pp. 458-461, doi: 10.1109/CSIT56902.2022.10000499.

63. Аксак Н.Г., Василевська О.О., Шеліхов Ю.О. Архітектура мультиагентної системи для забезпечення комфорту та безпеки у розумних будинках. The aspects of contemporary scientific research that encompass both theoretical and practical components: матеріали VI Міжнародної науково-практичної конференції, м.Венеція, Італія, 10 січня, 2024 рік.

64. Василевська О.О. Методи кодування зображень і відео для наднизької затримки. Науковий простір: аналіз, сучасний стан, тренди та перспективи: матеріали III Всеукраїнської студентської наукової конференції, м.Київ, 16 червня, 2023 рік /ГО «Молодіжна наукова ліга».— Вінниця: ГО «Європейська наукова платформа», 2023.с. 125 –127. ISBN 978-617-8126-55-1. DOI10.36074/liga-ukr-16.06.2023.