

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління  
(повна назва)

Кафедра електронних обчислювальних машин  
(повна назва)

**АТЕСТАЦІЙНА РОБОТА**  
**Пояснювальна записка**

Рівень вищої освіти другий (магістерський)

Модель та методи моніторингу систем керування  
за допомогою елементів штучного інтелекту

(тема)

Виконав:

студент II курсу, групи СПМ-19-1  
Казьміна Д.Р.  
(прізвище, ініціали)

Спеціальність 123 – Комп'ютерна інженерія  
(код і повна назва спеціальності)

Тип програми освітньо-професійна  
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування  
(повна назва освітньої програми)

Керівник: доц. Ляшенко О.С.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ

(підпис)

Коваленко А.А.

(прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ комп'ютерної інженерії та управління \_\_\_\_\_

Кафедра \_\_\_\_\_ електронних обчислювальних машин \_\_\_\_\_

Рівень вищої освіти \_\_\_\_\_ другий (магістерський) \_\_\_\_\_

Спеціальність \_\_\_\_\_ 123 – Комп'ютерна інженерія \_\_\_\_\_  
(код і повна назва)

Тип програми \_\_\_\_\_ освітньо-професійна \_\_\_\_\_  
(освітньо-професійна або освітньо-наукова)

Освітня програма \_\_\_\_\_ Системне програмування \_\_\_\_\_  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

“ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ**

**НА АТЕСТАЦІЙНУ РОБОТУ**

студентові \_\_\_\_\_ Казьміній Дарині Радіонівні \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи Модель та методи моніторингу систем керування  
за допомогою елементів штучного інтелекту

затверджена наказом по університету від “ 30 ” жовтня 2020 р. № 1486Ст

2. Термін подання студентом роботи до екзаменаційної комісії 14 грудня 2020 р.

3. Вхідні дані до роботи \_\_\_\_\_

SCADA-система зерносушильного комплексу

Напівавтоматичний метод управління

Агентно-орієнтовані системи

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

1. Концепція scada-систем та сучасні методи моніторингу систем керування

2. Агентно-орієнтоване моделювання для використання у методах управління

3. Інтелектуалізація напівавтоматичного методу управління для SCADA-системи на  
прикладі сучасного елеватора

4. Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Демонстраційні матеріали. Плакати – 10 арк. ф. А4

---

---

---

---

---

---

---

---

---

---

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1 )

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
	залишити пустим, якщо кон-		
	сультантом є керівник роботи		

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз проблеми та огляд існуючих рішень	02.11 – 06.11	
2	Вибір технології розробки та інструментальних засобів	06.11 – 13.11	
3	Розробка моделі управління	14.11 – 20.11	
4	Розробка алгоритмів та модулів для покращення напівавтоматичного методу	21.11 – 01.12	
5	Оформлення матеріалів атестаційної роботи	02.12 – 04.12	

Дата видачі завдання 02 листопада 2020 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_  
(підпис)

доц. Ляшенко О.С.  
(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка атестаційної роботи: 84 с., 21 рис., 1 табл., 2 дод., 19 джерел.

SCADA, ІНТЕЛЕКТУАЛІЗАЦІЯ, ІНТЕЛЕКТУАЛЬНИЙ АГЕНТ, АГЕНТНО-ОРІЄНТОВАНЕ МОДЕЛЮВАННЯ.

Метою атестаційної роботи є розробка моделі моніторингу SCADA-системи для сучасного зерносушильного комплексу та покращення напівавтоматичного методу управління SCADA-системами шляхом застосування агентно-орієнтованого підходу та інтелектуалізації.

У ході виконання атестаційної роботи було визначено концепцію SCADA-систем, поняття інтелектуалізації, поняття агентно-орієнтованого моделювання, надано класифікацію інтелектуальних агентів, розроблено модель моніторингу та покращено напівавтоматичний метод.

## ABSTRACT

Master's thesis: 84 pages, 21 figures, 1 tables, 2 appendices, 19 sources.

SCADA, INTELLECTUALIZATION, INTELLECTUAL AGENT,  
AGENT-ORIENTED MODELING.

The major goal of this thesis is to develop a model of SCADA-system monitoring for a modern grain drying complex and to improve the semi-automatic method of SCADA-systems management by applying an agent-oriented approach and intellectualization.

During the attestation work, the concept of SCADA-systems, the concept of intellectualization, the concept of agent-oriented modeling were defined, the classification of intelligent agents was provided, the monitoring model was developed and the semi-automatic method was improved.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ .....	8
ВСТУП .....	9
1 КОНЦЕПЦІЯ SCADA-СИСТЕМ ТА СУЧАСНІ МЕТОДИ МОНІТОРИНГУ СИСТЕМ КЕРУВАННЯ.....	11
1.1 Концепція SCADA-систем .....	11
1.2 Інтелектуалізація SCADA-систем .....	13
1.3 Загальні функції SCADA-систем.....	16
1.4 Методи управління SCADA-системами .....	17
1.4.1 Автоматичний метод управління SCADA-системами .....	17
1.4.2 Напівавтоматичний метод управління SCADA-системами .....	18
2 АГЕНТНО-ОРІЄНТОВАНЕ МОДЕЛЮВАННЯ ДЛЯ ВИКОРИСТАННЯ У МЕТОДАХ УПРАВЛІННЯ .....	20
2.1 Методологія інтелектуальних агентів.....	20
2.1.1 Визначення поняття «інтелектуальний агент» .....	20
2.1.2 Класифікація агентів.....	23
2.2 Особливості агентного моделювання як методу імітаційного моделювання.....	33
2.3 Процедура формування агентної моделі .....	36
Більшість агентних моделей складається з наступних компонентів:.....	37
2.4 Модель акторів .....	38
2.5 Методи агентного моделювання .....	40
2.5.1 Метод «теорії ігор» .....	40
2.5.2 Метод еволюційного програмування.....	41
2.5.3 Метод Монте-Карло.....	45

3 ІНТЕЛЕКТУАЛІЗАЦІЯ НАПІВАВТОМАТИЧНОГО МЕТОДУ УПРАВЛІННЯ ДЛЯ SCADA-СИСТЕМИ НА ПРИКЛАДІ СУЧАСНОГО ЕЛЕВАТОРА.....	47
3.1 Технологічний процес сучасного елеватора .....	47
3.2 Поняття інтелектуалізації стосовно моделей та методів управління.....	49
3.2 Процес інтелектуалізації .....	52
3.3 Модель інтелектуалізації для сучасної SCADA-системи .....	60
3.3.1 Модель інтелектуалізації.....	62
3.3.2 Механіка процесу інтелектуалізації.....	64
3.3.3 Логіка управління .....	66
3.3.4 Інтелектуалізація імітації .....	67
ВИСНОВКИ.....	71
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	72
ДОДАТОК А Графічний матеріал атестаційної роботи .....	74
ДОДАТОК Б ПУБЛІКАЦІЇ ЗА ТЕМОЮ АТЕСТАЦІЙНОЇ РОБОТИ.....	80

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ  
І ТЕРМІНІВ

АОП – агентно-орієнтоване програмування

АРМ – автоматизовані робочі місця

АСУТП – автоматизовані системи управління технологічними процесами

ОПР – особа, що приймає рішення

ШНМ – штучні нейронні мережі

CS – Communication System

GSM – Groupe Special Mobile

HMI – Human-Machine Interface

MTU – Mater Terminal Unit

PLC – Power Line Communication

RTU – Remote Terminal Unit

SCADA – Supervisory Control And Data Acquisition

SOAP – Simple Object Access Protocol

## ВСТУП

Четверта індустріальна революція (Індустрія 4.0) – це перехід на повністю автоматизоване цифрове виробництво, кероване інтелектуальними системами у режимі реального часу в постійній взаємодії із зовнішнім середовищем, що виходить за межі одного підприємства, з перспективою об'єднання в глобальну промислову мережу речей і послуг.

У вузькому сенсі Індустрія 4.0 (Industry 4.0) – це назва одного з 10 проєктів державної Ні-Tech стратегії Німеччини до 2020 року, що описує концепцію розумного виробництва (Smart Manufacturing) на базі глобальної промислової мережі інтернету речей і послуг (Internet of Things and Services) [14] .

У широкому сенсі, Індустрія 4.0 характеризує поточний тренд розвитку автоматизації та обміну даними, який включає в себе кіберфізичні системи, Інтернет Речей і хмарні обчислення та являє собою новий рівень організації виробництва та управління ланцюжком створення вартості протягом усього життєвого циклу продукції, що випускається.

До компонентів Індустрії 4.0 входять:

- елементи Інтернету речей;
- штучний інтелект, машинне навчання і робототехніка;
- хмарні обчислення;
- Big Data;
- кібербезпека;
- інтеграційна система;
- моделювання;
- доповнена реальність.

Багато з цих елементів вже давно і успішно застосовуються на практиці, але саме об'єднання їх в одну цілісну систему дозволить розвинути концепцію «Індустрії 4.0» та забезпечити новий рівень ефективності

виробництва і додатковий дохід за рахунок використання цифрових технологій, формування мережної взаємодії постачальників і партнерів, а також реалізації інноваційних бізнес-моделей [2].

Більша частина даних, що використовуються промисловими рішеннями IoT, надходить з програмного забезпечення автоматизації HMI/SCADA. Технології автоматизації, такі як програмне забезпечення HMI/SCADA, існують вже багато років. Рішення для автоматизації є невід'ємною і ключовою частиною процесу, відкриваючи еру, в якій оперативні дані негайно аналізуються за допомогою алгоритмів штучного інтелекту/машинного навчання. HMI/SCADA дозволяють приймати більш зважені рішення для швидкого реагування [14]. Перетворення оперативних даних в аналітику, яку потім можна використовувати для оптимізації процесів, приносить справжню цінність для бізнесу та науки, перетворюючи автоматизацію у шар, де користувачі можуть будувати свої цифрові перетворення. По суті, автоматизація на даний час є основою стратегії оцифровки компаній .

85% даних, що використовуються аналітичними інструментами, такими як програмне забезпечення управління ефективністю активів або програмне забезпечення управління операціями, надходять з обчислювальні техніки. Таким чином, правильне налаштування рівня автоматизації необхідне для забезпечення можливості прийняття рішень в Індустрії 4.0 [2].

Метою атестаційної роботи є розробка моделі моніторингу SCADA-системи для сучасного зерносушильного комплексу та покращення напівавтоматичного методу управління SCADA-системами шляхом застосування агентно-орієнтованого підходу та інтелектуалізації.

# 1 КОНЦЕПЦІЯ SCADA-СИСТЕМ ТА СУЧАСНІ МЕТОДИ МОНІТОРИНГУ СИСТЕМ КЕРУВАННЯ

## 1.1 Концепція SCADA-систем

SCADA-системи призначені для здійснення моніторингу та диспетчерського контролю великого числа видалених об'єктів (від 1 до 10000, іноді на відстані в тисячі кілометрів один від одного) або одного територіально розподіленого об'єкта. До таких об'єктів належать нафтопроводи, газопроводи, водопроводи, електророзподільні підстанції, водозабори, дизель-генераторні пункти і т.д[3, 5].

Головне завдання SCADA-систем – це збір інформації про безліч видалених об'єктів, що надходить з пунктів контролю, і відображення цієї інформації в єдиному диспетчерському центрі. Також, SCADA-система повинна забезпечувати довгострокове архівування отриманих даних. Диспетчер часто має можливість не тільки пасивно спостерігати за об'єктом, але й керувати ним, реагуючи на різні ситуації.

Термін SCADA зазвичай відноситься до централізованих систем контролю і управління всією системою, або комплексами систем, здійснюваного за участю людини. Більшість керуючих впливів виконується автоматично RTU або PLC. Безпосереднє управління процесом зазвичай забезпечується RTU або PLC, а SCADA управляє режимами роботи. Цикл управління зі зворотним зв'язком проходить через RTU або PLC, в той час як SCADA система контролює повне виконання циклу.

Збір даних починається в RTU або на рівні PLC і включає свідчення вимірювального приладу. Далі дані збираються і формуються таким способом, щоб оператор диспетчерської, використовуючи НМІ міг прийняти контролюючі рішення - коригувати або перервати стандартне управління коштами RTU/PLC. Дані можуть також бути записані в архів для побудови

трендів та іншої аналітичної обробки накопичених даних.

Після того, як SCADA системи стали застосовувати в системах віддаленого доступу за допомогою мережі Інтернет, різко підвищилася вразливість SCADA до протиправних дій з боку зловмисників. Ставитися з зневагою до цієї проблеми немає можливості, оскільки це може привести до серйозних збоїв у функціонуванні різних промислових і інфраструктурних об'єктів. Що загрожує людськими життями і настільки чималим економічним збитком. В SCADA системах застосовуються такі способи для підвищення рівня безпеки їх роботи:

- здійснення розмежування рівнів доступу до системи між різними категоріями користувачів (оператор, програміст, технолог і директор повинні мати різні рівні доступу до наявної в системі інформації і до модифікації налаштувань системи);

- організація захисту інформації (шифрування даних, забезпечення максимальної надійності від вразливостей протоколів передачі інформації);

- проведення заходів щодо забезпечення підвищення безпеки оператора шляхом його віддалення від небезпечного процесу, яким він керує (дистанційне керування або remote control). Що важливо, застосування дистанційного керування є стандартною вимогою Ростехнагляду і здійснюється за допомогою провідної мережі, мережі інтернет, через радіоканал (радіо або GSM-модем) і інші види зв'язку;

- застосування спеціальних заходів і методів захисту інформації від атак зловмисників;

- використання файрволів і інших мережних захистів.

Завдання, що покладаються на SCADA системи, виходять далеко за рамки локальних об'єктів автоматизації [4, 5, 8]. Так, з їх допомогою реалізуються різноманітні розподілені структури управління підприємством, аналітична діяльність, планове виробництво, віддалений моніторинг, WEB-доступ і т.д.

Використання SCADA-технологій дозволяє не тільки збільшити

продуктивність виробництва, підвищити якість продукції, оптимізувати режими використання наявного обладнання, а й реалізувати високонадійні системи автоматизації, що відповідають найвищим вимогам безпеки.

## 1.2 Інтелектуалізація SCADA-систем

Інтелектуалізація є головним напрямком розвитку сучасних технологій, а властивість інтелектуальності притаманна всім новітнім інформаційно-керуючим системам. Ці висновки випливають з практичного досвіду роботи провідних промислових фірм і компаній, що займаються проблемами автоматизації управління у різних областях. Досвід останнього десятиліття за рішенням безлічі практичних завдань і створення сотень практично діючих систем показав, що саме інтелектуальні технології виявляються найбільш конструктивними та економічно виправданими під час розробки сучасних систем автоматизованого управління.

На даний час системи класу SCADA (Supervisory Control And Data Acquisition, диспетчерське управління і збір даних) є вельми ефективною технологією автоматизованого управління динамічними системами в багатьох галузях промисловості. Сучасні SCADA-системи мають схожі можливості і принципи функціонування, які дозволяють вирішити типові завдання, такі як: диспетчерський моніторинг і збір даних під час технологічного процесу, управління за наявністю чітких алгоритмів і повної формалізованої моделі об'єкта управління. Однак промислові та транспортні підприємства XXI-го століття є складними динамічними комплексами, і звичайного результату, який звичайні SCADA-системами надавали раніше, вже недостатньо [4-7].

