

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту  
(повна назва)

Кафедра Інформатики  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

**ДОСЛІДЖЕННЯ ТА РЕАЛІЗАЦІЯ ПІДХОДІВ ДО ПОБУДОВИ  
БІЗНЕС- ПРОЦЕСІВ У НОТАЦІЇ VRMN ІЗ ЗАСТОСУВАННЯМ  
МОДЕЛЕЙ ПРИРОДНОЇ МОВИ ТА SAMUNDA**

(тема)

Виконав:

здобувач 2 року навчання,

групи ІНФМ-24-1

Іткін Д. О.

(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки

(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика

(повна назва освітньої програми)

Науковий керівник доц. Любченко В. А.

(посада, прізвище, ініціали)

Допускається до захисту

Завідувач кафедри інформатики \_\_\_\_\_

(підпис)

Кобилін О. А.

(прізвище, ініціали)

2025 р.

## Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджментуКафедра ІнформатикиРівень вищої освіти другий (магістерський)Спеціальність 122 Комп'ютерні науки  
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

«\_\_\_\_\_» \_\_\_\_\_ 2025 р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУздобувачеві Іткіну Денису Олександровичу  
(прізвище, ім'я, по батькові)1. Тема роботи Дослідження та реалізація підходів до побудови бізнес- процесів у нотації BPMN із застосуванням моделей природної мови та Camunda

---

затверджена наказом університету від 14 листопада 2025 року № 1045Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 22 листопада 2025 р.

3. Вихідні дані до роботи текстові описи бізнес-процесів різного рівня складності, літературні джерела щодо застосування методів моделювання бізнес-процесів та обробки природної мови, програмні засоби для реалізації системи автоматичної генерації BPMN-діаграм, база XML-шаблонів елементів BPMN, що використовується для побудови структур процесів, Golang, React.JS, СУБД PostgreSQL.  

---

---

---

---

---

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

1. Аналіз сучасних методів моделювання бізнес-процесів та їх автоматизації.

2. Аналіз літературних джерел щодо застосування обробки природної мови для генерації структурованих моделей процесів.

3. Формування покрокового алгоритму автоматичної генерації BPMN-діаграм на основі текстового опису бізнес-процесу.

4. Візуалізація сформованих етапів побудови BPMN-моделі та логіки обробки текстових даних.

5. Розробка програмної системи, що забезпечує генерацію BPMN-діаграм із текстового опису за допомогою мовної моделі та MCP-сервер.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність проблеми автоматизації моделювання бізнес-процесів, діаграма об'єкта та мети дослідження, постановка задачі перетворення тексту на BPMN, блок-схеми етапів обробки текстового опису, схема семантичної інтерпретації тексту, архітектурна схема системи, приклади XML-шаблонів BPMN-елементів із MCP-сервер, приклади згенерованих BPMN-діаграм для різних типів процесів, ілюстрація інтерфейсу системи з модулем візуалізації BPMN, схеми тестових сценаріїв, результати тестування.

6. Консультанти розділів роботи (п.6 включається до завдання 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

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	29.09.2025	
2	Аналіз завдання, підбір літератури	30.09.25-07.10.25	
3	Аналіз літератури з досліджуваної проблеми	08.10.25-14.10.25	
4	Особливості методів обробки природної мови та підходів до автоматичної генерації BPMN-діаграм	15.10.25-20.10.25	
5	Дослідження методів аналізу текстових описів бізнес-процесів	21.10.25-27.10.25	
6	Програмна реалізація	28.10.25-05.11.25	
7	Обґрунтування отриманих результатів	06.11.25-11.11.25	
8	Оформлення пояснювальної записки	12.11.25-14.11.25	
9	Перевірка на нормоконтроль	21.11.25	
10	Перевірка на плагіат	22.11.25	
11	Рецензування	23.11.25	
12	Підготовка презентації та доповіді	24.11.25	
13	Занесення роботи в електронний архів	03.12.25	
14	Попередній захист кваліфікаційної роботи	03.12.25	

Дата видачі завдання 29 вересня 2025 р.

