https://doi.org/10.31891/2219-9365-2025-82-14
УДК 004.738.5:004.021:681.5:655

ПАНОВИК Уляна
Національний університет «Львівська політехніка»
e-mail: uliana.p.panovyk@lpnu.ua
https://orcid.org/0000-0002-9663-4328
КУТАС Сергій
Національний університет «Львівська політехніка»
e-mail: serhii.a.kutas@lpnu.ua

АГЕНТНЕ МОДЕЛЮВАННЯ ПОВЕДІНКИ РОЗПОДІЛЕНОЇ IOT-СИСТЕМИ ДЛЯ ПОЛІГРАФІЧНОГО ВИРОБНИЦТВА

У статті представлено підхід до моделювання поведінки розподіленої IoT-системи на основі агентної архітектури, адаптованої до умов поліграфічного виробництва. Запропоновано концептуальну модель, у якій кожен функціональний модуль (агент) виконує роль автономної одиниці з можливістю локального прийняття рішень, обміну інформацією з іншими агентами та адаптації до змін середовища. Розроблено дворівневу структуру керування: високий рівень відповідає за логіку, координацію та онтологічну інтерпретацію подій, тоді як низький рівень реалізує взаємодію з фізичними сенсорами та виконавчими пристроями. Показано сценарії динамічної реконфігурації мережі в разі збою або додавання нових вузлів. Для опису логіки роботи використано псевдокод і функціональні схеми, які дають можливість реалізувати віртуальне моделювання без потреби в спеціалізованому програмному забезпеченні. Отримані результати свідчать про ефективність агентного підходу для побудови гнучких, адаптивних і масштабованих IIoT-систем у поліграфічній галузі.

Ключові слова: IoT, агентна система, поліграфія, розподілена архітектура, моделювання, самоадаптація, псевдокод, сенсори, автоматизація.

PANOVYK Ulyana, KUTAS Serhii
Lviv Polytechnic National University

AGENT-BASED MODELING OF THE BEHAVIOR OF A DISTRIBUTED IOT SYSTEM FOR PRINTING PRODUCTION

This paper presents an agent-based modeling approach to the behavior of distributed Internet of Things (IoT) systems designed for use in industrial printing environments. The study focuses on the development of a decentralized architecture, in which each functional component of the system operates as an autonomous agent capable of local decision-making, real-time sensor-actuator interaction, and peer-to-peer communication. A two-level control structure is proposed: the low-level control manages direct communication with sensors and actuators in real-time, while the high-level control performs contextual decision-making using ontology-driven logic and coordinates interactions between agents.

The model supports system self-adaptation and fault tolerance through dynamic reconfiguration mechanisms. In the event of node failures or changes in the network topology (e.g., adding a new functional unit), agents reorganize their connections and roles without centralized intervention. The paper includes behavioral pseudocode, architectural diagrams, and communication topology schemes that formalize the logic of interaction between components and enable virtual simulation of the system without the use of specialized software platforms.

The proposed agent-based approach demonstrates high flexibility, scalability, and robustness, making it well-suited for implementation in the printing industry, where operational conditions often require dynamic adaptation and intelligent control of technological processes. The research lays the groundwork for further development of digital twins, predictive maintenance strategies, and autonomous control systems in industrial IoT infrastructures.

Keywords: agent-based modeling, distributed IoT system, printing production, adaptive architecture, decentralized control, ontology, system simulation, autonomous agents.

Стаття надійшла до редакції / Received 16.04.2025
Прийнята до друку / Accepted 11.05.2025

ПОСТАНОВКА ПРОБЛЕМИ У ЗАГАЛЬНОМУ ВИГЛЯДІ ТА ЇЇ ЗВ'ЯЗОК ІЗ ВАЖЛИВИМИ НАУКОВИМИ ЧИ ПРАКТИЧНИМИ ЗАВДАННЯМИ

Сучасні поліграфічні підприємства активно впроваджують технології Інтернету речей (IIoT) для підвищення ефективності моніторингу та управління виробничими процесами. Високий рівень автоматизації та необхідність забезпечення стабільності технологічних параметрів зумовлюють потребу в створенні інтелектуальних, адаптивних систем, здатних до оперативного реагування на зміни в умовах експлуатації [1, 2]. Одним із ключових викликів у цьому контексті є обмеженість традиційних централізованих підходів, які характеризуються низькою масштабованістю, складністю інтеграції нових пристроїв, значним навантаженням на центральні обчислювальні вузли та високою вразливістю до відмов окремих компонентів. Відмова центрального контролера або перевантаження його обчислювальних ресурсів може призводити до порушення функціонування всієї системи, що є критичним для безперервного виробничого процесу в

поліграфії. Крім того, централізовані архітектури часто не забезпечують достатньої гнучкості у випадках зміни технологічного процесу або необхідності швидкої адаптації до нових вимог.