Основними областями застосування SCADA-систем (за даними зарубіжних джерел), є:

- управління передачею і розподілом електроенергії;
- промислове виробництво;

- виробництво електроенергії;
- водозабір, водоочищення і водорозподілення;
- видобуток, транспортування і розподіл нафти і газу;
- управління космічними об'єктами;
- управління на транспорті (всі види транспорту: авіа, метро, залізничний, автомобільний, водний транспорт);
- телекомунікації;
- військова область.

Термін «SCADA» охоплює процеси збору інформації реального часу з віддалених об'єктів для обробки, аналізу і можливого управління цими об'єктами. Під реальним часом розуміється режим роботи автоматизованої системи обробки інформації та управління, за умови, що існують жорсткі обмеження на її тимчасові характеристики. Помилково вважати, що реальний час завжди означає «швидко». Правильніше сказати, що реальний час означає «вчасно». Іншими словами, система реального часу гарантує, що відгук відбувається за необхідний час [5-8].

Системи реального часу бувають двох типів: системи жорсткого реального часу і системи м'якого реального часу.

Системи жорсткого реального часу не допускають ніяких помилок. Іншими словами, жорсткий реальний час – це такий режим роботи системи, при якому порушення тимчасових обмежень рівнозначне відмові системи.

М'який реальний час – режим роботи системи, при якому порушення часових обмежень призводять до зниження якості роботи системи.

Практично всі сучасні SCADA-системи включають три основних структурних компонента (рисунок 1.1): RTU, MTU і CS

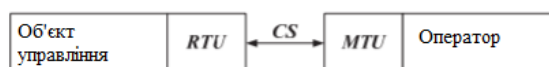


Рисунок 1.1 – Структурні компоненти SCADA-систем

Тут Remote Terminal Unit (RTU) – це віддалений термінал, що здійснює обробку завдання в режимі реального часу. Спектр втілення RTU широкий – від примітивних датчиків, які здійснюють зчитування інформації з об'єкта, до спеціалізованих багатопроцесорних відмовостійких обчислювальних комплексів, які здійснюють обробку інформації та управління в режимі жорсткого реального часу. Його реалізація визначається конкретною областю застосування [5, 8].

У свою чергу, Master Terminal Unit (MTU) є диспетчерським пунктом управління, який здійснює обробку даних і управління високого рівня, як правило, в режимі м'якого реального часу. Одна з основних функцій MTU – забезпечення інтерфейсу між людиною-оператором і системою.

Communication System (CS) є комунікаційною системою (канали зв'язку), необхідною для передачі даних з віддалених об'єктів на центральний інтерфейс оператора-диспетчера, і передача сигналів управління на RTU.

Таким чином, будь-яку автоматизовану систему управління технологічним процесом (АСУ ТП) можна представити у вигляді трирівневої системи (рисунок 1.2).

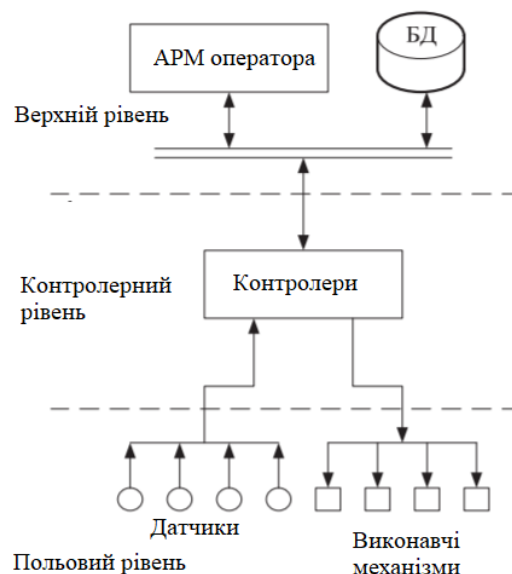


Рисунок 1.2 – Трирівнева система АСУ ТП

Перший (нижній) польовий рівень системи автоматизації включає контрольно-вимірювальні прилади та прилади автоматики, а також виконавчі пристрої управління та пульти сигналізації.

На другому (середньому) контролерному рівні для зв'язку з технологічними об'єктами управління передбачені програмовані логічні контролери (PLC). Вони забезпечують:

- збір інформації з польового обладнання, що входить до АСУ ТП;
- обробку та передачу інформації про стан об'єктів на верхній рівень системи;
- автоматичне регулювання та управління технологічним обладнанням і контроль його роботи;
- отримання інформації з верхнього рівня управління та формування керуючих впливів на електроприводи виконавчих механізмів.

Третій (верхній) рівень включає в себе:

- автоматизовані робочі місця (АРМ) операторів на базі персональних комп'ютерів зі SCADA-системою операторського управління;
- сервер баз даних (БД), якщо використовується автоматизована система оперативного або комерційного обліку.

### 1.3 Загальні функції SCADA-систем.

Диспетчер у багаторівневій АСУ ТП отримує інформацію з монітора ЕОМ або з електронної системи відображення інформації і управляє об'єктами, що знаходяться від нього на значній відстані, за допомогою телекомунікаційних систем, контролерів та інтелектуалізованих виконавчих механізмів.

Необхідною умовою ефективної реалізації диспетчерського управління, що має яскраво виражений динамічний характер, стає робота з інформацією, тобто процес збору, передачі, обробки, відображення і представлення інформації.

Таким чином, можна виділити наступні основні функції SCADA-систем:

- збір первинної інформації від пристроїв нижнього рівня;
- обробка первинної інформації;
- візуалізація інформації у вигляді графіків, гістограм і т.п. ;
- зберігання інформації з можливістю її подальшої обробки;
- управління та реєстрація сигналів про аварійні ситуації.

#### 1.4 Методи управління SCADA-системами

Існують два методи управління віддаленими об'єктами в SCADA-системах: автоматичне і напівавтоматичне (що ініціюється людиною).

##### 1.4.1 Автоматичний метод управління SCADA-системами

Велика частина наявних завдань по автоматизації управління виконується найчастіше за допомогою промислового контролера (ПЛК), але частково виконання завдань може бути покладено і на SCADA систему. Крім усього іншого, багато невеликих системи управління можуть взагалі не мати PLC, тому промисловий комп'ютер, з встановленою на нього SCADA стає єдиним засобом управління процесом [7, 8]. У сфері автоматичного управління (АСУ ТП) SCADA система, як правило, виконує наступні функції:

- ПД регулювання;
- відстеження послідовності виконання операцій в автоматизованій системі;
- автоматична перенастроювання алгоритмів роботи АСУТП до умов, що змінилися протікання керованого процесу;
- реалізація автоматичного блокування виконавчих пристроїв під час виконання раніше заданих алгоритмів.

Якщо знати передісторію об'єкта (процесу) управління, то можна значно поліпшити поведінку системи в майбутньому, проаналізувати і виявити причини виникнення ситуацій, пов'язаних з безпекою системи або появою браку продукції, визначити помилки, зроблені оператором. Щоб створити історію SCADA системою виконуються наступні операції:

- збір різних вхідних даних і твір їх обробки (цифрова фільтрація, нормалізація, інтерполяція, масштабування, стиснення та інше);
- архівування даних (дії оператора, файли конфігурації, зібрані і оброблені дані, електронні форми, звіти, події, графіки, аларми і т.д.);
- управління різними базами даних (архівні бази даних і бази даних реального часу).

#### 1.4.2 Напівавтоматичний метод управління SCADA-системами

Незважаючи на наявність безлічі функцій, які виконують SCADA системи, основною відмінністю SCADA від інших систем є наявність розробленого інтерфейсу для користувача. Якщо вилучити цей інтерфейс, то всі зазначені вище функції співпадуть з функціями, які виконують засоби програмування промислових контролерів (ПЛК), і управління стане повністю автоматизованим на противагу диспетчерського.

Іноді до комплектації SCADA системи входять засоби програмування контролерів, однак подібні рішення викликані швидше комерційним інтересом, ніж безпосередньо пов'язані з основними функціями SCADA систем [6-8].

Основною функцією SCADA системи по праву вважається створення людино-машинного інтерфейсу (HMI), тобто SCADA система виступає відразу в двох ролях – в ролі HMI і в ролі інструменту його створення. Швидкість проведених розробок в значній мірі впливає на конкурентоспроможність фірми (якої в більшості випадків є системний інтегратор), яка впроваджує системи промислової автоматизації (АСУ ТП),

саме тому швидкість розробки виступає в ролі основного показника з позиції системного інтегратора якості SCADA системи [7].

Якщо розглядати SCADA систему з точки зору диспетчерського управління, то їй належить виконання наступних завдань:

- аналіз проблемної ситуації;
- ідентифікація виниклого відхилення від нормального (штатного) режиму функціонування об'єкта;
- пошук можливих коригувальних рішень по впливу на об'єкт;
- прогнозування ситуацій;
- оцінка наслідків прийнятих рішень;
- видача команд на відпрацювання необхідних управляючих впливів;

Особливості методу напівавтоматичного управління в SCADA-системах:

- будь-яка неправильна дія може призвести до відмови об'єкта управління або навіть катастрофічних наслідків;
- диспетчер несе, як правило, спільну відповідальність за управління системою, яка, при нормальних умовах, тільки зрідка вимагає підстроювання параметрів для досягнення оптимального функціонування;
- велику частину часу диспетчер пасивно спостерігає за відображається інформацією. активну участь диспетчера в процесі управління відбувається нечасто, зазвичай в разі настання критичних подій - відмов, аварійних і позаштатних ситуацій тощо;
- дії оператора в критичних ситуаціях можуть бути жорстко обмежені за часом (декількома хвилинами або навіть секундами).

## 2 АГЕНТНО-ОРІЄНТОВАНЕ МОДЕЛЮВАННЯ ДЛЯ ВИКОРИСТАННЯ У МЕТОДАХ УПРАВЛІННЯ

### 2.1 Методологія інтелектуальних агентів

#### 2.1.1 Визначення поняття «інтелектуальний агент»

Термін інтелектуальний агент має два значення, унаслідок чого виникає плутанина. До теперішнього часу сформувалося декілька різних трактувань.

Від Дж. Холланда пішло уявлення про агента, як про штучний організм, що розвивається у популяції собі подібних, прагне навчатися та адаптуватися до зовнішнього середовища, щоб вижити у ньому та перемогти конкурентів. Ця інтерпретація агента спирається на теоретичні підходи і моделі штучної еволюції – мутації та модифікації агентів, їхня боротьба за існування, відбір найсильніших або відбраковування найслабших, принципи штучного життя (самовідтворення, самозбереження, самовизначення та саморегуляція агентів). Вона тісно зв'язана із робототехнікою (питаннями побудови інтегральних роботів і функціонування групи роботів), проблемами співвідношення «броні» та «снаряду» у комп'ютерних мережах, питаннями інформаційної безпеки та інформаційних атак, комп'ютерної вірусології та створення засобів Liveware – програмне забезпечення, що еволюціонує та розробляється з урахуванням принципів і механізмів поведінки живих організмів [9].

У комп'ютерних науках термін «інтелектуальний агент» означає деяку програму, що самостійно виконує завдання вказане користувачем комп'ютера, у доволі тривалі проміжки часу. Інтелектуальні агенти використовуються для підтримки оператора чи для збору інформації. Одним з прикладів завдань, що виконують агенти, може бути завдання постійного

пошуку та збору необхідної інформації в Інтернеті. Комп'ютерні віруси, боти, пошукові роботи – це все також можна віднести до «інтелектуальних» агентів [9-11]. Такі агенти, як і будь-які інші, мають складний, найчастіше реалізований нейромережами алгоритм, як, наприклад, у пошуковій системі Google (експериментальний пошук по відео). «Інтелектуальність» у цьому контексті розуміється як можливість зворотного зв'язку в співвідношенні, наприклад, із результатами аналізу пошукових запитів та їхньою видачею.

Виникла метафора агенту як персонального помічника користувача, чи пізніше, як інтелектуального посередника між користувачем та середою, де він працює. Зокрема, стратегія розробки штучних агентів, що представлена в IBM White Paper, виходить з цієї ідеї «персонального помічника», причому агентом вважається будь-яка програмна чи апаратна система, здатна діяти в інтересах досягнення цілей, що поставлені користувачем.

Ідея персональних помічників під час спілкування користувачів з ЕОМ у своєму найпростішому варіанті була втілена у ряді популярних програмних продуктів. Наприклад, фірма Microsoft вбудувала Wizards та System Agent у Windows 95, а у Microsoft Office з'явилась скріпка-помічник.

Агенти можуть розглядатися як активні об'єкти або метаоб'єкти, наділені деякою часткою суб'єктності, тобто здатні маніпулювати іншими об'єктами, створювати та винищувати їх, а також взаємодіяти із середовищем та іншими агентами. У цьому контексті вони можуть створюватися та основі програмування в обмеженнях із використанням технології активних об'єктів. Таким чином, програмна технологія агентів та агентно-орієнтований підхід у цілому розуміються як природній розвиток ідей об'єктно-орієнтованого програмування (ООП). При цьому агент являє собою самодостатній програмний процес, що включає деякий стан та має можливість взаємодіяти з іншими агентами через обмін повідомленнями. Відповідно, агентно-орієнтоване програмування (АОП) – це нова парадигма програмування, заснована на «соціальному погляді» на обчислення.

На рисунку 2.1 наведений приклад роботи інтелектуального агента та його взаємодія з навколишнім середовищем.

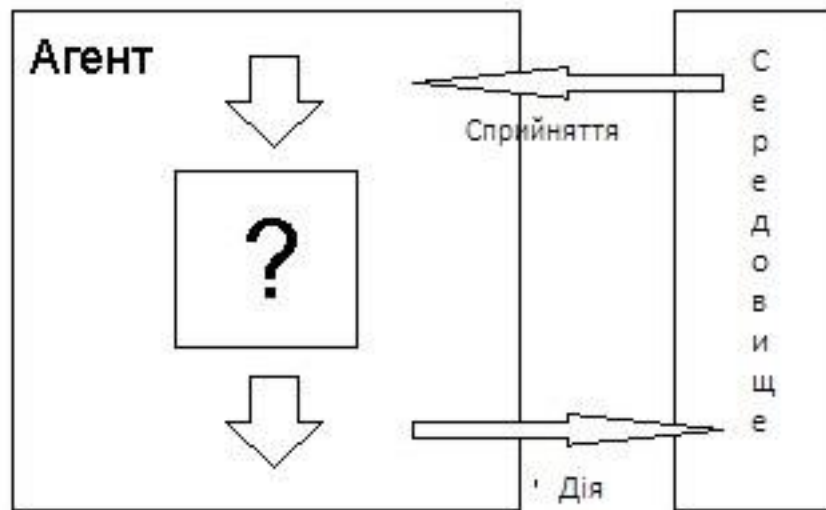


Рисунок 2.1 – Приклад роботи інтелектуального агента

З одного боку, введення неоднорідних об'єктів, зокрема, побудова ієрархії програмних об'єктів за ступінню їхньої активності чи автономії, являє собою розширення традиційної теорії програмних систем та концепції ООП [10]. З іншого боку, наявність у агентів специфічних протоколів комунікації, які зазвичай будуються на основі теорії мовних актів шляхом моделювання діалогу між агентом-замовником та агентом-виконавцем може розглядатися як деталізація та конкретизація ідей ООП.

У штучному інтелекті під терміном інтелектуальний агент розуміються явища, що отримують інформацію через систему сенсорів про стан процесів, що керуються ним та здійснюють вплив на них через систему актуаторів, при цьому їхня реакція раціональна у тому значенні, що процеси, які виконуються ними, сприяють досягненню певних параметрів. Найбільш близьким аналогом у живій природі є примітивна поведінка комах.