Здобувач \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_  
(підпис)

доц. Любченко В. А.  
(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи: 70 с., 4 табл., 15 рис., 45 джерел.

АВТОМАТИЗАЦІЯ, БІЗНЕС-ПРОЦЕСИ, ГЕНЕРАЦІЯ ДІАГРАМ, ІНТЕЛЕКТУАЛЬНІ АГЕНТИ, ОБРОБКА ПРИРОДНОЇ МОВИ, XML-ШАБЛони, BPMN, BPMS, CAMUNDA, LLM, MCP-СЕРВЕР, OPENAI, GEMINI.

Об'єктом дослідження є процес моделювання бізнес-процесів у сучасних інформаційних системах.

Метою дослідження є розроблення підходу до генерації бізнес-процесів у нотації BPMN із використанням інтелектуальних агентів і мовних моделей, що дозволяє автоматизувати етапи аналізу, структурування та побудови XML-моделі процесу для систем управління бізнес-процесами Camunda.

У роботі використано методи обробки природної мови, структурного аналізу текстів, шаблонного моделювання, а також засоби інтеграції з MCP-сервер для зберігання та отримання XML-шаблонів елементів BPMN. Система взаємодіє з мовною моделлю, яка аналізує текст користувача та формує запити для побудови BPMN-діаграми.

Наукова новизна роботи полягає у створенні архітектури, що поєднує мовної моделі у єдину систему автоматичної генерації BPMN-процесів, придатних для імпорту в Camunda.

Взаємозв'язок з іншими роботами полягає у розвитку напрямку інтелектуальної автоматизації процесів.

Рекомендації щодо використання результатів роботи – впровадження розробленого підходу у середовища розробки BPMN-моделей для скорочення часу створення процесів і підвищення їх точності. У результаті дослідження створено прототип системи, яка на основі текстового опису формує готову BPMN-діаграму, використовуючи мовну модель для аналізу запитів і MCP-сервер для отримання XML-шаблонів.

## ABSTRACT

Explanatory note to the qualification work: 70 pages, 4 table, 15 figures, 45 sources.

AUTOMATION, BPMN, BPMS, BUSINESS PROCESSES, CAMUNDA, DIAGRAM GENERATION, INTELLIGENT AGENTS, LLM, MCP SERVER, NATURAL LANGUAGE PROCESSING, OPENAI, GEMINI, SEQUENCE FLOW, XML TEMPLATES.

The object of the study is the process of business process modeling in modern information systems.

The purpose of the research is to develop an approach for generating business processes in BPMN notation using intelligent agents and large language models, which enables the automation of the analysis, structuring, and construction stages of XML-based process models for business process management systems such as Camunda.

The research employs methods of natural language processing, structural text analysis, and template-based modeling, as well as integration tools with the MCP-сервер for storing and retrieving XML templates of BPMN elements. The system interacts with a language model, which analyzes the user's text and generates requests to build the BPMN diagram.

The scientific novelty of the research lies in the development of an architecture that integrates LLM models into a unified system for automatic BPMN process generation suitable for import into Camunda.

The relationship with other studies is found in the advancement of intelligent process automation.

The recommendations for practical application include the implementation of the proposed approach in BPMN model development environments to reduce process design time and improve accuracy. As a result of the study, a prototype system was created that can automatically generate a complete BPMN diagram from a textual description using an LLM for text analysis and the MCP-сервер for retrieving XML templates.