Зважаючи на ці обмеження, актуальним напрямом досліджень є розроблення нових підходів до побудови IIoT-систем, що поєднують розподілене управління, адаптивність та можливість динамічного масштабування без необхідності значних модифікацій програмного та апаратного забезпечення. Одним із перспективних рішень є застосування агентного підходу, який передбачає використання автономних програмно-апаратних модулів, здатних до локального ухвалення рішень, самоналаштування та інтеграції у виробничий процес без централізованої координації. Впровадження агентно-орієнтованої моделі дозволяє реалізувати систему, у якій кожен пристрій або група пристроїв виступають як автономні агенти, що взаємодіють один з одним, обмінюються даними, адаптуються до змін у середовищі та забезпечують безперервність функціонування, навіть у разі відмови окремих елементів. Такий підхід створює умови для більшої стійкості системи до збоїв, зменшує залежність від єдиного контролера, сприяє оптимізації обчислювальних ресурсів та дає змогу інтегрувати нові пристрої без значного перепроєктування наявної інфраструктури.

АНАЛІЗ ДОСЛІДЖЕНЬ ТА ПУБЛІКАЦІЙ

Сучасні дослідження в області агентного моделювання охоплюють широкий спектр застосувань — від оптимізації технологічних процесів до машинного навчання, інженерного дизайну та побудови інтелектуальних багаторівневих систем. Вивчення наукових джерел дає змогу виокремити основні напрями розвитку цієї методології та її застосування в контексті Інтернету речей (IoT) і промислових середовищ. Агентне моделювання постає як ефективний підхід до реалізації гнучких, адаптивних і автономних систем, здатних до самоорганізації та масштабування без централізованого керування.

У роботі [3] описано принципи агентно-орієнтованого моделювання та його життєвий цикл, що базується на спіральній моделі розвитку інформаційних систем. Це забезпечує гнучкість при створенні складних технічних рішень. У дослідженні [5] розглянуто поєднання агентного підходу з методами машинного навчання, що дає змогу агентам адаптувати поведінку до змін середовища й підвищити ефективність IoT-систем. Книга [7] надає теоретичний огляд моделей з автономними агентами, які взаємодіють за заданими правилами, і демонструє переваги цього підходу над класичними методами симуляції. У роботі [4] показано, як багатоагентні системи можуть моделювати виробничі процеси з адаптивними параметрами та складною динамікою. Огляд [6] висвітлює застосування генеративних моделей у проєктуванні технічних систем та їх інтеграцію в агентні структури. Стаття [9] демонструє використання AgentScript для створення багаторівневих моделей без спеціалізованого ПЗ. У роботі [8] розглядається оптимізація складальних процесів засобами агентного моделювання, що сприяє підвищенню ефективності управління.

Проведений аналіз наукових публікацій демонструє, що агентне моделювання є перспективним напрямом, що дає можливість створювати автономні, адаптивні системи, здатні до ефективної взаємодії та самоналаштування в складних технічних середовищах.

ФОРМУЛЮВАННЯ ЦІЛЕЙ СТАТТІ

Метою роботи є розроблення моделі поведінки розподіленої IoT-системи на основі агентної архітектури, яка забезпечує ефективний моніторинг і керування технологічними процесами в поліграфічному виробництві. З огляду на складність сучасних виробничих систем та необхідність їхньої адаптації до змінних умов експлуатації, застосування багатоагентної організації дає змогу створити самоадаптивну систему, що об'єднує автономні інтелектуальні вузли, здатні до локального прийняття рішень і взаємодії між собою.

ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ

У сфері автоматизованого управління поліграфічними процесами побудова ефективної IoT-системи вимагає дотримання принципів масштабованості, адаптивності та надійності. Централізовані архітектури з логікою на сервері й пасивними периферійними пристроями демонструють обмеження в умовах динамічного середовища, зокрема, у зниженій відмовостійкості та складності масштабування. Альтернативою є агентно-орієнтований підхід, що передбачає розподіл інтелектуальних функцій між автономними агентами. Кожен вузол системи функціонує як самостійний агент, здатний до локального аналізу, прийняття рішень і взаємодії з іншими компонентами. Така децентралізація забезпечує гнучке управління без залежності від єдиного центру обробки (табл. 1).