Термін «інтелектуальний» не означає наявності будь-якого інтелекту, але підкреслює більш високий рівень технології керування у порівнянні з

примітивними тригерними системами автоматичного керування. Такий агент може бути як програмною системою, так і складною автоматизованою системою, наприклад, станком з ЧПУ чи комплексом керування технологічними, логістичними, фінансовими чи будь-якими іншими процесами. Про «інтелектуальність» агента можна говорити, якщо його взаємодія із зовнішнім середовищем є адекватною тій чи іншій системі вимог. Жодного відношення навіть до інтелекту найвищих тварин і вже тим більше людини, подібна функціональність не має.

Концепція розвитку інтелектуального агента, як штучного діяча заснована на специфікаціях Міжнародної федерації з розробки інтелектуальних фізичних агентів, де під поняттям агент розуміється актуатор у проблемній області, що містить одну чи декілька службових функцій, поєднує їх в інтегровану модель виконання дій, а також має доступ до зовнішніх програм, користувачам та засобам комунікації. Очевидно, що розвиток даного підходу в теорії агентів повинен спиратися на результати, отримані у психології діяльності та інженерній психології.

### 2.1.2 Класифікація агентів

Будь-який агент являє собою відкриту систему, що розміщена у деякому середовищі, причому ця система володіє власною поведінкою, яка задовольняє деякі екстремальні принципи. Таким чином, агент вважається здатним сприймати інформацію із зовнішнього середовища з обмеженим дозволом, оброблювати її на підставі власних ресурсів, взаємодіяти з іншими агентами та діяти на середовище у проміжок деякого часу, переслідуючи свої власні цілі. Це означає, що під час побудови штучного агента мінімальний набір базових характеристик включає такі властивості, як:

- адаптивність – здатність навчатися;
- автономність – агент працює як самостійна програма, що ставить собі цілі і вживає заходів для їх досягнення;

- інтелектуальність – агенти можуть володіти знаннями або механізмами виведення на основі узагальнення, наприклад, мати вбудовану штучну нейронну мережу;
- колаборативність – взаємодія з іншими агентами, причому агент може грати різні ролі під час взаємодії з одним і тим же агентом;
- комунікативність – здатність вирішувати свої завдання сумісно із іншими агентами;
- мобільність – зданість до просторового переміщення агента або міграції агента з однієї агентської групи до іншої;
- реактивність – здатність до сприйняття зовнішнього середовища;
- цілеспрямованість – передбачає наявність власних джерел мотивації, а у більш ширшому плані, особливих інтеціональних характеристик.

Проста агентна програма може бути математично описана як агентська функція, яка проектує будь-який відповідний результат сприйняття на дію, яку агент може виконати, чи у коефіцієнт, елемент зворотного зв'язку, функцію або константу, які можуть вплинути на подальші дії:

$$f: P^* \rightarrow A \quad (2.1)$$

По одній з існуючих класифікацій всіх агентів можна розділити на п'ять груп по типу обробки сприйнятої інформації.

1. Агенти із простою поведінкою – агенти із простою поведінкою виконують дії тільки на підставі поточних знань. Їхня агентська функція заснована на схемі умова-дія. Така функція може бути успішною, тільки якщо зовнішнє середовище повністю піддається спостереженню. Деякі агенти також можуть мати інформацію щодо поточного стану, що дозволяє їм не звертати уваги на умови, передумови яких вже виконані.

2. Агенти із поведінкою, заснованою на моделі – можуть оперувати із середовищем, що лише частково піддається спостереженню. Всередині агента зберігається уявлення про ту частину, що знаходиться поза межами

огляду. Для того, щоб мати таке уявлення, агенту необхідно знати, як виглядає зовнішнє середовище, як воно побудовано. Ця додаткова інформація доповнює «картину світу».

3. Цілеспрямовані агенти – схожі з попереднім типом, але вони, окрім іншого, зберігають інформацію про ті ситуації, які для них є бажаними. Це надає агенту можливість обрати серед багатьох шляхів той, який призведе до потрібної мети.

4. Практичні агенти – цілеспрямовані агенти розрізняють тільки стани, коли мета є досягнутою, і коли вона не є досягнутою. Практичні агенти, окрім цього здатні, наскільки бажаним для них є поточний стан. Така оцінка може бути отримана за допомогою «функції корисності», яка проектує множину станів на множину мір корисності станів.

5. Агенти, що навчаються – у деякій літературі агенти, що навчаються також називають автономними інтелектуальними агентами, що означає їхню незалежність та здатність до навчання та пристосуванню до обставин, що змінюються. Система агентів, що навчаються повинна проявляти наступні здібності:

- навчатися та розвиватися у процесі взаємодії із зовнішнім середовищем;
- пристосовуватися у режимі реального часу;
- швидко навчатися на основі великого об'єму даних;
- покроково пристосовувати нові способи рішення проблем;
- володіти базою прикладів з можливістю її поповнення;
- мати параметри для моделювання швидкої та довгої пам'яті, віку та ін..;
- аналізувати себе у термінах поведінки, помилки та успіху.

Інша класифікація агентів передбачає поділ за критеріями, що пов'язані з полярними шкалами «природне – штучне» та «матеріальне – віртуальне». За першим критерієм виділяються:

- натуральні агенти – тварини, людини;

- штучні агенти – роботи, колективи автоматів, складні комп'ютерні програми.

Згідно другому критерію всі штучні агенти поділяються на:

- матеріальні, фізично існуючі та працюючі у реальному просторі, наприклад, інтегральні роботи, наділені різноманітними засобами «одухотворення», маніпуляторами чи педіпуляторами.

- віртуальні, існуючі лише у деякій програмній середі (віртуальному просторі), які нерідко можна представити у якості роботів, що зайняті не фізичною, а інформаційною роботою, такі «програмні роботи» (software robots) називають скорочено софтботами (softbots).

Ще одна пара взаємопов'язаних критеріїв класифікації спирається на дихотомії «зосереджене – розподілене» та «нерухоме – рухоме». Прикладом нерухомого агента є промисловий маніпуляційний робот, а прикладом рухомого – програмний пошуковий агент, що мігрує по комп'ютерній мережі у цілях відшукування потрібної інформації [9-11].

Важливою засадою для класифікації слугує наявність (відсутність) у агентів характеристик навченості або адаптивності. У агентів, що навчаються, поведінка заснована на попередньому досвіді.

Ще одною важливою засадою для класифікації штучних агентів є прийняття або психологічної, або біологічної метафори під час розглядання природи їхніх дій (дихотомія «психологічне – біологічне»). В одному випадку йдеться про трактування агентів, як квазісуб'єктів, що самостійно розв'язують поставлені перед ними завдання, а в іншому вони уподібнюються найпростішим організмам, що безпосередньо реагують на зміни середовища в інтересах виживання та адаптації.

В цілому дана типологія агентів тісно пов'язана із класичною проблемою взаємодії «суб'єкт – об'єкт». Рівень суб'єктності агента безпосередньо залежить від того, чи наділений він символічними уявленнями, які потрібні для організації міркувань, чи у протилежність цьому він працює тільки на рівні образів (субсимвольному), що пов'язані із сенсомоторною

регуляцією. Тоді класифікацію агентів можна побудувати за наступною схемою, наведеною на рисунку 2.2.



Рисунок 2.2 – Класифікація агентів

За першою ознакою виділяються інтелектуальні (когнітивні, міркуючи, комунікативні та ресурсні) та реактивні агенти.

Інтелектуальні агенти мають добре розвинену символічну модель навколишнього середовища, яка постійно поповнюється та досягається завдяки наявності у них бази знань, механізмів рішення та аналізу дій. Близький термін «розмірковуючий» зарезервований для позначення агента, який на основі символічної моделі зовнішнього середовища здатний проводити власні міркування, наприклад, використовуючи метод порівняння за зразком, та на їх основі приймати власні рішення або виконувати дії, що змінюють середу.

Невелике розходження між цими типами інтелектуальних агентів пов'язані із розстановкою акцентів на тих чи інших інтелектуальних функціях: або на отримання знань про середовище, або на міркування про можливі дії. У комунікативних агентів внутрішня модель світу перетворюється головним чином у модель спілкування, що складається з моделей учасників, процесу та бажаного результату спілкування. Нарешті, база знань ресурсного агента містить в основному знання про структуру та

стан ресурсів, що визначають різні форми поведінки.

У повноцінного інтелектуального агента обов'язково повинні бути присутніми як мінімум чотири перелічених функції: когнітивна, розмірковуюча (у більш загальному контексті, регулятивна), комунікативна та ресурсна.

У той же час реактивні агенти не мають ні скільки-небудь розвиненого уявлення зовнішнього середовища, ні механізму багатокрокових міркувань, ні достатньої кількості власних ресурсів. Звідси впливає ще одна істотна відмінність між інтелектуальними та реактивними агентами, що пов'язано з можливостями прогнозування змін зовнішнього середовища і, як наслідок, свого майбутнього.

Реактивні агенти володіють дуже обмеженим діапазоном передбачення. Вони практично не здатні планувати свої дії, оскільки реактивність у чистому вигляді означає таку структуру зворотного зв'язку, який не містить механізмів прогнозу. Тоді як інтелектуальні агенти завдяки багатим внутрішнім уявленням зовнішнього середовища та можливостям міркувань можуть запам'ятовувати та аналізувати різні ситуації, передбачати можливі реакції на власні дії, робити з цього висновки, які є корисними для подальших дій та у результаті планувати свою поведінку. Саме розвинені когнітивні здатності дозволяє таким агентам будувати віртуальні світи, працюючи у яких вони формують плани дій.



Рисунок 2.3 – Властивості інтелектуального агента

Таблиця 2.1 – Порівняльний аналіз реактивних та когнітивних агентів

Характеристики	Когнітивні агенти	Реактивні агенти
Внутрішня модель зовнішнього світу	Розвинута	Примітивна
Міркування	Складні та рефлексивні міркування	Прості однокрокові міркування
Мотивація	Розвинута система мотивації, що включає до себе переконання, бажання та наміри	Найпростіші переконання, що пов'язані із виживанням
Пам'ять	Присутня	Відсутня
Реакція	Повільна	Швидка
Адаптивність	Маленька	Велика
Модульна архітектура	Є	Немає
Склад багатоагентної системи	Невелика кількість автономних агентів	Велика кількість залежних один від одного агентів

Інтелектуальні агенти, будучи значно автономніше реактивних, мають куди яскравіше виражену індивідуальність і характеризуються доцільним поведінкою в співтоваристві агентів, а також прагненням використовувати ресурси інших агентів для досягнення власних цілей. У той же час, реактивні агенти, як це видно з самого їх назви, працюють в основному на рівні стимульно-реактивних зв'язків, володіючи дуже бідної індивідуальністю і сильною залежністю від зовнішнього середовища (спільноти агентів). Результати порівняльного аналізу реактивних і когнітивних агентів представлені в таблиці 2.1.

Далі, за типом поведінки інтелектуальні агенти діляться на інтенціональних і рефлексорних, а реактивні – на спонукаючих (імпульсивних) і трофічних. Більшість інтелектуальних (когнітивних) агентів

можна віднести до числа інтенціональних. Подібні агенти наділені власними механізмами мотивації. Це означає, що в них так чи інакше моделюються внутрішні переконання, бажання, наміри і мотиви, які породжують цілі, які і визначають їх дії.

У свою чергу, модульні чи рефлекторні агенти не мають внутрішніх джерел мотивації та власних цілей, а їхня поведінка характеризується найпростішими (однокроковими) висновками чи автоматизмами.

Таким чином, вони являють собою граничний випадок поняття когнітивного агента та можуть використовуватися як допоміжні агенти. Такі агенти здатні відповідати на питання та виконувати завдання, які ставлять перед ними інші агенти, але розв'язання цих задач не призведе до появи у них власних цілей. Типовими прикладами таких вироджених агентів є системи пошуку в базах даних та найпростіші логічні регулятори.

У свою чергу реактивні агенти містять скопійовані знання щодо потрібних дій: їм не потрібно будувати детальне внутрішнє уявлення зовнішнього середовища, оскільки цілком достатніми є реакції на набір представлених ситуацій, тобто характер їхньої реакції визначається тільки поточною інформацією.

За складністю цих реакцій та походженню джерел мотивації реактивні агенти діляться на імпульсивних та трофічних агентів. У випадку трофічних агентів поведінка визначається найпростішими трофічними зв'язками. Фактично вона зводиться до відповіді на стимули, що поступають із зовнішнього середовища (власних мотивів та цілей немає), тобто повністю визначається її локальним становищем. Типовою моделлю подібних агентів є клітинні автомати, де основними параметрами виступають: радіус сприйняття агента, кількість умовних одиниць живлення у зовнішньому середовищі та енергетична цінність одиниці. Тут кожний трофічний агент володіє невеликим набором ситуаційних правил, що задають його реакції на сигнали із середовища. Прикладами подібних правил є вираження типу «якщо у радіусі сприйняття не виявлена одиниця живлення, то випадковим

чином обрати один з вільних сусідніх квадратів та пересунути на цей квадрат».

Між тим, реактивні агенти, що називаються імпульсивними, також можуть мати примітивний механізм мотивації, який штовхає їх на виконання задачі, наприклад, задоволення набору життєвих потреб. Йдеться про підтримку енергетичного балансу, чи у більш ширшому значенні, про умови виживання агента. Дія механізму самозбереження в агента характеризується здібностями визначення та збільшення відстані меж свого існування. Збудження до якої-небудь дії імпульсивного агента засновується на таких факторах, як відхилення деякої життєво важливої змінної (потреби) від норми та ваги (суб'єктивна важливість) цієї потреби [11].

Отже, когнітивні агенти, завдяки їхній складності, наявності знань та здатностей до міркувань про свою поведінку та зовнішнє середовище, можуть бути більш автономними, ніж реактивні. Вони працюють відносно незалежно, демонструючи доволі гнучку поведінку. Але все та ж складність автономних агентів, що виливається в здатність опиратися зовнішнім впливам, викликає певні труднощі під час організації їх ефективної взаємодії. Тому у складі багатоагентної системи, що включають тільки інтелектуальних агентів, як правило, присутнє не більш  $7 + 2$  автономних одиниць.

Напроти, доволі проста структура реактивних агентів, зумовлює їхню жорстку залежність від середовища. Тобто, їх можливості порівняно невеликі, коли вони функціонують поодиноці та обмежені своїми власними ресурсами. Однак їм легше створити групу чи організацію, здатну гнучко адаптуватися до змін середовища під дією механізму природного відбору. Тому реактивні агенти представляють інтерес не на індивідуальному рівні, а не колективному рівні, причому їхні здібності до адаптації та розвитку виникають спонтанно у результаті локальних взаємодій. Таким чином, реактивні агенти, які практично не мають індивідуальності, розчиняючись у загальній масі за рахунок свого великого числа та надмірності можуть розв'язувати складні задачі. У межі, відповідні багатоагентні системи можуть

формуватися у результаті спонтанних взаємодій без точної специфікації окремих агентів. Подібні «хмари» або «рої» (swarms), що складаються із значного числа рухомих, реактивних агентів, можна порівняти з деяким зверх організмом. Взаємна адаптація та кооперація клітин у такому організмі дозволяє створити загальний ланцюг зворотного зв'язку, що забезпечує гомеостазис усій системи.

Інтелектуальні агенти за характером інформаційної взаємодії між собою можуть поділятися на правдивих (що прямують до передачі істинної інформації) та недостовірних (що орієнтовані на дезінформацію інших агентів).