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів .....	8
Вступ.....	10
1 Огляд основних методів моделювання бізнес-процесів у BPMN та інтелектуальних підходів до їх автоматизації теми .....	12
1.1 Еволюція підходів до моделювання бізнес-процесів .....	12
1.1.1 Методології IDEF, DFD, EPC .....	12
1.1.2 Поява BPMN як універсального стандарту.....	13
1.2 Концепція BPMN та її роль у цифровій трансформації бізнесу .....	13
1.2.1 Основні елементи та принципи BPMN .....	13
1.2.2 Переваги BPMN у порівнянні з іншими нотаціями.....	14
1.3 Методи інтелектуальної інтерпретації текстових описів процесів .	14
1.3.1 Основні елементи та принципи BPMN .....	15
1.3.2 Мовна модель у розпізнаванні логічних структур і контексту	15
1.4 Застосування інтелектуальних агентів у BPMN-моделюванні.....	16
1.4.1 Поняття інтелектуальних агентів та їх роль у моделюванні BPMN .....	16
1.4.2 Логіка роботи інтелектуального агента при побудові BPMN-моделі.....	17
1.4.3 Взаємодія між системами.....	17
1.4.4 Переваги та обмеження використання інтелектуальних агентів .....	19
1.4.5 Перспективи розвитку агентних технологій у BPMN-моделюванні .....	19
1.5 Постановка задачі дослідження.....	20
2 Моделі та методи автоматизації побудови BPMN-діаграм .....	22
2.1 Модель обробки природної мови в задачах побудови BPMN .....	22
2.1.1 Основні принципи обробки природної мови .....	22
2.1.2 Використання мовних моделей для семантичного аналізу.....	23
2.1.3 Етапи перетворення тексту у формальні структури .....	24

2.2	Формальна модель перетворення тексту в BPMN-структуру.....	24
2.2.1	Концептуальні основи формальної моделі .....	25
2.2.2	Семантичні правила класифікації елементів процесу .....	26
2.2.3	Формування потоків керування.....	27
2.2.4	Валідація BPMN-моделі.....	27
2.3	Формалізація процесу побудови BPMN-діаграм за допомогою моделей обробки природної мови .....	29
2.3.1	Етапи формалізації тексту бізнес-процесу.....	29
2.3.2	Використання моделей для контекстного розпізнавання.....	30
2.3.3	Формування потоків та залежностей між елементами.....	31
2.4	Алгоритм перетворення текстового опису бізнес-процесу у BPMN-діаграму .....	31
2.4.1	Попередня обробка тексту .....	32
2.4.2	Визначення типів елементів процесу .....	32
2.4.3	Побудова логічної структури процесу .....	33
2.4.4	Формування повної BPMN-діаграми.....	34
2.5	Архітектура системи автоматичної генерації BPMN-діаграм .....	34
2.5.1	Загальна структура архітектури.....	34
2.5.2	Логіка послідовної обробки даних .....	35
2.5.3	Роль мовної моделі у системі.....	35
2.5.4	MCP-сервер як джерело структурної істини.....	37
2.5.5	Гнучкість, масштабованість і адаптивність архітектури.....	38
3	Реалізація системи автоматичного моделювання бізнес-процесів.....	40
3.1	Загальна характеристика системи .....	40
3.2	Реалізація серверної частини .....	42
3.2.1	Структура запити генерації BPMN.....	43
3.2.2	Взаємодія з API мовної моделі .....	44
3.2.3	Взаємодія з MCP-сервер.....	45
3.2.4	Робота з базою даних .....	46
3.2.5	Узагальнення результатів реалізації серверної частини .....	47
3.3	Реалізація клієнтської частини .....	49

3.3.1	Опис інтерфейсу користувача.....	49
3.3.2	Механізм взаємодії з сервером .....	51
3.4	Тестування та обґрунтування результатів .....	52
3.4.1	Методика тестування .....	52
3.4.2	Тестові сценарії .....	53
3.4.3	Перевірка помилок .....	58
3.4.4	Порівняння мовних моделей.....	59
	Висновки.....	64
	Перелік джерел посилання .....	66

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

AI – Artificial Intelligence (штучний інтелект)

BPMN – Business Process Model and Notation (нотація моделювання бізнес-процесів)

BPMS – Business Process Management System (система управління бізнес-процесами)

Camunda – програмна платформа для моделювання, виконання та моніторингу бізнес-процесів на основі BPMN

DI – Diagram Interchange (стандарт обміну графічними елементами BPMN)

DC – Diagram Common (графічні об'єкти, що використовуються в BPMN-нотації)

DFD – Data Flow Diagrams (діаграми потоків даних, призначені для опису переміщення інформації в системі)

EPC – Event-driven Process Chain (ланцюги процесів, керовані подіями, що відображають зв'язок між подіями та функціями у процесі)