У виробничих середовищах, зокрема, в поліграфії, агентна архітектура забезпечує гнучку адаптацію до змін — зміни режимів друку, заміни обладнання чи відмов окремих вузлів. Завдяки локальній автономії агенти можуть динамічно перебудовувати взаємозв’язки, змінювати параметри керування і підтримувати безперервність процесу. Такий рівень самоналаштування є суттєвою перевагою над класичними централізованими IoT-рішеннями. В умовах експлуатації різнорідного обладнання (друкарські машини,

сушильні модулі, системи контролю мікроклімату) агентна система спрощує автоматичну інтеграцію нових пристроїв без потреби в складному конфігуруванні.

Таблиця 1

Порівняльна характеристика централізованих та агентно-орієнтованих IoT-систем

Ознака	Централізована IoT-система	Агентно-орієнтована IoT-система
Архітектура	Ієрархічна, із центральним вузлом	Розподілена, з автономними агентами
Масштабованість	Обмежена через перевантаження центру	Висока завдяки децентралізованому керуванню
Надійність	Низька: відмова центрального вузла критична	Висока: система самостійно адаптується
Адаптивність до змін	Потребує ручної реконфігурації	Автоматичне самоналаштування
Взаємодія компонентів	Через центр	Пряма між агентами
Інтеграція нових пристроїв	Складна, потребує втручання	Автоматична, через динамічне включення агентів

Отже, агентно-орієнтована модель формує нову парадигму управління, де функціонування IoT-системи ґрунтується не на центральному контролері, а на взаємодії автономних вузлів. Це дає можливість ефективно вирішувати типові для поліграфії завдання: адаптивне керування мікрокліматом, стабілізацію параметрів друку та підтримку виробничого процесу, навіть за умов часткових збоїв.

Агентна архітектура в розподіленій IoT-системі для поліграфічного виробництва ґрунтується на децентралізованій структурі, де кожен функціональний вузол виступає як автономний інтелектуальний агент із можливістю прийняття рішень, взаємодії та адаптації до змін. Такий підхід забезпечує гнучкість, масштабованість і відмовостійкість у динамічних виробничих умовах. До основних компонентів належать: сенсори, що зчитують параметри середовища в реальному часі; контролери-агенти, які обробляють дані, керують виконавчими пристроями та координуються між собою; а також координаційний вузол, який синхронізує взаємодію. На відміну від централізованих SCADA-систем, управлінська логіка розподілена між edge-вузлами (STM32, ESP32, Raspberry Pi тощо), що дає можливість легко інтегрувати нові пристрої без зупинки процесу чи зміни програмної архітектури.

У межах запропонованої архітектури агент функціонує як автономний програмно-апаратний модуль із дворівневою структурою: низький рівень (Low-Level Control) забезпечує взаємодію з фізичним середовищем, високий (High-Level Control) – реалізує логіку прийняття рішень (рис. 1). Такий підхід дає змогу поєднувати обробку даних у реальному часі з когнітивною поведінкою на рівні вузла, підвищуючи надійність системи без постійної залежності від централізованих служб.

Низький рівень відповідає за зчитування даних із сенсорів (температура, вологість, тиск тощо), фільтрацію, виявлення аномалій та генерацію подій. Він також керує виконавчими елементами (вентиляторами, клапанами, приводами) через інтерфейси PWM, аналогові або цифрові виходи. Цей рівень реалізується на мікроконтролерах з обмеженими ресурсами (STM32, ESP32, ATmega) під RTOS або в екосередовищному режимі. Високий рівень обробляє контекстну інформацію та координує дії на основі онтологій, логічних правил (if–then, дерева рішень, нечітка логіка) та інформації від інших агентів. Він також відповідає за передачу команд низькому рівню, логування та аналітичну обробку. Комунікація між агентами забезпечується через MQTT, CoAP або OPC UA. Програмне ядро високого рівня функціонує на edge-пристроях із розширеними можливостями (Raspberry Pi, STM32H7, Linux-платформи) з використанням Python, C++ або JavaScript, що забезпечує масштабованість і адаптивність у межах розподіленої IoT-інфраструктури поліграфії.