За рівнем «свободи волі», характером намірів та відношенням до партнерів пов'язані уявлення щодо благонамірених (benevolent) та зленамірених, егоїстичних (self-interested) та альтруїстичних агентів. Зокрема, вважається, що благонамірени агенти завжди намагаються робити тільки те, що від них вимагається, і уникають конфліктних ситуацій.

Нарешті, ще один варіант класифікації, де додатково до біологічного та психологічного рівнів агентстворення вводиться соціальний та використовуються аналогії з тріадою «рослина – тварина – людина». Реактивних, інтеціональних та соціальних агентів можна уподібнити компонентам цієї тріади. Агенти, що подібні рослинам, характеризуються реактивністю, виконанням стереотипних програм та посилкою повідомлень іншим агентам та в середовище. Агенти, що подібні тваринам, здатні обирати цілі, будувати плани дій та забезпечувати їхнє виконання. Вони координують свої дії, обмінюючись інформацією щодо індивідуальних перевагах або задачах. Агенти, що подібні людині, володіють внутрішніми моделями інших агентів (та здатністю до рефлексії), характеризуються соціальною (рольовою) поведінкою. Складність внутрішніх моделей залежить від рівня знань та досвіду такого агента.

## 2.2 Особливості агентного моделювання як методу імітаційного моделювання

Агенти можуть мати штучний інтелект, зокрема, реалізовуватися за допомогою штучних нейронних мереж (ШНМ), карт станів або вбудованих системно-динамічних оптимізаційних моделей. Також можлива комбінація різних методів і моделей в рамках одного агента.

Крім зовнішнього управління, що реалізується в організаційних структурах, як правило, корпоративним центром, можливо і внутрішнє управління, що реалізується самим агентом при виборі певної стратегії поведінки [10].

На рисунку 2.4 показаний двоспрямований зв'язок між агентами і зовнішнім середовищем, де вони функціонують, оскільки у реальних системах сукупна діяльність агентів істотно впливає на загальну середу. Типовим прикладом є поведінка окремих інтелектуальних агентів, що звертають увагу на вибір інших агентів в системі під час виконання конкретних завдань (ефект більшості).

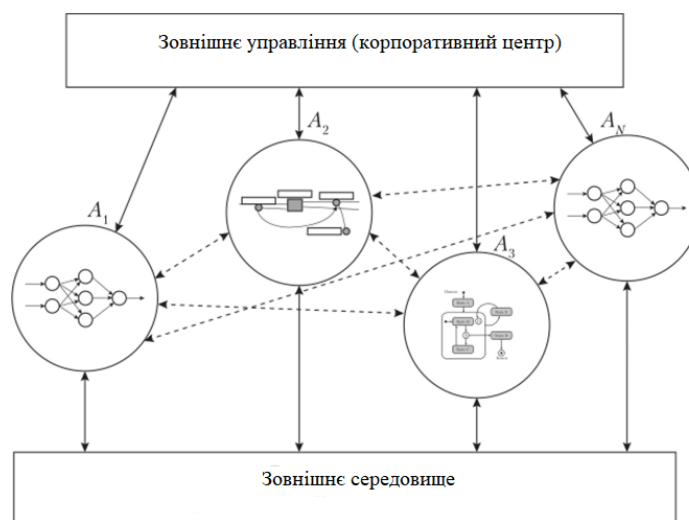


Рисунок 2.4 – Приклад моделі управління інтелектуальними агентами

Під час проектування мультиагентних систем важливо врахувати взаємовплив агентів різної природи з урахуванням їх підпорядкованості та в залежності від центрального завдання. Зокрема, для систем оперативного управління виробництвом першорядне значення має взаємодія технічних агентів (наприклад, обладнання в цеху), а роль соціальних агентів враховується при моделюванні споживчого попиту.

Такий поділ агентів вельми умовний і залежить від завдання. Разом з тим вибір ефективного методу моделювання поведінки агентів залежить від їх природи. Очевидно, найскладнішими є соціальні агенти, поведінка яких носить далеко не завжди раціональний характер і багато в чому залежить від групових (кластерних) особливостей.

У реальних системах зв'язки між агентами, як правило, обмежені. Цей фактор має суттєвий вплив на стан системи в цілому і призводить до появи стійких структур – об'єднань агентів в групи із загальними ознаками або цілями. Прикладами таких структур є штучні суспільства – комп'ютерні моделі, що описують групову поведінку агентів - членів штучних товариств.

У складних організаційних системах агенти можуть мати вкладену структуру (рисунок 2.5).

Між елементами вкладеної структури (рисунок 2.5) також можуть бути встановлені певні зв'язки. При цьому ступінь вкладеності практично необмежений (наприклад, починаючи від окремого інтелектуального агента, закінчуючи «родиною» агентів у цілому).

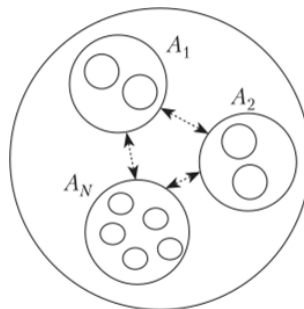


Рисунок 2.5 – Вкладена структура мультиагентной системи

Поведінка агента визначається набором правил. У результаті моделювання можна оцінити кінцеві результати діяльності всієї соціально-економічної системи.

Це можна назвати підходом «знизу-вгору»: можна зробити висновки про роботу системи в цілому, ґрунтуючись на поведінці її окремих елементів – індивідуальних агентів. З практичної точки зору Агентне моделювання можна визначити як метод імітаційного моделювання, який досліджує поведінку децентралізованих агентів і то, як це поведінка визначає поведінку всієї системи в цілому.

Іншою причиною високого інтересу є широта можливостей застосування моделей, а також їх гнучкість.

Незалежні агенти мають заздалегідь визначеними характеристиками. На основі заздалегідь прописаного набору правил, кожен агент оцінює поточну ситуацію і приймає рішення, а це впливає на систему, яка відповідно змінюється.

Агенти представляють собою реальні компоненти системи (наприклад, людей, групи осіб, організації, підприємства, навіть міста або комп'ютерні програми і т.д.), які діють і реагують в контексті певних умов.

На основі первинних або постійно змінюються параметрів в процесі моделювання оцінюється поведінка агентів протягом певного часу з метою прогнозування можливого розвитку системи в цілому. Єдиної думки з приводу визначення поняття «агент» в науковій літературі не існує, в цілому можна визначити агента як активний елемент системи, створений людиною для певної мети.

Кожен агент повинен мати здатність вирішувати аналізовані проблеми в певному середовищі. Вибір здібностей залежить від складності та типу розв'язуваної проблеми. До найбільш важливим здібностям відносяться:

- здатність обробляти інформацію, що надходить від інших агентів, і передавати її;
- здатність приймати рішення на основі даної інформації;

- здатність ставити власні цілі (в кінцевому підсумку цілі для інших агентів) і здійснювати дії для їх виконання;
- вміння співпрацювати з іншими агентами і обмірковувати їх здатності;
- здатність до зміни моделей власної поведінки.

### 2.3 Процедура формування агентної моделі

На відміну від системної динаміки поведінку агентів визначається на індивідуальному рівні, а глобальне поведінка виникає як результат діяльності багатьох агентів (моделювання «знизу вгору»).

Мультиагентні системи являють собою сукупність інтелектуальних агентів. Агентне моделювання включає в себе елементи теорії ігор, складних систем, мультиагентних систем і еволюційного програмування, методи Монте-Карло, використовує випадкові числа та ін.

Поведінка агента задається так званої картою станів (statechart), програмованої на індивідуальному рівні. Агентно-орієнтований підхід дозволяє врахувати індивідуальні особливості активних елементів програмно-апаратного комплексу.

Карта станів (рисунок 2.6) – це фактично кінцевий автомат з декількома зручними доповненнями, що ввійшли в стандартний UML (the unified modeling language уніфікована мова моделювання), та є запропонованими Девидом Харелом. Карти станів дозволяють графічно визначити можливі стани агента, переходи між ними, події, що викликають ці переходи, тимчасові затримки та дії, що здійснюються агентом протягом свого життєвого циклу. Такі конструкції, як вкладені стани, дозволяють задавати режими функціонування агента.

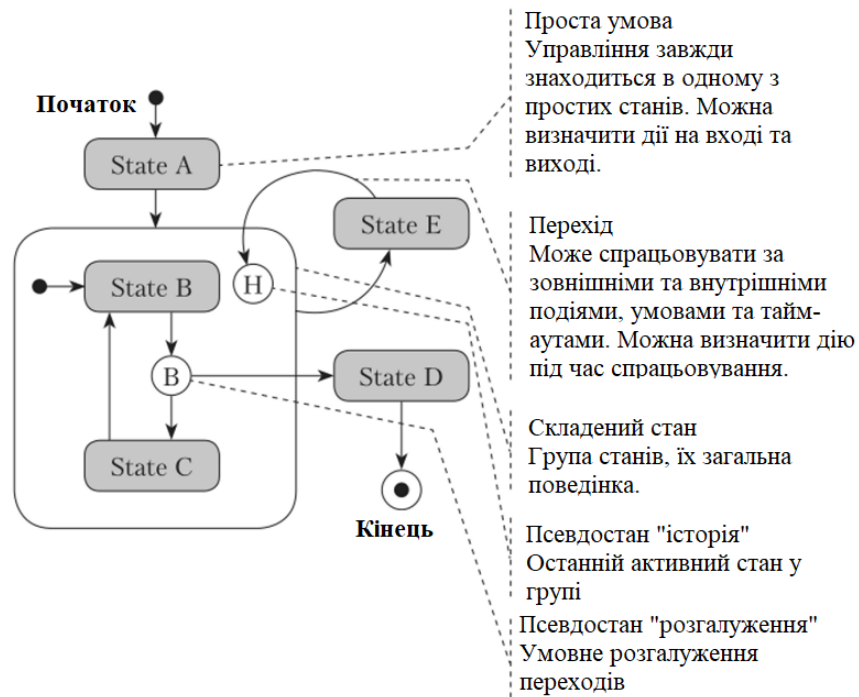


Рисунок 2.6 – Карта станів інтелектуального агента

Агент може мати кілька паралельно активних і взаємодіючих карт станів, кожна з яких відповідає за будь-який аспект його діяльності.

Формальної універсальної процедури формування агентної моделі не існує. Кожного разу треба створювати індивідуально виходячи зі специфічних умов кожної середовища. Незважаючи на це, можна виділити наступні загальні основні етапи процедури агентного моделювання: формулювання проблеми (завдання):

- вибір середовища реалізації;
- визначення агентів;
- характеристика агентів (створення атрибутів і правил поведінки);
- реалізація моделі;
- експериментів з моделлю (пошук стабільності);
- перевірка моделі;
- аналіз результатів.

Більшість агентних моделей складається з наступних компонентів:

- окремі агенти, визначені відповідно до певними ознаками,

- правила прийняття рішень і правила для навчання агентів (їх адаптація),
- правила для взаємодії між агентами,
- зовнішнє середовище,
- правила для реалізації в рамках програмного забезпечення для моделювання.

Агентні моделі створюються за допомогою спеціальних програмних засобів. Використовувані для цього в більшості випадків мови програмування – Java або C ++.

Є один принциповий недолік цих методів – вимога уміння програмувати на відповідних мовах програмування. На щастя, є деякі інтегровані комп'ютерні середовища - наприклад, Netlogo або AnyLogic, які роблять визначення графічного інтерфейсу і контроль елементів управління простіше для користувачів. Моделі зазвичай використовують такі мови комунікації як ACL, KQML або KIF.

## 2.4 Модель акторів

Модель акторів (Actor Model) – математична модель паралельних обчислень, що будується навколо поняття «актора», що вважається універсальним примітивом паралельного виконання. Актор в даній моделі взаємодіє шляхом передачі повідомлень з іншими акторами, у відповідь на що отримуються повідомлення може приймати локальні рішення, створювати нові актори, посилати свої повідомлення, встановлювати, як слід реагувати на повідомлення.

За аналогією з філософією об'єктно-орієнтованого програмування, де кожен примітив розглядається як об'єкт, модель акторів виділяє в якості універсальної сутності поняття «актора». Актор є обчислювальною сутністю, яка у відповідь на отримане повідомлення може одночасно:

- відправити кінцеве число повідомлень іншим акторам;

- створити кінцеве число нових акторів;
- обрати поведінку, яка буде використовуватися під час обробки наступного отриманого повідомлення.

Не передбачається існування певної послідовності вищеописаних дій і всі вони можуть виконуватися паралельно.

Відділення відправника від посланих повідомлень стало фундаментальним досягненням моделі акторів: тим самим забезпечується асинхронна зв'язок і управління структурами у вигляді форми передачі повідомлень.

Одержувачі повідомлень ідентифікуються за адресою, який іноді називають «поштовою адресою». Таким чином, актор може взаємодіяти тільки з тими акторами, адреси яких він має, може витягти адреси з отриманих повідомлень або знати їх заздалегідь, якщо актор створений їм самим.

Модель характеризується внутрішньо властивим паралелізмом обчислень всередині одного актора і між акторами, динамічним створенням акторів, включенням адрес акторів в повідомлення, а також взаємодією тільки через прямий асинхронний обмін повідомленнями без будь-яких обмежень на порядок прибуття повідомлень.

Модель акторів може використовуватися в якості основи для моделювання, розуміння і аргументації по широкому спектру паралельних систем, наприклад:

- електронна пошта (e-mail) може бути змодельована як система акторів: клієнти моделюються як актори, а адреси електронної пошти – як адреси акторів;
- веб-сервіси із кінцевими точками SOAP можуть бути змодельовані як адреси акторів;
- об'єкти з семафора (наприклад, в Java і C#) можуть бути змодельовані як паралельно-послідовний перетворювач, за умови, що їх реалізація така, що повідомлення можуть приходити постійно (можливо,

вони зберігаються у внутрішній черги);

- мова програмування SmallTalk побудований виключно на взаємодії об'єктів за допомогою повідомлень один одному;

- нотація тестування і управління тестами (як TTCN-2, так і TTCN-3) досить близько відповідає моделі акторів.

Серед ранніх мов програмування з підтримкою акторів – Act 1, 2 і 3, Acttal, Ani, Cantor, Rosette.

Пізніші мови, орієнтовані на модель акторів: Actor-Based Concurrent Language (ABCL), ActorScript, AmbientTalk, Axum. Серед мов програмування загального призначення, де використовується поняття актора – E, Elixir, Erlang, Io, SALSA, Scala.

Розроблено бібліотеки і табличні структури з акторами для забезпечення актороподібного стилю програмування на мовах, які не мають вбудованих акторів.

## 2.5 Методи агентного моделювання

### 2.5.1 Метод «теорії ігор»

Типова агентна модель може використовувати елементи теоретико-ігрового підходу. Теорія ігор – математичний метод вивчення оптимальних стратегій в іграх. Під грою розуміється процес, в якому беруть участь дві і більше сторін, що ведуть боротьбу за реалізацію своїх цілей. Кожна зі сторін має свою мету та використовує певну стратегію, що може вести до виграшу або програшу – залежно від поведінки інших гравців [11, 12]. Теорія ігор допомагає вибрати найкращі стратегії з урахуванням уявлень про інших учасників, їх ресурсах і можливих вчинках. Можна стверджувати, що в мультиагентних системах поведінку агента, зокрема вибір оптимальної стратегії, залежить від стратегії інших агентів. Так, наприклад, володіючи штучним інтелектом або перебуваючи під впливом зовнішнього управління,

агент може домовлятися з іншими агентами, вступаючи з ними в коаліційні відносини, щоб координувати свої дії. У той же час між певними агентами можлива конкуренція, навіть якщо вони є учасниками однієї організаційної структури. Типовим прикладом такої ситуації є конкуренція дочірніх класів однієї програмно-апаратної системи за перерозподіл обов'язків між собою.