IDE – Integrated Development Environment (інтегроване середовище розробки)

IDEF – Integrated DEFinition (сімейство методів структурного моделювання, призначених для опису функцій, процесів, даних та взаємодій у складних системах; використовується для аналізу, проектування та оптимізації бізнес-процесів)

LLM – Large Language Model (велика мовна модель)

MCP-сервер – Model Context Protocol Server (сервер контекстного протоколу моделі, що зберігає XML-шаблони BPMN елементів)

NLP – Natural Language Processing (обробка природної мови)

Gemini – мовна модель штучного інтелекту, що використовується для аналізу текстових запитів користувача

REST API – Representational State Transfer Application Programming Interface (інтерфейс програмування для взаємодії між клієнтом і сервером)

UI – User Interface (інтерфейс користувача)

XML – eXtensible Markup Language (розширювана мова розмітки, що використовується для представлення BPMN-структур)

Sequence Flow – потік послідовності, що визначає порядок виконання завдань у BPMN-діаграмі

Start Event / End Event – початкова та кінцева події бізнес-процесу

Gateway – вузол розгалуження або злиття потоків у BPMN

Service Task – автоматизоване завдання, яке виконується системою або зовнішнім сервісом

User Task – завдання, що виконується користувачем

LLM Agent – інтелектуальний агент, який взаємодіє з мовною моделлю для генерації BPMN

## ВСТУП

Актуальність роботи полягає у зростаючій потребі автоматизації управління бізнес-процесами в умовах цифрової трансформації підприємств і організацій. Сучасні компанії дедалі частіше прагнуть підвищити ефективність своєї діяльності за рахунок оптимізації внутрішніх процесів, скорочення часу на прийняття рішень та зменшення кількості рутинних операцій. Одним із ключових інструментів для опису, аналізу та вдосконалення процесів є BPMN – стандартизована мова моделювання бізнес-процесів, яка забезпечує єдине уявлення про роботу системи для аналітиків, розробників і керівників проєктів.

Однак традиційний підхід до побудови BPMN-діаграм потребує залучення кваліфікованих фахівців – бізнес-аналітиків або архітекторів процесів, які володіють спеціалізованими знаннями та навичками роботи з інструментами моделювання. Це обмежує масштабність використання BPMN у малих і середніх компаніях та ускладнює швидке оновлення моделей процесів у динамічних умовах бізнесу. Саме тому особливої актуальності набуває розроблення систем, здатних автоматично перетворювати текстові описи бізнес-процесів на формалізовані BPMN-діаграми за допомогою інтелектуальних агентів і моделей обробки природної мови.

Завдяки стрімкому розвитку великих мовних моделей, таких як GPT, Gemini, стало можливим створювати системи, що розуміють природну мову користувача і можуть генерувати структуровані результати на її основі. У сфері моделювання бізнес-процесів це відкриває нові можливості – автоматичне визначення елементів процесу (подій, завдань, шлюзів), встановлення зв'язків між ними, а також формування коректного XML-представлення BPMN, яке може бути імпортоване у системи на кшталт Camunda.

У сучасних дослідженнях простежується тенденція інтеграції AI-компонентів із BPMS-платформами, що дозволяє створювати гібридні рішення для інтелектуального моделювання процесів. Одним із таких підходів є використання MCP-сервер – контекстного серверу, який зберігає шаблони

ВРМН-елементів та забезпечує зв'язок між мовною моделлю і процесом генерації діаграми. Це дозволяє мовній моделі не вигадувати XML-фрагменти самостійно, а формувати їх на основі перевірених шаблонів, що гарантує правильність синтаксису та відповідність ВРМН-стандарту.

Наукова задача, яку вирішує дана робота, полягає у створенні моделі взаємодії між мовною моделлю, МСР-сервер та ВРМС, що забезпечує автоматичну генерацію ВРМН-діаграм з текстових описів процесів. Для цього необхідно розробити архітектуру, яка б дозволила інтелектуальному агенту аналізувати вхідні дані, звертатися до бази шаблонів ВРМН-елементів, формувати XML-фрагменти та поєднувати їх у єдину структурну модель процесу.