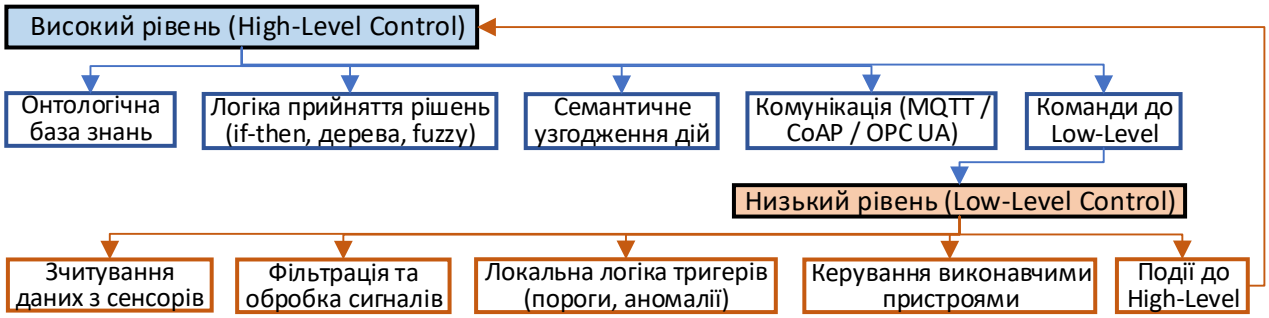

Рис. 1. Структура логіки роботи агентного модуля в IoT-системі

На основі побудованої блок-схеми структури агентного модуля нижче подано узагальнений псевдокод, який описує послідовність дій і взаємодію функціональних компонентів у процесі його роботи.

Pseudocode

```
initialize_agent():
    load_ontology()
    establish_communication_channels()
    register_with_coordinator()

main_loop():
    while system_active:
        # Low-level processing
        sensor_data = read_sensors()
        filtered_data = preprocess(sensor_data)
        if detect_event(filtered_data):
            generate_local_event(filtered_data)
            send_event_to_high_level(filtered_data)

        # High-level logic
        if event_received():
            context = analyze_context(received_event)
            decision = make_decision(context)
            send_command_to_actuators(decision)
            notify_other_agents(decision)

        # Periodic maintenance
        check_neighbor_status()
        log_activity()
        sleep(cycle_interval)
```

Псевдокод агентного модуля описує типову поведінку вузла в розподіленій IoT-системі з дворівневою архітектурою. Його структура охоплює етап ініціалізації та основний цикл роботи. На початковому етапі агент завантажує онтологію, встановлює комунікаційні канали з іншими вузлами та, за потреби, реєструється в координаційному модулі. Це забезпечує готовність до автономної роботи в мережі.

У головному циклі низький рівень зчитує дані із сенсорів, виконує фільтрацію, нормалізацію і визначає значущі події за порогами або аномаліями. У разі виявлення події інформація передається на високий рівень, де виконується контекстний аналіз із використанням онтологічних знань. На основі правил (if–then, дерева рішень, нечітка логіка) формується керуюча дія, яка надсилається до виконавчих пристроїв, а також за потреби — до інших агентів. Додатково в циклі реалізується періодичне обслуговування: перевірка зв'язку із сусідніми вузлами, логування подій та короткі затримки для стабілізації частоти виконання. Така структура забезпечує адаптивність, автономність і узгоджену поведінку системи без централізованого контролю. Псевдокод є універсальним і може застосовуватися для реалізації як сенсорних, так і керувальних вузлів у промислових IIoT-мережах.

Однією з основних переваг агентно-орієнтованих IoT-систем є розподілений обмін даними, що забезпечує гнучкість, масштабованість і стійкість до збоїв. У контексті поліграфічного виробництва агенти взаємодіють через логічні канали, обмінюючись станами, подіями та командами, що дає змогу формувати узгоджену реакцію без централізованого керування. У структурі системи виокремлюють кілька типів агентів: сенсорні – здійснюють первинне вимірювання параметрів; контролери – аналізують стан середовища й керують виконавцями; шлюзи – агрегують дані та виконують конвертацію протоколів; координатори – маршрутизують повідомлення та синхронізують дії; інтерфейсні агенти – забезпечують взаємодію з оператором; аналітичні модулі – виконують прогнозування та оцінку трендів. Комунікація відбувається на трьох рівнях: фізичному (інтерфейси RS-485, I2C, SPI, UART), логічному (програмні зв'язки між агентами) та інформаційному (обмін подіями, командами, даними у форматах JSON або XML через MQTT, CoAP або OPC UA).