Під час управління активною системою важливо розуміти, які агенти будуть діяти коаліційно, а які – конкурентно. Можливо також зміна стратегії агентів у процесі взаємодії, наприклад, перехід від конкурентної стратегії до коаліційної і навпаки.

Агенти функціонують і взаємодіють один з одним в певному зовнішньому середовищі. Модель такого середовища може включати до себе різні характеристики, наприклад, різний набір параметрів, єдині центри управління, зовнішні системні чинники і т.д.

### 2.5.2 Метод еволюційного програмування

Еволюційні обчислення – це досить широкий і розпливчастий «зонтичний» термін, який об'єднує в собі безліч різних, хоча і схожих технік. Якщо конкретніше, еволюційне програмування – це повторюваний алгоритм, який буде постійним, але його параметри залишаться відкритими для оптимізації.

Еволюційне програмування полягає в простій імітації механізму природного відбору (рисунок 2.7):

- Initialisation (ініціалізація) – еволюція повинна стартувати з початкового рішення. Вибір хорошої стартової точки важливий, тому що він може привести до абсолютно різних результатів;

- Duplication (дублікація) – створюється безліч копій поточного рішення;

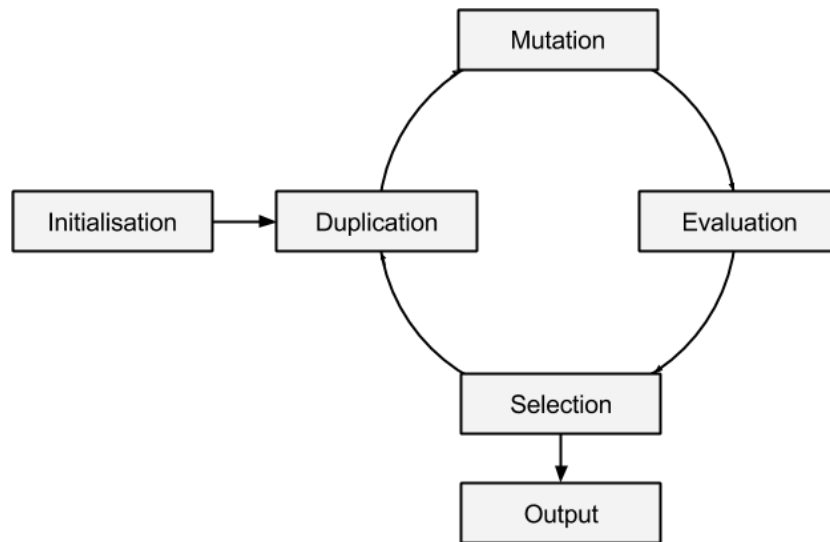


Рисунок 2.7 – Імітація механізму природного відбору

- Mutation (мутація) – кожна копія випадковим чином мутує. Величина мутації критична, тому що вона управляє швидкістю виконання еволюційного процесу;

- Evaluation (оцінка) – змінюється оцінка гена, що залежить від показників, продемонстрованих згенерованою істотою. У багатьох випадках для неї потрібний етап інтенсивної симуляції;

- Selection (відбір) – після оцінки істот кращим з них дозволяється реплікація, що дозволяє стати основою наступного покоління;

- Output (висновок) – еволюція (ітеративний процес). На будь-якому етапі її можна зупинити, щоб отримати поліпшену (або ту ж саму) версію попереднього покоління.

Мета еволюції – максимізація пристосованості генома, кращої відповідності певному середовищі. Незважаючи на глибокий зв'язок з біологією, її можна розглядати як чисте завдання оптимізації. Є функція пристосованості і потрібно знайти точку її максимуму. Проблема в тому, що ніхто не знає, як виглядає функція і можна тільки робити припущення, обираючи (симулюючи) точки навколо поточного рішення [12].

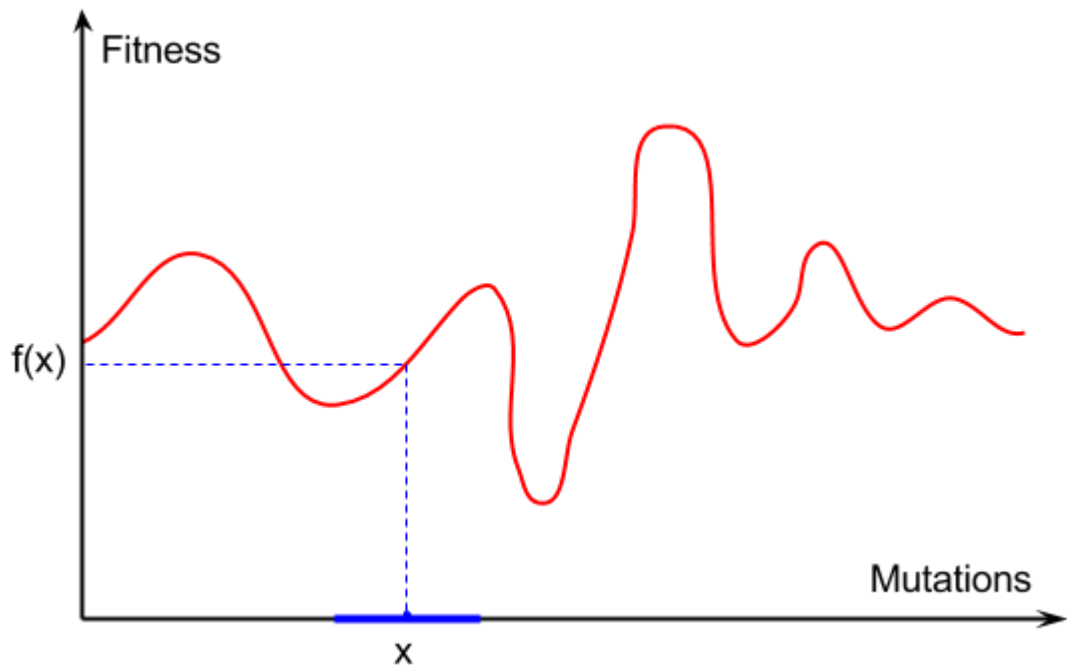


Рисунок 2.8 – Локальний максимум

На цьому графіку видно, як різні мутації (на осі X) призводять до різних оцінок пристосованості (вісь Y). Еволюція займається вибіркою випадкових точок в локальній околиці, намагаючись збільшити пристосованість. Якщо врахувати розмір околиці, то для досягнення максимуму потрібно безліч поколінь. І на цьому етапі все стає складним. Якщо околиця занадто мала, еволюція може застрягти в локальних максимумах. Це рішення, які оптимальні локально, але не глобально. Дуже часто локальні максимуми виявляються посередині западин пристосованості. Коли таке відбувається, еволюція може і не знайти поблизу більш гарне рішення, після чого застрягти. Часто кажуть, що таргани застрягли в еволюційних локальних максимумах (Can An Animal Stop Evolving?): Вони неймовірно успішні в своєму способі життя, і будь-які довготривалі поліпшення вимагають занадто високою короткочасною розплати зниженням їх пристосованості.

Еволюційне програмування має певні недоліки, але їх можна позбутися використовуючи наступні покращення:

Кращі  $K$  геноми. Геноми, що починають наступне покоління, є кращими з усіх поколінь. Це означає, що якщо поточне покоління не може поліпшити пристосованість, то наступне покоління почнеться з того ж генома. Очевидна пастка тут в тому, що можна застрягти в циклі, в якому один і той же геном симулюється знову і знову, без будь-яких розумних поліпшень. Можливим рішенням було б завжди брати кращий геном з поточного прогону. Недолік такого підходу в тому, що пристосованість може знизитися, якщо жодна з мутацій не принесе поліпшень. Більш розумним підходом став би вибір кращих  $k$  геномів з кожного покоління, і використання їх в якості основи наступної ітерації. Можна додати немутіровавший кращий геном попереднього покоління, щоб пристосування не погіршилася, але була можливість і для пошуку кращих рішень.

Адаптивна швидкість навчання. Під час кожної мутації геному змінюється тільки один параметр на певну величину. Кількість виконаних мутацій (швидкість навчання) генома повинна бути пов'язана зі швидкістю, з якою покращується оцінка.

Раннє завершення. Легко побачити, що деякі симуляції можуть завести у глухий кут. Хороший алгоритм повинен розпізнавати їх і переривати такі симуляції для економії ресурсів.

Зміна функції пристосованості. Якщо те, чого розробник навчає агентів дається досить складно, то можливо варто робити це поетапно. Це можна реалізувати, поступово міняючи функцію пристосованості. Так можна зосередити зусилля по навчанню на простого завдання.

Кілька перевірок. Коли справа доходить до симуляції фізики, то є ймовірність, що можна ніколи не отримати одного результату двічі. Є хороший спосіб – створити кілька примірників агента з одним геномом в одних і тих же умовах, а потім усереднити їх оцінки, щоб отримати більш надійну оцінку.

### 2.5.3 Метод Монте-Карло

Під методом Монте-Карло розуміється чисельний метод рішення математичних задач за допомогою моделювання випадкових величин.

Агентне моделювання можна розглядати у якості певної математичної моделі, а окремі інтелектуальні агенти та родини агентів – об'єктами цієї моделі.

Великою перевагою методу Монте-Карло є те, що він дозволяє врахувати в моделі елемент випадковості і складність реального світу. Крім того, метод є робастним по відношенню до зміни різних параметрів, таких як розподіл випадкової величини. В його основі лежить закон великих чисел.

Одним з типових прикладів використання методу Монте-Карло є завдання, в яких необхідно знайти математичне очікування деякої випадкової величини (стосовно агентного моделювання це може бути завдання пошуку часу роботи певних інтелектуальних агентів). Для цього потрібно згенерувати набір випадкових значень цієї величини і знайти середнє. Випадкова величина зазвичай характеризується певним розподілом ймовірностей [11, 12].

Процес моделювання включає наступні етапи:

- Визначення моделі або алгоритму, які найбільш точно описують поведінку досліджуваної системи.

- Багаторазове застосування моделі з використанням генератора випадкових чисел для отримання вихідних даних моделі (моделювання системи). При необхідності моделюють вплив невизначеності. Модель записують у формі рівняння, що виражає співвідношення між вхідними та вихідними параметрами. Значення, відібрані в якості вхідних даних, отримують виходячи з відповідних розподілів ймовірностей, що характеризують невизначеності даних.

- За допомогою комп'ютера багаторазово використовують модель (часто до 10000 раз) з різними вхідними даними і отримують вихідні дані.

Вони можуть бути оброблені за допомогою статистичних методів для отримання оцінок середнього, стандартного відхилення, довірчих інтервалів.

Вихідними даними можуть бути значення характеристик або розподіл ймовірності або частоти відмови, або виходом може бути ідентифікація основних функцій моделі, які роблять основний вплив на вихідні дані.

Метод Монте-Карло зазвичай використовують для оцінки розподілу вхідних або вихідних результатів або характеристик розподілу, в тому числі для оцінки:

- ймовірності встановлених станів;
- значень вихідних величин, для яких встановлено межі, відповідні деякому рівні довіри, які не повинні бути порушені.

Аналіз взаємозв'язку вхідних і вихідних величин може виявити відносне значення факторів роботи системи та ідентифікувати способи зниження невизначеності вихідних величин.

## 3 ІНТЕЛЕКТУАЛІЗАЦІЯ НАПІВАВТОМАТИЧНОГО МЕТОДУ УПРАВЛІННЯ ДЛЯ SCADA-СИСТЕМИ НА ПРИКЛАДІ СУЧАСНОГО ЕЛЕВАТОРА

### 3.1 Технологічний процес сучасного елеватора

У атестаційній роботі у якості реальної моделі для інтелектуалізації напівавтоматичного методу та для розробки моделі моніторингу був використаний технологічний процес реального елеватора.

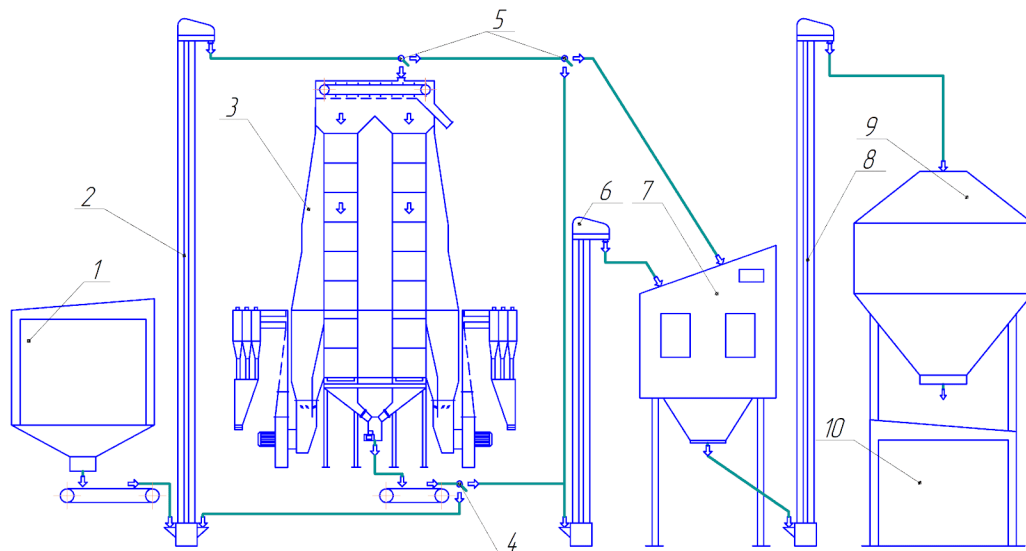


Рисунок 3.1 – Схема комплексу зберігання та обробки зерна

Алгоритм роботи комплексу являє собою наступне: зерно, яке надійшло до комплексу (рисунок 3.1), зважується на вагах (9) і направляється в завальну яму (1). Далі по стрічкових транспортерах зерно з завальної ями (1) направляється в норію сирого зерна, де скидається в самоплив. Залежно від положення рейкових засувки (5) направляється в сушарку (3) на сушку з подальшим очищенням або безпосередньо на очистку. Зерно з сушарки, в залежності від вологості, за допомогою перекидного клапана (4)

направляється, або на рециркуляцію в сушарку (3), або в норію сухого зерна (6). Зерно, яке не потребує сушіння, направляється безпосередньо на очистку, минаючи сушилку. Очищення зерна відбувається в приміщенні сепараторної (7), на сепараторах БСХ-100. Очищене зерно по самопливах направляється в норію готової продукції (8). Далі по самопливах в бункери готової продукції (9). Заповнення бункерів відбувається послідовно. Спочатку основний, потім резервний бункер. Розподіл зерна в бункерах здійснюється завантажувальним скребковим транспортером. Повне заповнення бункерів готової продукції контролюється датчиком рівня пов'язаним з автоматизованою системою управління комплексом. Відвантаження та одночасне зважування готової продукції відбувається в приміщенні вагової (10) [1].