Сучасний стан досліджень у цій галузі демонструє високу перспективність такого підходу. Компанії, що працюють у сфері автоматизації бізнес-процесів, вже інтегрують моделі у свої рішення, однак більшість із них мають експериментальний характер і не забезпечують достатньої узгодженості між аналітичними та технічними компонентами. Тому розробка системи, яка поєднує інтелектуальну обробку природної мови, шаблонне моделювання та валідацію ВРМН, має як наукову, так і практичну значущість.

Отже, дослідження спрямоване на розроблення підходу до моделювання бізнес-процесів у нотації ВРМН із використанням інтелектуальних агентів та мовних моделей, який дозволяє автоматизувати перехід від текстового опису до повноцінної ВРМН-діаграми, що може бути виконана у системах типу Camunda.

# 1 ОГЛЯД ОСНОВНИХ МЕТОДІВ МОДЕЛЮВАННЯ БІЗНЕС-ПРОЦЕСІВ У VRMN ТА ІНТЕЛЕКТУАЛЬНИХ ПІДХОДІВ ДО ЇХ АВТОМАТИЗАЦІЇ ТЕМИ

## 1.1 Еволюція підходів до моделювання бізнес-процесів

Моделювання бізнес-процесів як наукова та практична дисципліна бере початок у 1970-х роках, коли компанії почали систематизувати управлінські процедури для оптимізації діяльності. Розвиток інформаційних технологій зумовив потребу у створенні методів, які дозволяли б наочно відображати логіку функціонування організації [1].

### 1.1.1 Методології IDEF, DFD, EPC

Першим системним підходом до моделювання став метод IDEF0, розроблений для Міністерства оборони США. Він описував функції системи, її вхідні та вихідні дані, механізми управління та ресурси. IDEF0 базувався на принципі ієрархічної декомпозиції, що дозволяло деталізувати процес на кілька рівнів. Основна перевага методу – чіткість структурного опису, проте він не передбачав динамічної логіки виконання.

Паралельно розвивався метод DFD, який зосереджувався на переміщенні інформації між окремими частинами системи. Він давав змогу визначити, які дані надходять до системи, як вони опрацьовуються та куди передаються. Проте цей метод не враховував часової та логічної послідовності дій, що ускладнювало його використання для опису й аналізу бізнес-процесів.

У 1990-х роках на основі систем класу ERP сформувалася методологія EPC (ланцюги процесів, керовані подіями), розроблена компанією SAP. EPC запровадила принцип «подія – функція», за яким кожна подія запускає або завершує певну дію. Такий підхід уперше дав можливість формально відобразити умови переходів між станами процесу.

### 1.1.2 Поява BPMN як універсального стандарту

BPMN – універсальна нотація для моделювання бізнес-процесів, створена консорціумом OMG на початку 2000-х років для аналітиків, розробників і керівників; містить графічні елементи для опису подій, завдань, розгалужень, потоків та ролей у процесі).

Особливістю BPMN є можливість автоматичного виконання моделей завдяки представленню у форматі XML. Це забезпечило інтеграцію з BPMS, зокрема Camunda, BonitaSoft, IBM BPM, Appian, що дозволило перейти від статичного моделювання до динамічного управління процесами.

## 1.2 Концепція BPMN та її роль у цифровій трансформації бізнесу

BPMN стала одним із центральних інструментів цифрової трансформації підприємств. Вона не лише документує бізнес-процеси, але й дає змогу створювати системи, які виконують ці процеси автоматично.

### 1.2.1 Основні елементи та принципи BPMN

BPMN 2.0, ухвалена у 2011 році, визначає понад сто елементів, однак основними є події, що відображають початкові, проміжні або кінцеві стани процесу і подані на рисунку 1.1 а); завдання, які представляють конкретні дії, що виконуються користувачем або системою та наведені на рисунку 1.1 б); шлюзи, що слугують логічними точками розгалуження або об'єднання потоків і зображені на рисунку 1.1 в); потоки, які визначають послідовність виконання завдань; а також пули та доріжки, що відображають ролі або учасників процесу.