Кожен агент формує повідомлення зі структурованим вмістом, що містить ідентифікатор джерела, мітку часу, тип події, значення та (опційно) контекст. Нижче наведено приклад повідомлення, яке публікується через MQTT:

```
Topic: "print/zone2/temp"
Message:
{
  "agent_id": "drying_zone_2",
  "timestamp": "2025-03-20T12:33:44Z",
```

```
  "event": "temp_warning",
  "value": 53.2
}
```

На основі типу події інші агенти ухвалюють рішення щодо зміни власної поведінки, наприклад: зменшення швидкості подачі паперу, активація додаткової вентиляції чи перехід до резервного сценарію.

Взаємодія між компонентами організована згідно з наступною розширеною схемою логічної топології, яка описує не лише напрями обміну, а й ролі вузлів у системі (рис. 2).

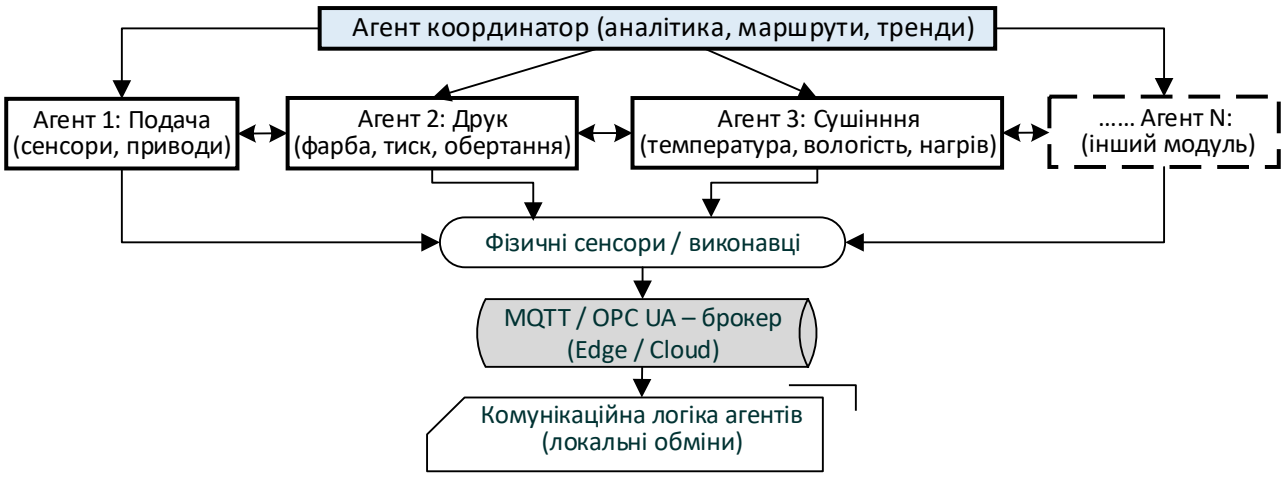

Рис. 2. Топологія взаємодії агентів у розподіленій IoT-системі

Така модель дає можливість реалізовувати децентралізовану маршрутизацію, в якій агенти обмінюються повідомленнями через спільний брокер (MQTT), але при цьому можуть встановлювати прямі peer-to-peer зв'язки через WebSocket, Modbus TCP або інші протоколи.

Обмін даними в системі класифікується на такі типи:

Комунікація	Протокол	Частота / обсяг	Тип даних
Агент ↔ Агент	MQTT (QoS 1)	1–10 сек	JSON-повідомлення, події
Агент ↔ Шлюз	Modbus/TCP	100 мс	Поточні значення, стани
Агент ↔ Координатор	MQTT / CoAP	подієва	Рішення, сценарії
Агент ↔ Інтерфейс	HTTP/WebSocket	на запит	Дані візуалізації

Запропонована структура забезпечує динамічну самоконфігурацію мережі, у якій нові пристрої інтегруються автоматично, а агенти адаптують поведінку на основі локального контексту. Відсутність жорсткої ієрархії сприяє гнучкій реконфігурації, масштабуванню і реалізації віртуального моделювання без зміни логіки системи. Особливістю архітектури є вбудована онтологічна модель, що функціонує локально на рівні кожного агента без залучення централізованих хмарних сервісів. Це дозволяє оперативно трактувати події, враховуючи специфіку поліграфічного процесу, зокрема, зміни в типі паперу, вологості чи навантаженні на друкарське обладнання.

Агентно-орієнтована IoT-система для поліграфічного виробництва має підтримувати динамічну реконфігурацію в разі змін у мережі, відмов вузлів або підключення нових компонентів. Самоорганізація та адаптивність забезпечують стійкість і гнучкість функціонування без потреби ручного втручання. У разі втрати зв'язку з вузлом (наприклад, за відсутності heartbeat-повідомлень) агент фіксує збій та ініціює локальну реконфігурацію: виконує діагностику, оцінює критичність, активує резервний режим, сповіщає координатора та синхронізує оновлену конфігурацію з іншими агентами.