Технологічна схема об'єкта для більш детального уявлення про систему представлена на рисунку 3.2

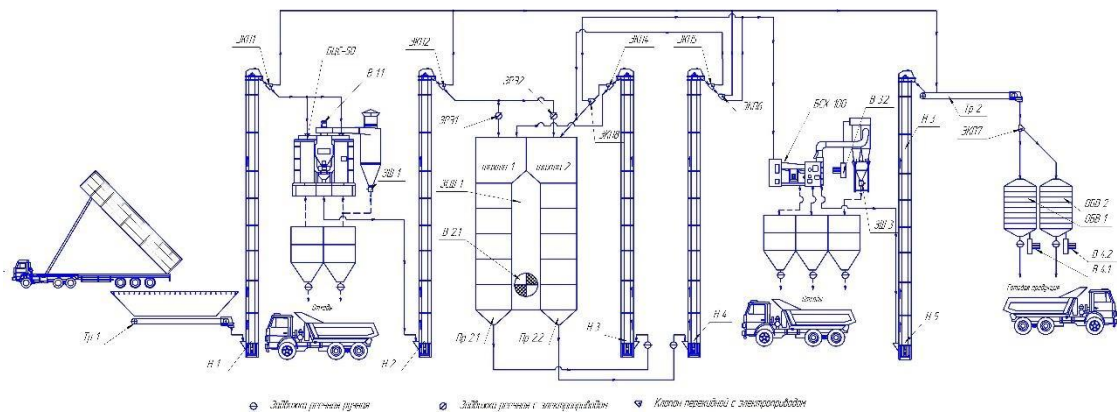


Рисунок 3.2 – Технологічна схема зерносушильного комплексу

Робота комплексу відбувається у двох режимах: ручному та автоматичному, тому для покращення працездатності та спрощення роботи оператора системи було обрано модернізацію напівавтоматичного методу.

### 3.2 Поняття інтелектуалізації стосовно моделей та методів управління

Високий ступінь автономності роботи штучної системи дозволяє їй самостійно приймати рішення в рамках своєї функціональності та закладеної експертності. Можливість адаптації до постійно змінюваних умов довкілля під час експлуатації або постійно змінюваних умов самої експлуатації дозволяє штучній системі навчатися (самостійно або за допомогою людини) і продовжувати функціонувати. Пропонується вважати, що чим вище ступінь автономності та адаптивності штучної системи, тим вище її інтелектуальність та здатність до подальшого розвитку [1, 13].

З іншого боку, під інтелектуалізацією розуміється процес підвищення ступеня інтелектуальності технічної системи (наприклад, такої, як комплекс «Сваркове», використаний у роботі). Виходячи з того, що інтелектуальність є складовою характеристикою з двох більш простих властивостей, – інтелектуалізація є траєкторією розвитку системи в рамках свого життєвого циклу від низького рівня до високого за двовимірним простором станів. Якщо розглянути відносні шкали для опису та класифікації технічних систем за ступенями їх адаптивності та автономності, то можна уявити двовимірний простір (можливо, навіть, безперервний), де кожній певній технічній системі з заданими ступенями адаптивності і автономності буде відповідати деяка точка, що лежить в квадрантах від точки  $(0, 0)$  – повністю неадаптивна і неавтономна система до точки  $(1, 1)$  – абсолютно адаптивна і автономна система (рисунок 3.3).

У якості типових прикладів систем, які знаходяться в квадрантах двовимірного класифікатора інтелектуальності систем, можна навести такі:

1. Робот-маніпулятор на промисловому виробництві – зазвичай такий робот налаштований на виконання заданої послідовності дій (низька адаптивність) та не призначений для створення будь-яких рішень (відсутність автономності).

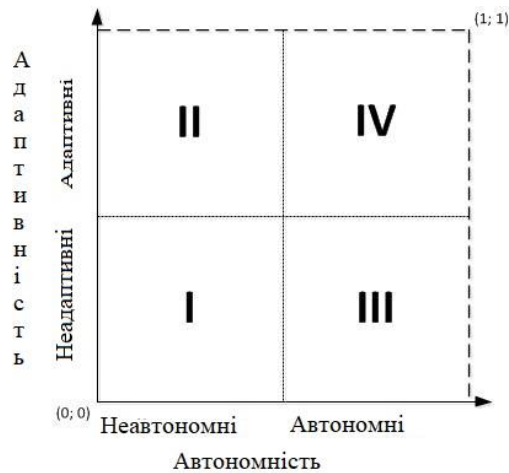


Рисунок 3.3 – Квадранти для класифікації технічних систем по адаптивності і автономності для виявлення ступеня інтелектуальності систем

2. Робот-пилосос – цей робот під час попадання в незнайоме оточення обстежує новий простір і будує його модель для подальшого використання у своїй роботі (висока адаптивність), але при цьому виконує лише одну функцію – прибирання приміщення з запуском по команді власника або по триггеру в календарі (низька автономність).

3. Система підтримки прийняття рішень в заданій проблемній області – така система зазвичай вже налаштована під певні ситуації, і додавання нової ситуації потребує суттєвого доопрацювання системи (низька адаптивність), при цьому система сама пропонує рішення і пояснює їх (висока автономність).

4. Персональний помічник – система для планування, нагадування, виконання рутинних дій, яка постійно навчається у взаємодії зі своїм користувачем (висока адаптивність) і при цьому може самостійно запускати різні сценарії спілкування зі своїм користувачем, і в рамках такого спілкування самостійно робити вибір на користь того чи іншого варіанта вирішення на основі минулих уподобань користувача (висока автономність) [13, 16].

Таким чином, інтелектуалізація являє собою процес переведення

(еволюції) технічної системи з її поточного становища в просторі «Адаптивність – Автономність» в четвертий квадрант, що є якомога ближче до точки (1;1). З квадрантів II і III такий перехід може бути здійснений безпосередньо, в той час як з квадранта I може існувати три можливих траєкторії інтелектуалізації системи (рисунок 3.4).

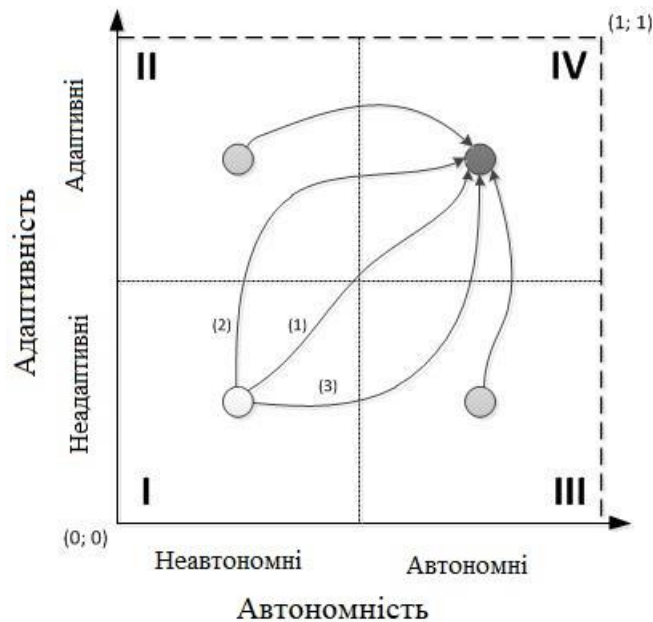


Рисунок 3.4 – Можливі траєкторії інтелектуалізації технічних систем

Традиційною методологією наукового пізнання, яка застосовується для роботи з технічними системами, є системний підхід. Цей методологічний інструментарій пропонує широкі можливості по аналізу і синтезу складних систем практично будь-якої природи. Разом з тим, незважаючи на гадану універсальність, системний підхід має істотний недолік – системи розглядаються в якості взаємопов'язаного набору елементів, під яким часто розуміються деякі об'єкти у відриві від загального призначення системи, її емерджентні виникають функції і їх взаємозв'язку з функціональністю кожного елемента [16, 17].

Функціональний підхід розроблений в рамках декларативного програмування та заснований на понятті математичної функції.

Функціональне програмування оперує функціями, як базовими примітивами для опису алгоритмів. Це дозволяє розглядати обчислювальні процеси в якості послідовності викликів функцій, кожна з яких може виражатися через атомарні дії та виклик інших функцій, у тому числі і самих себе при організації рекурсивних викликів.

### 3.2 Процес інтелектуалізації

У цілому сутністю функціонального підходу в рамках теорії та практики управління є розгляд цілей і потреб через призму безлічі функцій, які необхідно виконати для досягнення цілей чи задоволення потреб. Під час специфікації цієї безлічі функцій реалізується один або кілька суб'єктів для виконання цих функцій, після чого вибирається такий функціональний суб'єкт, для роботи якого потрібна мінімальна кількість сукупних витрат протягом усього життєвого циклу об'єкта управління на одиницю корисного ефекту.

Якщо розглядати тільки штучні технічні системи довільної складності, то функціональний підхід можна спростити до аналізу і синтезу функціональної структури системи. Для цього здійснюється розгляд комплексу функцій системи і формування відповідної функціональної структури (таку роботу було зроблено під час розгляду системи «Сваркове»). Процес проектування або вдосконалення технічної системи, окремих її підсистем і елементів являє собою моделювання функціональної структури в такій послідовності:

- формування «дерева» функцій системи;
- декомпозиція «дерева» функцій системи до рівня набору базових функцій;
- формування моделі функціональної структури системи;
- формування морфологічних моделей системи;
- багатокритерійна оцінка і вибір кращих варіантів функціональної

структури системи.

Виконання перерахованих вище кроків формує так званий функціональний опис системи. На самому верхньому рівні цей опис являє собою перелік функцій системи, які можуть бути навіть не пов'язані між собою. Однак на першому ж рівні декомпозиції починають з'являтися зв'язки між функціями, самі функції прив'язуються до підсистем і елементів, і вся проєктована або аналізована технічна система наповнюється функціональним змістом. При цьому на кожному рівні кожна функція відображає призначення або окремий аспект такого призначення системи в своїй надсистемі та являє собою відображення робочої дії, виконання якої покладається на систему при заданих вимогах, умови та обмеження. У результаті такої декомпозиції з'являється специфічна ієрархія функцій, загальну структуру якої можна пояснити за допомогою наступної схеми, показаної на рисунку 3.5.

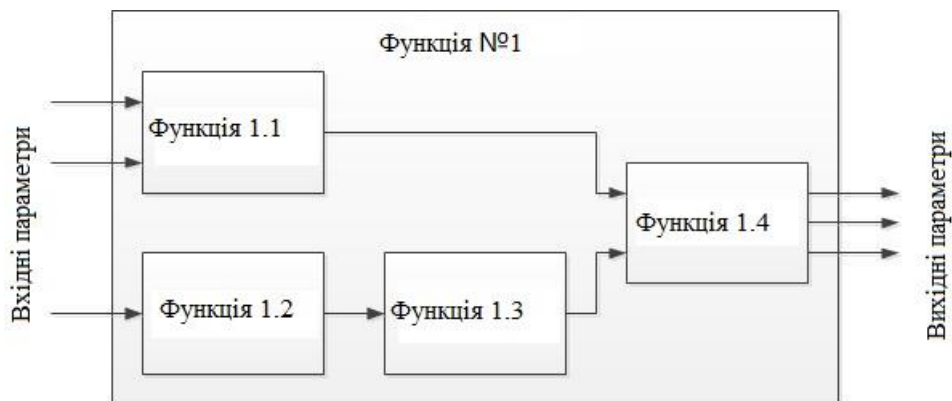


Рисунок 3.5 – Приклад схеми роботи функціонального підходу

Представлена схема демонструє, що між функціями є два типи відносин:

1. Включення. Наприклад, функція 1, визначена через виклики чотирьох функцій, – більш низького рівня декомпозиції.

2. Передача результату. Наприклад, функція 1.1 передає результат своїх обчислень до функції 1.4 у якості першого параметра.

Таким чином, функціональний підхід під час проектування технічних систем ґрунтується на припущенні про те, що увага проектувальника має бути сконцентровано на завданнях та функціях, що виконуються для їх вирішення, а не на об'єктах або абстракціях, як існуючих, так і запланованих до реалізації.

Функціональний підхід передбачає, що будь-яка технічна система, яка проектується для подальшого впровадження та експлуатації, повинна виконувати будь-які корисні дії і надавати корисну функції. Якщо, наприклад, здійснюється автоматизація деякої діяльності, то потрібно розуміти сенс і мету цієї діяльності, призначення системи та її функціональний взаємозв'язок із суміжними системами, своєю надсистемою і середовищем, де вона функціонує. При цьому необхідно оптимально розподілити функції між людиною і комплексом засобів автоматизації, оскільки сам сенс автоматизації полягає у тому, щоб рутинні, складні або небезпечні для людини дії виконували засоби автоматизації.

Загальна схема автоматизованої системи управління для комплексу «Сваркове» може бути представлена схемою, показаною на рисунку 3.6

Фактично, ця схема в іншому аспекті показує те ж саме, що зображено на рисунку 3.5, але при цьому з середовищем тепер взаємодіє деяка система управління.

Сенсори, сама система управління і виконавчі пристрої – все це можна сприймати як деякі функції, через які здійснюються реактивний вплив на середовище.

Разом з тим, це саме та активна схема управління, до якої потрібно внести зміни для підвищення ступеня інтелектуальності. Інтелектуалізація представленої схеми управління полягає у нарощуванні додаткового контуру прийняття рішень, робота якого заснована на двох додаткових можливостях – машинному навчанні (у випадку комплексу «Сваркове» у якості машинне навчання було замінене інтелектуальними агентами) та логічному висновку.



Рисунок 3.6 – Загальна схема автоматизованої системи управління

Розширена (інтелектуалізована) схема управління показана на рисунку 3.7



Рисунок 3.7 – Загальна схема інтелектуальної системи управління

Як видно, єдина система управління розбита на дві підсистеми:

1. Реактивна підсистема управління – фактично, реалізує традиційну схему управління, коли сигнали з сенсорів обробляються системою управління і по них здійснюється формування управлінських впливів на середовище (об'єкт управління) через виконавчі пристрої.

2. Проактивна підсистема управління – надає додатковий проміжний рівень, який дозволяє здійснювати навчання системи, побудову прогнозу на основі моделювання середовища і своєї поведінки у ній та побудова плану дій і порівняння факту з прогнозом і планом для здійснення навчання (адаптації) системи до постійно змінюваних умов середовища.

Ці підсистеми пов'язані один з одним за допомогою передачі фокусу управління. Коли проактивна підсистема створює новий патерн поведінки системи в умовах, що змінилися, і середовище зафіксувалося, створений варіант управлінського впливу спускається до реактивної системи та відбувається автоматизація реакції, оскільки, фактично, для незмінних умов діяльності навчання проактивна поведінка не потрібна, і тому реактивна реакція буде працювати швидше. Якщо ж в процесі «рефлекторної» дії реактивної системи виявляються зміни у середовищі або об'єкті управління, то реактивна система ставить фокус уваги на проактивну дію для обробки мінливих умов і вироблення нових правил і патернів поведінки.

Цикл управління у рамках такої інтелектуалізованої системи управління полягає в послідовному виконанні наступних кроків:

1. Збір вхідної інформації з усіх сенсорів, які здійснюють моніторинг різних параметрів об'єкта управління і середовища, де функціонує система. Кожен тип сенсора в цьому випадку є окремою індивідуальною модальністю сприйняття системи.

2. Сенсори очищають вхідну інформацію від шумів і здійснюють перший вибір шляху подальшої обробки. Якщо вхідна інформація з сенсорів відповідає будь-яким автоматичним патернам поведінки системи, то фокус управління передається в реактивну підсистему, яка вибирає конкретний патерн і виконує його. Однак, якщо в процесі реагування за реактивним сценарієм система виявляє, що щось пішло не так, здійснюється ескалація на проактивний підсистему управління, як було описано раніше.