Головний принцип BPMN – поєднання наочності та формальності. Діаграма має бути зрозумілою для користувача, але водночас однозначною для виконання в системі.



створювати діаграму вручну. Це призвело до появи систем, які застосовують інтелектуальні методи обробки природної мови та великі мовні моделі.

### 1.3.1 Основні елементи та принципи BPMN

NLP забезпечує синтаксичний і семантичний розбір тексту. На цьому етапі з опису виділяються:

- дії (дієслова – кандидати на завдання BPMN);
- суб'єкти (учасники процесу, ролі);
- умови (для шлюзів і подій);
- часові маркери (для таймерів).

Алгоритми на основі POS-тегінгу, залежнісного парсингу та семантичного аналізу дозволяють визначити, які частини тексту відповідають бізнес-діям.

### 1.3.2 Мовна модель у розпізнаванні логічних структур і контексту

Великі мовні моделі, такі як OpenAI, Gemini, Claude, значно розширили можливості NLP. Вони здатні розуміти контекст, розпізнавати намір користувача і будувати причинно-наслідкові зв'язки між діями.

Наприклад, якщо користувач вводить опис «Клієнт заповнює форму, система перевіряє дані. Якщо все вірно, надсилається підтвердження, інакше — повідомлення про помилку», мовна модель не лише виокремлює ключові логічні кроки, а й визначає їхню роль у процесі. На основі цього тексту вона ідентифікує три основні завдання, шлюз для прийняття рішення та два можливі завершення процесу. Таким чином, мовна модель перетворює неструктурований опис у чіткий та формалізований набір BPMN-елементів, забезпечуючи однозначність інтерпретації та можливість подальшої автоматизації.

## 1.4 Застосування інтелектуальних агентів у BPMN-моделюванні

Застосування інтелектуальних агентів у процесі моделювання бізнес-процесів у нотації BPMN є одним із найактуальніших напрямів сучасної автоматизації. Такі агенти дозволяють перетворювати неструктуровані текстові описи на формалізовані моделі, забезпечуючи ефективну взаємодію між користувачем, системою управління процесами та мовною моделлю. Їх використання значно знижує вимоги до технічних знань користувача, адже замість ручного створення діаграм достатньо описати процес природною мовою. Це сприяє розвитку концепції «інтелектуального моделювання», де системи на основі штучного інтелекту виконують більшість аналітичних і технічних дій, що раніше вимагали участі бізнес-аналітика.

### 1.4.1 Поняття інтелектуальних агентів та їх роль у моделюванні BPMN

Інтелектуальний агент у загальному розумінні – це автономна програмна сутність, здатна сприймати навколишнє середовище, аналізувати отримані дані та приймати рішення відповідно до поставлених цілей. На відміну від традиційних алгоритмів, які виконують наперед визначені команди, агент діє адаптивно та може змінювати свою поведінку залежно від контексту. У BPMN-моделюванні інтелектуальні агенти виступають центральними компонентами систем, які забезпечують перехід від опису процесу в природній мові до побудови повноцінної XML-моделі.

Основними властивостями агентів є автономність, тобто здатність працювати без безпосереднього контролю користувача; реактивність, що полягає у швидкому реагуванні на зовнішні зміни; і цілеспрямованість, яка виражається у плануванні дій для досягнення кінцевої мети. У BPMN-системах інтелектуальні агенти можуть аналізувати вхідні запити, визначати типи елементів процесу, формувати логічні зв'язки між ними та створювати завершену діаграму.

#### 1.4.2 Логіка роботи інтелектуального агента при побудові BPMN-моделі

Робота агентної системи у BPMN-моделюванні розпочинається з моменту введення користувачем опису процесу у вигляді тексту. Наприклад, користувач може ввести фразу: «Клієнт створює заявку, система перевіряє дані, після чого надсилає підтвердження». На цьому етапі агент обробки природної мови виконує синтаксичний і семантичний аналіз тексту. Він визначає дієслова як потенційні завдання, іменники як учасників процесу або об'єкти взаємодії, а також встановлює послідовність дій.