Pseudocode алгоритму локальної реконфігурації:

```
def monitor_neighbors():
    for agent in neighbor_list:
        if not heartbeat_received(agent):
            handle_agent_failure(agent)

def handle_agent_failure(agent):
    log_event("Agent down", agent)
    if agent.type == "sensor":
        activate_backup_sensor(agent.zone)
```

```
        elif agent.type == "controller":
            redistribute_tasks(agent.zone)
        notify("coordinator", f"{agent} failure")
```

У прикладному сценарії, коли виходить з ладу агент температурного контролю в зоні сушіння, його сусідні агенти автоматично виявляють відсутність активності та перебудовують логіку взаємодії. Агент подачі повітря активує резервний температурний сенсор, а координатор формує рекомендації зниження швидкості друку. Після відновлення – новий агент автоматично ініціалізується, передає службове повідомлення типу "hello" і реєструється в системі. Процедура гарячої інтеграції нового вузла реалізується без зупинки виробничого процесу. Новий агент після запуску виконує базову ініціалізацію, повідомляє систему про свій статус і функції, отримує контекст від координатора, проходить реєстрацію в MQTT/OPC UA-брокері та включається в локальну взаємодію.

Pseudocode ініціалізації нового агента:

```
def initialize_new_agent():
    publish("network", {
        "type": "hello",
        "agent_id": self.id,
        "zone": self.zone,
        "capabilities": self.functions
    })
    context = await receive_context_from("coordinator")
    register_self(context)
```

Ці процеси дозволяють системі підтримувати стабільну роботу, навіть при непередбачених змінах конфігурації. Комунікація між агентами здійснюється асинхронно, з мінімальною затримкою, а функціональні обов'язки можуть бути оперативно перерозподілені на основі оновленої топології взаємодії (рис. 3).

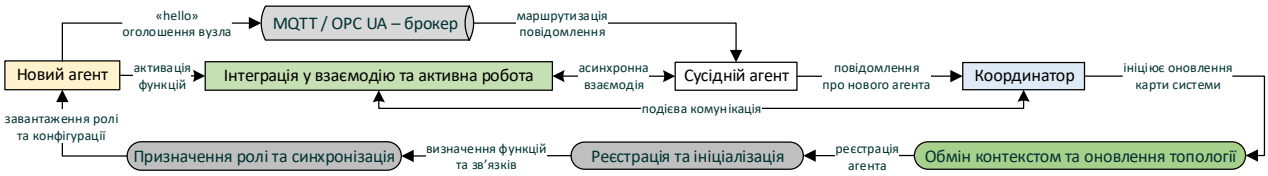

Рис. 3. Схема сценарію самоорганізації агентів при зміні конфігурації

Перевагами реалізованого підходу є забезпечення безперервності виробничих процесів за рахунок можливості підключення нових вузлів і реконфігурації системи без її зупинки. Система зберігає живучість: відмова окремого вузла не призводить до критичного збою всього комплексу. Архітектура вирізняється гнучкістю — логіка агентів оновлюється динамічно відповідно до змін середовища та контексту. Крім того, запропонована модель придатна до віртуального моделювання: усі процеси адаптації можуть бути описані на рівні псевдокоду, логічних діаграм та сценаріїв без використання спеціалізованого програмного забезпечення.

Функціонування агентної IoT-системи доцільно формалізувати як динамічну мережу взаємодіючих вузлів, чия поведінка описується моделями комунікації, прийняття рішень і відмовостійкості. Це дозволяє кількісно оцінити ефективність обміну даними, реакцію на події, стійкість до збоїв і здатність до масштабування без втрати продуктивності.

Агентна система математично представлена у вигляді орієнтованого графа $G=(V, E)$, де множина $V=\{a_1, a_2, \ldots, a_n\}$ містить агентів, а множина $E \subseteq V \times V$ визначає логічні зв'язки між ними. Кожен агент a_i описується трійкою $\langle S_i, D_i, A_i \rangle$, де S_i – набір сенсорних входів, D_i – вихідні дії (керуючі сигнали), а A_i – функція ухвалення рішень. Динаміка поведінки агента моделюється функцією переходу:

$$y_i(t) = A_i\big(x_i(t), X_{N(i)}(t), K_i\big), \tag{1}$$

де $x_i(t)$ – локальні сенсорні значення, $X_{N(i)}(t)$ – інформація від сусідніх агентів, K_i – контекст у вигляді онтологічних знань, а $y_i(t)$ – вихідний керуючий вплив.