3. Якщо вхідна інформація не має автоматичної реакції для своєї обробки, то здійснюється інтеграція всіх модальностей сприйняття системи в

єдиний блок опису об'єкта управління і середовища. На виході у цьому модулі з'являється цілісна картина сприйняття, яка передається в проактивну підсистему управління.

4. Проактивна підсистема управління приймає рішення на основі наявних у неї динамічних моделей власно себе, об'єкта управління і середовища. Тут повинні використовуватися деякі методи машинного навчання та парадигми штучного інтелекту. На виході у проактивній підсистемі управління з'являється управлінський вплив, що записується в реактивну підсистему в якості нового правила або сценарію, а також направляється на виконання.

5. Для виконання управлінський вплив перекладається на мову конкретних виконавчих пристроїв, які взаємодіють з об'єктом управління і середовищем. Виконавчі пристрої виконують команду. Цикл роботи завершується.

Разом з тим в описаній схемі відсутній важливий компонент інтелектуальності – петлі контролю свого власного стану від кожного елемента інтелектуальної системи управління. Додавання таких петель здійснюється через сенсори – у наборі модальностей сприйняття з'являються внутрішні модальності. Залучення сенсорів необхідно для очищення внутрішньої інформації, яка так само, як і будь-яка інша, може бути зашумлена.

Інтерес викликає те, що система саме з такою архітектурою при переході через певний поріг складності може вважатися розумною. Дійсно, відповідно до визначення, даного нейрофізіологом Е. Р. Джоном, свідомість – це «процес, в ході якого інформація про множинних індивідуальних модальностей сприйняття і відчуття зводиться в єдине багатопланове уявлення про стан системи і її оточення і інтегрується з інформацією про спогади і потребах організму, породжуючи емоційні реакції і програми поведінки, що сприяють пристосуванню організму до його оточенню».

Представлена на рисунку 3.7 схема описує те, як функціонує нервова

система людини і вищих тварин, але на прикладі системи, що використовувалась у роботі. Адже «розумність» в цьому сенсі визначається як адекватне реагування не тільки на стимули зовнішнього середовища, а й на внутрішні стани, що також включає в себе постійний моніторинг стану власної підсистеми управління, що називається «саморефлексія», яка і призводить до усвідомлення. Таким чином, при розгляді деякої заданої технічної або кіберфізической системи для її інтелектуалізації необхідно здійснити наступні кроки:

1. Функціональний аналіз. Необхідно побудувати якомога повну модель функціонування системи з декомпозицією функцій і розподілом їх по підсистемах і елементах. Це дозволить розподілити наявні функції по сенсорах, виконавчим пристроям та реактивній підсистемі управління.

2. Проектування на основі системотехніки. За заданим цільовим станом системи та її призначенням у рамках її надсистеми необхідно спроектувати оптимальну структуру з урахуванням виявленої на попередньому кроці функціональності. При цьому в процесі проектування якісь функції можуть бути змінені або виключені, а якісь введені до складу системи. На цьому кроці необхідно побудувати проактивну підсистему управління з додатковими елементами, що здійснюють інтеграцію інформації з сенсорів, трансляцію управлінських впливів і взаємодіють з реактивною підсистемою управління.

3. Доповнення проекту на основі комплексотехніки. При розгляді системи в комплексі з її оточенням, а особливо з надсистемою і зовнішньою організаційною системою управління (якщо така існує і використовується), необхідно розглянути додаткову функціональність системи, що вивчається так, щоб нові функції мали певну цінність на більш високих рівнях застосування. Це дозволить вивести процес підвищення ефективності та оптимальності на більш високий рівень розгляду і не допустити прояви ситуацій, коли інтелектуалізація системи і підвищення ефективності її функціонування призводить до зниження ефективності в цілому, в рамках

комплексу систем.

4. Функціональний синтез. У підсумку вся оптимізована функціональність інтелектуалізованої системи збирається знову і вибудовується у вигляді функціональних схем взаємодії системи з її оточенням, яке включає надсистему, всі суміжні системи і навколишнє середовище. Результатом процесу повинна стати система з підвищеними рівнями автономності функціонування і адаптивності до змін зовнішніх умов діяльності.

Іншими словами, рішення задачі інтелектуалізації складних технічних (і соціотехнічних), а також кіберфізичних систем засновано на використанні методів функціонального аналізу, системотехніки, комплексотехніки і функціонального синтезу. Процес рішення задачі інтелектуалізації можна проілюструвати за допомогою схеми, показаної на рисунку 3.7



Рисунок 3.8 – Процес рішення задачі інтелектуалізації заданої системи

Саме така схема використовувалась у роботі для автоматизації технологічних процесів. Практично будь-який автоматизований технологічний процес можна укласти в описану схему, так як будь-який процес укладається в шаблон, показаний на рисунку 3.8.

Здійснення інтелектуалізації засноване на внесеннях до системи управління технологічним процесом моделей об'єкта управління, середовища функціонування і самої системи управління для прогнозування і планування

управлінських впливів з подальшим самонавчанням або навчанням в автоматизованому режимі на основі порівняння прогнозу, плану і факту.

### 3.3 Модель інтелектуалізації для сучасної SCADA-системи

Будь-яка система являє собою певний проєкт з особистим набором характеристик та параметрів. Відповідно до РМВоК 6 виділяють кілька областей знань управління сами такими проєктами. У кожній з областей проєкт розглядається з різних боків, виділяються сутності/об'єкти, методи управління та їх вплив на проєкт, як на спосіб організації роботи для досягнення конкретної мети або виконання завдання. Далі буде надано опис типових об'єктів, які існують у комплексі «Сваркове», та які можна виділити під час управління проєктами, їх характеристики, взаємозв'язок, а також загальну механіку імітаційного інтелектуального моделювання та його відповідність до життєвого циклу проєкту. Типові об'єкти та їх характеристики:

1. Проєкт має наступні характеристики: керівник, найменування, тип, планована дата початку, фактична дата початку, планована дата закінчення, фактична дата закінчення, поточний стан життєвого циклу, початковий баланс проєкту, поточний баланс проєкту. Розрахункові або визначальні компоненти визначаються на підставі інших об'єктів характеристики: команда проєкту, відсоток виконаного обсягу робіт, відставання або випередження за обсягом виконаних робіт, відставання або випередження за термінами, планована вартість.

2. Завдання/Робота – тут вказуються схожі характеристики з проєктом, до яких додаються такі: приймальник, відповідальний виконавець, тип виконуваної роботи, проєкт, місце, відсоток готовності. Розрахункові або визначальні компоненти визначаються на підставі інших об'єктів характеристики: послідовність виконання всередині проєкту, склад виконавців, історія зміни стану, вартість виконання завдання/роботи.

3. Матеріальний ресурс (основні засоби): тип об'єкта, дата постановки на облік, дата введення в експлуатацію, назва, балансова вартість. Розрахункові або визначальні компоненти: амортизація, поточний стан, де задіяний зараз, розклад використання.

4. Споживаний ресурс (сировина, запасні частини): тип ресурсу, початкові запаси, місце розташування, дата поставки, термін придатності. Розрахункові або визначальні компоненти: поточні запаси, інтенсивність витрачання

5. Персонал: ПІБ, постійне проживання. Розрахункові або визначальні компоненти: доступність для роботи, сумісність з іншими співробітниками, поточне розташування на час виконання роботи, де задіяний, розклад роботи.

6. Ризик: ймовірність виникнення, ціна збитку, опис, тривалість впливу, індикатор спрацьовування ризику. Розрахункові або визначальні компоненти: заходи щодо усунення наслідків, заходи щодо недопущення виникнення або ухилення, вартість, терміни реалізації.

На рисунку 3.9 представлена схема взаємозв'язків та залежностей, розроблена відповідно до вищезазначених пунктів.

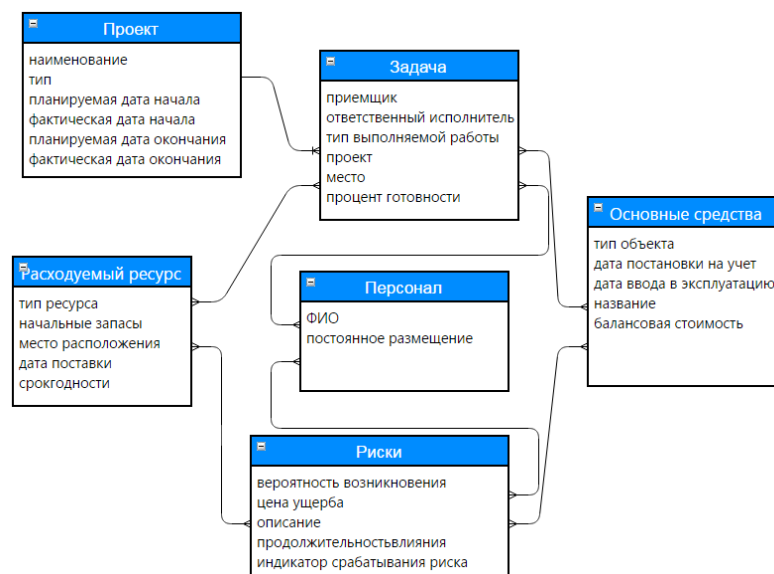


Рисунок 3.9 – Схема взаємозв'язків та залежностей

Проект-[1: M]-завдання – виконуються в обмеженнях термінів проекту.

Задача [1: M]-завдання – можуть мати ієрархічний зв'язок (вертикальний), можуть мати зв'язок у вигляді вказівки послідовності виконання (горизонтальний).

Матеріальний ресурс-[M: M]-завдання – прив'язується через ставлення розкладу до задачі із зазначенням розкладу використання.

Споживаний ресурс-[M: M]-завдання – прив'язується через ставлення розкладу до задачі із зазначенням необхідного запасу для її виконання.

Персонал-[M: M]-завдання – можуть бути задіяні в рамках декількох завдань, для чого вказується розклад робіт і відсоток використання в задачі.

Ризик [M: M]-[Об'єкт] – при вказівці взаємозв'язку з [Об'єктом] вказується ймовірність виникнення.

Представлена схема (рисунок 3.8) демонструє типи взаємозв'язків та залежностей, що використовувалися під час процесу інтелектуалізації. На кожному з етапів були розроблені окремі інтелектуальні агенти та моделі агентно-орієнтованих систем, що беруть на себе роботу оператора SCADA-системи.

### 3.3.1 Модель інтелектуалізації

SCADA-система, що була використана у роботі використовує під час роботи два методи: повністю автоматичний та повністю ручний. Для спрощення роботи як автоматичного, так і ручного методу управління було покращено вже існуючий напівавтоматичний метод та розроблено модель управління з елементами штучного інтелекту (рисунок 3.10).

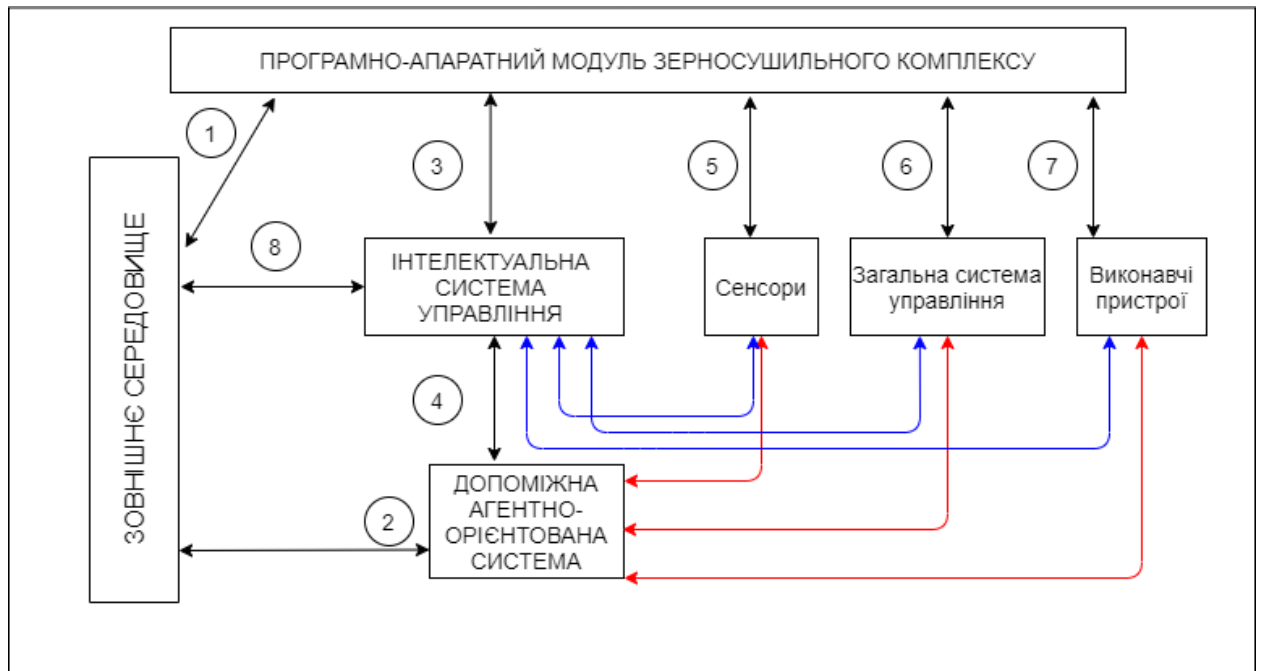


Рисунок 3.10 – Модель інтелектуалізованої системи моніторингу

Програмно-апаратний модуль відповідає за роботу зерносушильного комплексу, отже він повністю пов'язаний із процесом інтелектуалізації. У якості інтелектуальної системи управління на рисунку та у моделі використовувалась схема, представлена на рисунку 3.7. До програмно-апаратного модуля входять сенсори, загальна система управління та виконавчі пристрої, аде на моделі вони представлені окремо, для того, щоб показати їх власну взаємодію із інтелектуальною системою, програмно-апаратним модулем та допоміжною агентно-орієнтованою системою.

У якості допоміжної агентно-орієнтованої системи представлені окремі типи агентів та окремі родини, що утворюють агенти у процесі навчання та донавчання, які виступають у ролі помічників основної інтелектуальної системи, але вони також мають власний зв'язок із сенсорами, виконавчими пристроями та загальною системою управління. Це розроблено для того, щоб навчання цих окремих агентів не заважало роботі основної інтелектуальної системи, а працювало тільки їй на користь.

Автоматизація збору інформації з програмно-апаратного модуля

зерносушильного комплексу – інтелектуальна система управління за допомогою власних внутрішніх компонентів та агентно-орієнтованої системи отримує дані, аналізує їх, ділить їх на категорії (все це архівується на хмарний сервер) та визначає корисність отриманих та проаналізованих даних.

Стрілки з номерами 1-8 показують та описують взаємодію елементів, як між собою, так із зовнішнім середовищем, оскільки навчання агентів та робота інтелектуальної системи не може відбуватися без взаємодії із зовнішнім середовищем. Червоні та сині стрілки описують взаємодію окремо виведених компонентів із двома системами.

Між собою інтелектуальна та агента система також взаємодіють: агентна система допомагає інтелектуальній новими агентами або старими донавченими для поліпшення працездатності всього комплексу в цілому, а інтелектуальна система надає агентній нову інформацію щодо стану програмно-апаратного модулю.

### 3.3.2 Механіка процесу інтелектуалізації

Кожен такт моделювання відповідає фіксованому часу – 1 день/годину виконуваного проекту. Для цього приймемо всі терміни, і інтервали в проекті – кратними величині 1 день/годину. Схема циклу моделювання зображена далі:

Цикл моделювання полягає в наступному:

1. Встановлюються початкові значення симуляцій для проекту. Створюється проект, готується розклад проекту, дерево ризиків. На цьому етапі так само доступні функції інтелектуальної підтримки управління проектами, але цей крок не може бути виконаний без ОПР.