Для оцінювання ефективності роботи системи розраховується максимальний час реакції, який визначається як сума часу передачі повідомлення між агентами та часу обробки події:

$$T_{max} = \underbrace{max}_{(a_i,a_j)\in E} \{T_{ij}^{comm} + T_j^{proc}\}. \tag{2}$$

Продуктивність агентної взаємодії може бути виражена як середня кількість ефективно оброблених подій за одиницю часу:

$$P_{sys} = \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{n}\delta_i(t), \quad (3)$$

де $\delta_i(t)$=1, якщо агент успішно обробив подію в момент часу t, або 0 у протилежному випадку.

Надійність системи визначається ймовірністю її безвідмовної роботи. Якщо припустити, що кожен агент має ймовірність відмови p_i, тоді загальна ймовірність збереження функціональності системи в часі t набуває вигляду:

$$R_{sys}(t) = \prod_{i=1}^{n}(1-p_i)^t. \quad (4)$$

У разі наявності механізмів резервування функцій між агентами, наприклад, горизонтального дублювання, ймовірність зростає:

$$R_{sys}^{(res)}(t) = \prod_{i=1}^{n}[1-p_i^2]^t, \quad (5)$$

що демонструє підвищену стійкість багатоагентної системи до часткових збоїв.

Окремо варто оцінити навантаження на комунікаційну інфраструктуру. Середня кількість повідомлень між агентами за період T може бути визначена як:

$$L(t) = \frac{1}{T}\sum_{t=1}^{T}\sum_{(a_i,a_j)\in E} m_{ij}(t), \quad (6)$$

де $m_{ij}(t)$ – кількість повідомлень від агента a_i до a_j у момент часу t. Масштабованість системи вважається прийнятною, якщо похідна навантаження за кількістю агентів не перевищує порогове значення: $dL/dn \leq \epsilon$.

Отже, запропонована математична модель дає можливість комплексно оцінити ключові властивості агентної IoT-системи – від затримок у реагуванні й загальної продуктивності до надійності та стабільності комунікацій. Ці показники є критично важливими під час створення прототипу майбутньої системи автоматизації поліграфічного виробництва. Їх урахування дає змогу не лише перевірити поведінку системи на етапі моделювання, а й обґрунтувати архітектурні рішення, адаптивні сценарії і рівень живучості всієї інфраструктури в умовах змін конфігурації або часткових відмов.

ВИСНОВКИ З ДАНОГО ДОСЛІДЖЕННЯ І ПЕРСПЕКТИВИ ПОДАЛЬШИХ РОЗВІДОК У ДАНОМУ НАПРЯМІ

У процесі дослідження розроблено концептуальну модель розподіленої IoT-системи для поліграфічного виробництва на основі агентного підходу. Запропонована архітектура дозволяє створити систему з автономних модулів, які здатні самостійно аналізувати сенсорні дані, приймати локальні рішення, координуватися з іншими агентами та адаптуватися до змін у середовищі. Модель демонструє ефективність у подоланні обмежень централізованих IIoT-систем – зокрема, в аспектах масштабованості, відмовостійкості та інтеграції нових пристроїв. Дворівнева структура агентів (сенсорна й логічна) забезпечує оперативне реагування та гнучку поведінку системи загалом.

Подальші дослідження передбачають розширення агентної моделі шляхом інтеграції інтелектуальних механізмів прогнозування, зокрема, машинного навчання та нечіткої логіки. Перспективним напрямом є створення симуляційного середовища на основі цифрового двійника поліграфічного підприємства для тестування складних сценаріїв. Також передбачається впровадження прототипу системи в реальних умовах із поетапним розгортанням на ключових ділянках — сушіння, подача, контроль якості. Отримані результати підтверджують ефективність агентного підходу для побудови інтелектуальних, адаптивних і надійних IIoT-рішень, що сприяють цифровій трансформації поліграфічної галузі.

Література

1. Пановик У. П., Кутас С.А. Інтернет речей для інтелектуального поліграфічного виробництва. *Поліграфія і видавнича справа*, № 1(87),2024, С. 61–74. URL: https://doi.org/10.32403/0554-4866-2024-1-87-61-74

2. Пановик У.П., Кутас С.А. Трансформація поліграфічної індустрії за допомогою цифровізації та інтернету речей. *Автоматизація та комп'ютерно-інтегровані технології у виробництві та освіті: стан, досягнення, перспективи розвитку:* тези доповідей Всеукраїнської науково-практичної Internet-конференції, Черкаси, 2024, с. 119-121. URL: https://conference.ikto.net/pub/akit_2024_11-17march_1.pdf

3. Примостка А. О. Концептуальні засади агентно-орієнтованого моделювання економічних процесів. *Формування ринкової економіки*. Київ: КНЕУ, 2014. Вип. 32. с. 402–412. URI: https://ir.kneu.edu.ua:443/handle/2010/12297

4. Bobkov S. P., Astrakhantseva I. A. The use of multi-agent systems for modeling technological processes. *Journal of Physics: Conference Series*. 2021. Vol. 2001, no. 1. P. 012002. URL: https://doi.org/10.1088/1742-6596/2001/1/012002

5. Nascimento N., Alencar P., Lucena C., & Cowan D. (2018). An IoT analytics embodied agent model based on context-aware machine learning. In *2018 IEEE International Conference on Big Data (Big Data)*. pp. 5170-5175. URL: https://doi.org/10.48550/arXiv.1812.06791

6. Regenwetter L., Nobari A. H., Ahmed F. Deep Generative Models in Engineering Design: A Review. *Journal of Mechanical Design*. 2022. Vol. 144, no. 7. URL: https://doi.org/10.1115/1.4053859

7. Sayama H. Introduction to the Modeling and Analysis of Complex Systems. Open SUNY Textbooks, 2015. 496 p.

8. Ushakova I. Application of computer agent modeling for optimization of the assembly process. *Системи обробки інформації*, 2020. No. 1(160). P. 18–25. URL: https://doi.org/10.30748/soi.2020.160.02

9. Yezhova Y., Maslova N. Using agentscript to produce multi-level agent-based modelling models. *Scientific Papers of Donetsk National Technical University. Series: «Computer Engineering and Automation»*, 2024. P. 54–66. URL: https://doi.org/10.31474/2786-9024/v2i3(35).319553

References

1. Panovyk, U. P., Kutas, S. A. (2024). Internet of Things for Intelligent Printing Production. *Polihrafiia i vydavnycha sprava*, 1(87), 61–74. URL: https://doi.org/10.32403/0554-4866-2024-1-87-61-74

2. Panovyk, U. P., Kutas, S. A. (2024). Transformation of the Printing Industry through Digitalization and the Internet of Things. In: *Automation and Computer-Integrated Technologies in Production and Education: State, Achievements, Prospects of Development. Proceedings of the All-Ukrainian Scientific and Practical Internet Conference*, Cherkasy, pp. 119–121. URL: https://conference.ikto.net/pub/akit_2024_11-17march_1.pdf

3. Prymostka, A. O. (2014). Conceptual Principles of Agent-Oriented Modeling of Economic Processes. *Formuvannia rynkovoi ekonomiky*, Issue 32, Kyiv: KNEU, pp. 402–412. URI: https://ir.kneu.edu.ua:443/handle/2010/12297

4. Bobkov S. P., Astrakhantseva I. A. The use of multi-agent systems for modeling technological processes. *Journal of Physics: Conference Series*. 2021. Vol. 2001, no. 1. P. 012002. URL: https://doi.org/10.1088/1742-6596/2001/1/012002

5. Nascimento N., Alencar P., Lucena C., & Cowan D. (2018). An IoT analytics embodied agent model based on context-aware machine learning. In *2018 IEEE International Conference on Big Data (Big Data)*. pp. 5170-5175. URL: https://doi.org/10.48550/arXiv.1812.06791

6. Regenwetter L., Nobari A. H., Ahmed F. Deep Generative Models in Engineering Design: A Review. *Journal of Mechanical Design*. 2022. Vol. 144, no. 7. URL: https://doi.org/10.1115/1.4053859

7. Sayama H. Introduction to the Modeling and Analysis of Complex Systems. Open SUNY Textbooks, 2015. 496 p.

8. Ushakova I. Application of computer agent modeling for optimization of the assembly process. *Системи обробки інформації*, 2020. No. 1(160). P. 18–25. URL: https://doi.org/10.30748/soi.2020.160.02

9. Yezhova Y., Maslova N. Using agentscript to produce multi-level agent-based modelling models. *Scientific Papers of Donetsk National Technical University. Series: «Computer Engineering and Automation»*, 2024. P. 54–66. URL: https://doi.org/10.31474/2786-9024/v2i3(35).319553