2. Ітерація починається з визначення діючих значень.

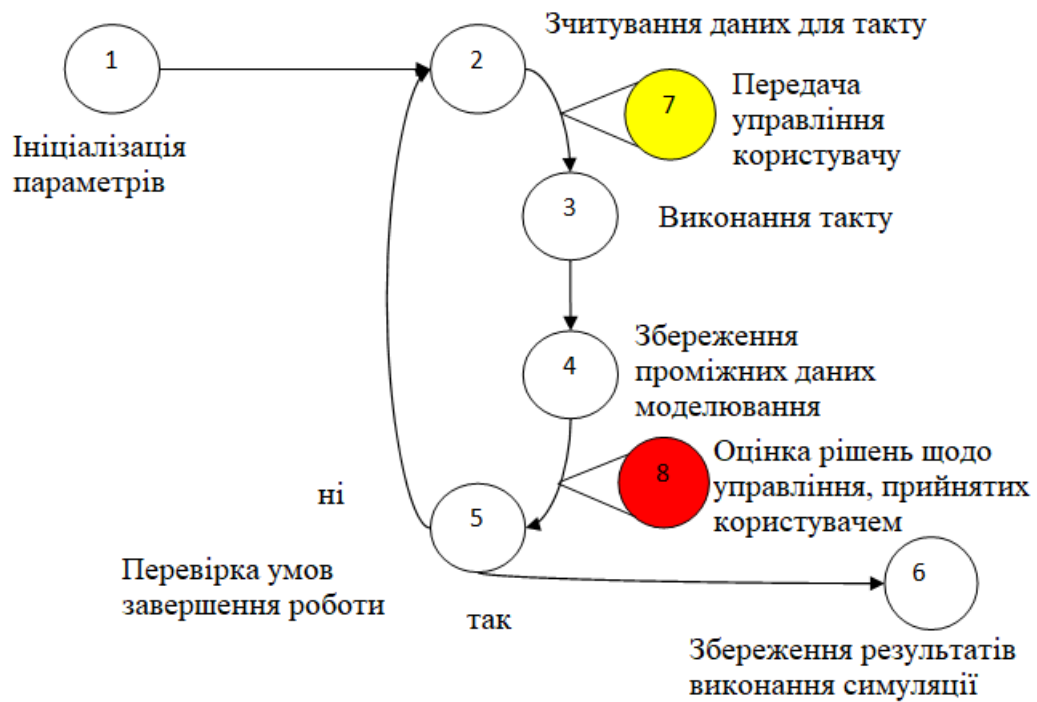


Рисунок 3.11 – Механіка циклу моделювання

3. Виконання такту. Кожен такт моделювання виконуються наступні операції:

- витрачаються ресурси за завданнями,
- перевіряється ймовірність відмов (ризиків),
- виконується певний обсяг робіт з переліку робіт за проектом,
- виконуються фінансові операції по проекту.

4. Зберігаються розраховані значення для певного такту

5. Перевірка умов завершення моделювання.

6. Завершення моделювання та виведення результатів (аналітичних, агрегованих і докладних значень по кроках моделювання). При закінченні моделювання зберігаються останні (підсумкові) значення і причини припинення моделювання.

7. Видача користувачеві (або особі, що приймає рішення – ЛПР) інформації про стан проекту без використання оптимізацій, модулів аналітики та підтримки прийняття рішень. Від користувача необхідна реакція на поточний стан (за необхідності) або продовження моделювання.

8. Оцінка управлінських рішень користувача на основі поточних значень, а так само ретроспективи їх зміни і прийнятих користувачем управлінських рішень із застосуванням алгоритмів оптимізацій, модулів аналітики та підтримки прийняття рішень.

Всі дані проміжних кроків симуляції зберігаються і накопичуються в межах поточної симуляції. При подальшій роботі оптимізаційних алгоритмів (на 8 кроці циклу симуляції) можуть використовуватися дані як поточних, так і попередніх завершених симуляцій (з поправкою на результат завершення симуляції).

При декількох одночасно виконуваних роботах проекту симуляція для них виконується паралельно (тобто симулюється одночасне виконання), в разі відсутності розбіжностей по використовуваним ресурсам.

При декількох співробітників/типах ресурсів моделювання виконується для кожного з них паралельно (тобто витрачаються одночасно), в разі відсутності розбіжностей по використовуваних ресурсів.

### 3.3.3 Логіка управління

Для кожної з областей знань управління проектами існують постановки завдань і описані математичні методи їх вирішення. Залежно від моделі управління знання цих правил і способів вирішення завдань повинні перерозподілятися між системою і користувачем. Моделі управління виділені наступні:

- управління з повідомленнями – система не впливає на об'єкт (проект), але відображає повідомлення про зміни показників і можливості виконання дій (прийняття рішень і максимум знань потрібно від ОПР);
- інтерактивне управління – система пропонує управлінський вплив, але рішення залишається за ОПР (прийняття рішень залишається за ОПР);
- евристичне управління – система приймає рішення і виконує деякі дії самостійно (ОПР виключається з процесу управління).

Реалізація самого управління полягає в моніторингу та аналізі сукупності характеристик проекту і оцінці їх відхилення від «нормальних» для даного часу, з урахуванням динаміки їх зміни. Керуючі впливи підбираються на основі отриманих даних (тобто при наявності відповідності такої комбінації характеристик будь-якого впливу), а також аналізуються схожі проекти зі схожими ситуаціями і прийняті в них рішення. Відповідно до ступеня або рівня відхилення можуть застосовуватися ті чи інші способи впливу:

1. Перерозподіл ресурсів між завданнями;
2. Перерозподіл трудових ресурсів між завданнями;
3. Зміна розкладу виконання завдань;
4. Планування закупівель;
5. Ухилення або вжиття заходів щодо ліквідації наслідків ризиків.

Для способів впливу важливі такі характеристики: ступінь відповідності ситуації, тривалість реалізації, вартість реалізації, можливий час початку реалізації. Для визначення застосовуваного способу впливу важливо:

1. Зазначені експертами характеристики.
2. Наявність інформації в накопиченій базі виконаних проектів.

Дані механізми логічно будувати із застосуванням нейронних мереж і нечіткої логіки. Використовувати ці алгоритми можна як на етапі ініціалізації і планування проекту, так і на етапі його реалізації. Можливо виконання аналізу - як змінюватися характеристики після застосування керуючого впливу.

### 3.3.4 Інтелектуалізація імітації

На етапі виконання такту можливо повне виключення ОПР з процесу управління. Для моделювання подій потрібні уточнення деяких характеристик (наближені). Для виконання дій, що управляють система

повинна «знати» деяку додаткову інформацію щодо предметної області, наприклад:

#### 1. Перерозподіл ресурсів між завданнями:

- взаємозамінність ресурсів – можна задати таблицями-матрицями відповідності;
  - ймовірність виходу з ладу ресурсів – вказується ймовірність в діапазоні від  $X_{\min}$  до  $X_{\max}$ ;
  - можливість паралельного використання декількома виконавцями
- як логічне властивість завдання.

#### 2. Перерозподіл трудових ресурсів між завданнями:

- взаємозамінність і несумісність персоналу – можна задати таблицями-матрицями відповідності;
- продуктивність трудових ресурсів – як розрахункове значення на основі даних про: досвід роботи, віком, підвищення кваліфікації і т.п.
- співвідношення типів виконуваної роботи і необхідних для її виконання навичок - аналогічно вирішується матрицями;
- ймовірність невиходу трудових ресурсів (ймовірність хвороби) - вказується ймовірність в діапазоні від  $X_{\min}$  до  $X_{\max}$ ;
- можливість паралельного виконання однієї роботи декількома виконавцями - як логічне властивість завдання.

#### 3. Зміна розкладу виконання завдань:

- чи можлива припинення завдання, або виконання повинно бути безперервним - як логічне властивість завдання;
- чи входить завдання в «критичний шлях» (тобто терміни її виконання безпосередньо впливають на терміни завершення проекту) - визначається системою «нальоту».

#### 4. Планування закупівель.

- інтенсивність витрачання ресурсу - визначається системою «нальоту».
- можливість закупівлі необхідного обладнання - як логічне

властивість завдання.

#### 5. Ухилення або вжиття заходів щодо ліквідації наслідків ризиків.

- ймовірність відмов обладнання - вказується ймовірність в діапазоні від  $X_{\min}$  до  $X_{\max}$ ;

- можливі варіанти ухилення і ліквідації наслідків - вирішується матрицями або списками відповідності (із зазначенням ступеня відповідності).

Це не вичерпний список завдань. Тут так само необхідно відзначити той факт, що універсального рішення для будь-якого проекту бути не може і, що добре для одного проекту - для іншого смерть. Т.ч. необхідні певні ключові характеристики, їх сукупності, і їх значення, які дозволяли б типізувати і класифікувати, підбираючи схожі проекти для навчання системи, наприклад:

- типи задіяних ресурсів;
- типи поставлених завдань;
- кваліфікація та вміння задіяного персоналу;
- розмір бюджету;
- тривалість проекту;
- успішність проекту;
- кількість учасників і т.д.

Імітаційне моделювання або симуляція управління проектом дасть наступне:

- управління з повідомленнями - можна використовувати як тренування або тестування ОПР на знання певних принципів або вміння вирішувати завдання пов'язані з управлінням проектами.

- інтерактивне управління - відпрацювання деяких практик і перевірка їх на моделі. Що дасть можливість змінити модель для відповідності ситуації або навпаки оцінити володіння методами вирішення завдань УП самому ОПР (самоперевірка).

- евристичне управління - можливість великої кількості запусків

симуляції і накопичення певного досвіду (даних) про ці симуляції для їх подальшого аналізу.

Однак сама імітація і симуляція – не кінцева мета. В результаті накопичення досить точних простих і складних моделей в базі симуляції, розробки та налагодження поведінки імітаційної моделі і модулів, які здійснюють інтерактивне взаємодія і евристичне управління (без ОПР), можливе використання накопичених правил і алгоритмів для управління (або інтелектуальної підтримки управління) реальними проектами (3).

Реалізація такої системи у вигляді SaaS рішення, із залученням деякої кількості учасників, дозволить отримати доступ до досвіду роботи (знеособленому) інших учасників (з можливістю навчання системи).

## ВИСНОВКИ

В атестаційній роботі було визначено концепцію SCADA-систем, поняття інтелектуалізації, поняття агентно-орієнтованого моделювання, надано класифікацію інтелектуальних агентів, розроблено модель моніторингу та покращено напівавтоматичний метод:

- спрощення роботи диспетчера – розроблена допоміжна агентно-орієнтована система навчає та донавчає агентів, що беруть на себе практично всі обов'язки диспетчера (диспетчер відповідає тільки за критичні ситуації та спостерігає за вже автоматизованим процесом);

- автоматизація збору інформації з програмно-апаратного модуля зерносушильного комплексу – інтелектуальна система управління за допомогою власних внутрішніх компонентів та агентно-орієнтованої системи отримує дані, аналізує їх, ділить їх на категорії (все це архівується на хмарний сервер) та визначає корисність отриманих та проаналізованих даних;

- налагоджений зв'язок із зовнішнім середовищем за допомогою взаємодії із агентно-орієнтованою системою, інтелектуальною системою та програмно-апаратним модулем.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Liashenko O. Using the Adaptive Approach in the System of Monitoring the State of Grain Storage Technological Process / O. Liashenko, V. Znaiduk, D. Kazmina. – 2020. – №6. – С. 14.
2. Lasi H., Kemper H.-G., Fettke P., Feld T., Hoffmann M. Industry 4.0. In: Business & Information Systems Engineering 4(6), pp. 239-242.
3. Patrascu M., Dragoicea M. Integrating Services and Agents for Control and Monitoring: Managing Emergencies in Smart Buildings. Service Orientation in Holonic and Multi-Agent Manufacturing and Robotics. Studies in Computational Intelligence. Studies in Computational Intelligence. – Volume 544. – pp. 209-224. doi:10.1007/978-3-319-04735-5\_14. ISBN 978-3-319-04734-8.
4. Constable G., Somerville B. A Century of Innovation: Twenty Engineering Achievements That Transformed Our Lives. Joseph Henry Press. – ISBN 03090-89085.
5. Душкин Р. В. Особенности функционального подхода в управлении внутренней средой интеллектуальных зданий // Прикладная информатика, Том 13, № 6 (78), 2018. – с. 20-31. – ISSN 1993-8314.
6. Душкин Р. В. Почему за гибридными ИИ-системами будущее // Экономические стратегии, № 6 (156), 2018. – с. 84-93. – ISSN 1680-094X.
7. Андреева Е. А., Белкова Е. В., Душкин Р. В., Жарков А. Д., Курочкин Е. А., Левин Н. В., Морозов В. П. Тематический обзор Ассоциации Транспортных Инженеров: системы адаптивного управления дорожным движением и дорожные контроллеры. Вып. № 2/2017. – СПб.: ООО «Издательско-полиграфическая компания «КОСТА», 2017. – 48 с.
8. Russell S. J., Norvig P. Artificial Intelligence: A Modern Approach (2nd ed.), Upper Saddle River, New Jersey: Prentice Hall, ISBN 0-13-790395-2.
9. Многоагентные системы [Электронный ресурс] – Режим доступа до ресурсу: <http://www.aiportal.ru/articles/multiagent-systems/multiagent->

systems.html.

10. Классификация агентов [Электронный ресурс] – Режим доступа до ресурсу: <http://www.aiportal.ru/articles/multiagent-systems/agent-classification.html>.

11. Модель Акторов и C++: что, зачем и как? [Электронный ресурс]. – 2017. – Режим доступа до ресурсу: <https://habr.com/ru/post/322250/>.

12. Эволюционные вычисления [Электронный ресурс]. – 2017. – Режим доступа до ресурсу: <https://habr.com/ru/post/340772/>.

13. Назначение и задачи SCADA-систем [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://www.tadviser.ru/a/53433>.

14. Четвертая промышленная революция [Электронный ресурс]. – 2017. – Режим доступа до ресурсу: <https://www.tadviser.ru/a/371579>.

15. 6 этапов на пути к Индустрии 4.0 [Электронный ресурс]. – 2017. – Режим доступа до ресурсу: <https://www.tadviser.ru/a/376310>.

16. Основы SCADA систем и их функции [Электронный ресурс]. – 2013. – Режим доступа до ресурсу: <http://ruaut.ru/content/publikacii/scada/osnovy-scada-sistem-i-ikh-funktsii.html>.

17. Яковлев К. Перспективы интеллектуальных агентов [Электронный ресурс] / Константин Яковлев. – 2018. – Режим доступа до ресурсу: <https://postnauka.ru/faq/83426>.

18. Хьюз К. Реализация агентно-ориентированных архитектур [Электронный ресурс] / К. Хьюз, Т. Хьюз. – 2014. – Режим доступа до ресурсу: <https://litresp.com/chitat/ru/%D0%A5/hjyuz-kameron/paralleljnoe-i-rasprelennoe-programmirovanie-na-s/14>.

19. Городецкий В. И. Многоагентные системы (обзор) [Электронный ресурс] / В. И. Городецкий, М. С. Грушинский, А. В. Хабалов. – 2015. – Режим доступа до ресурсу: <http://spkurdyumov.ru/networks/mnogoagentnye-sistemy-obzor/3/>.