Після аналізу текст передається до когнітивного агента, який на основі великої мовної моделі формує логічну структуру процесу [2]. Цей агент здатний розпізнати паралельність або умовність дій, зрозуміти причинно-наслідкові зв'язки між подіями й визначити типи BPMN-елементів, необхідних для побудови моделі. Далі система звертається до MCP-сервер – спеціалізованого сховища XML-шаблонів, де зберігаються фрагменти коду для кожного елемента BPMN. Агент отримує відповідні шаблони, заповнює їх параметрами (ідентифікаторами, назвами, типами), а потім формує єдину BPMN-структуру, доповнену потоками послідовності.

Таким чином, побудова BPMN-діаграми відбувається не шляхом прямого програмування, а завдяки комунікації між кількома агентами, що взаємодіють між собою. Один відповідає за лінгвістичний аналіз, другий – за логічну побудову, третій – за формальну валідацію отриманого результату

#### 1.4.3 Взаємодія між системами

Сучасні системи моделювання бізнес-процесів дедалі частіше базуються на інтеграції декількох технологічних складових, серед яких важливу роль відіграють інтелектуальні агенти, системи керування бізнес-процесами (BPMS) та платформи обробки даних. Взаємодія між цими компонентами є ключовим

чинником ефективності автоматизації, оскільки забезпечує цілісність процесу – від сприйняття інформації до її візуалізації та виконання.

Інтелектуальні агенти виступають проміжною ланкою між користувачем і технічною інфраструктурою. Вони здатні аналізувати неструктуровані запити, отримані у вигляді тексту, перетворювати їх на структуровані дані та передавати далі у систему моделювання. Такі агенти не працюють ізольовано: вони взаємодіють із базами знань, серверами шаблонів, системами обробки природної мови, а також із середовищами виконання процесів. Ця взаємодія зазвичай відбувається через стандартизовані інтерфейси та протоколи обміну даними, що дозволяє об'єднувати різні технологічні платформи у спільну екосистему.

У межах бізнес-моделювання важливим є зв'язок між аналітичними модулями, які відповідають за інтерпретацію процесів, і системами управління процесами, що реалізують виконання цих моделей. Аналітичні модулі, побудовані на принципах штучного інтелекту, визначають логіку процесу, а BPMS-системи відповідають за його технічне втілення. Обмін даними між ними відбувається у формі структурованих описів процесів, зазвичай у форматі XML або JSON.

У такій архітектурі особливу роль відіграє рівень узгодженості між компонентами. Якщо аналітична система створює модель процесу, вона повинна бути сумісною з форматом, який може обробити система виконання. Тому використовуються універсальні нотації, зокрема BPMN 2.0, які забезпечують сумісність між різними інструментами. Це дозволяє створювати гнучкі системи, де процеси можуть бути спроектовані в одному середовищі, а виконані – в іншому, без втрати структури чи логіки.

Взаємодія між системами може відбуватись як синхронно, так і асинхронно. У першому випадку компоненти обмінюються запитом й відповідями у реальному часі, що забезпечує високу швидкість реакції, але вимагає стабільного підключення та низької затримки. В асинхронних сценаріях обмін даними відбувається через черги повідомлень або шини подій, що підвищує надійність і дозволяє масштабувати систему. Обидва підходи використовуються залежно від типу завдань і навантаження на систему.

У результаті така взаємодія створює єдине інтегроване середовище, у якому користувацькі запити перетворюються на формалізовані моделі, що можуть бути одразу виконані або збережені для подальшої обробки. Це сприяє розвитку концепції «розумних» бізнес-систем, у яких процес моделювання, виконання й оптимізації здійснюється автоматично, з мінімальним залученням людини.

#### 1.4.4 Переваги та обмеження використання інтелектуальних агентів

Використання агентного підходу має безперечні переваги. Насамперед, це суттєве скорочення часу, необхідного для створення моделей, адже система автоматично формує більшість елементів. Крім того, виключається ризик людських помилок – агент не допускає логічних або синтаксичних невідповідностей у BPMN. Ще однією перевагою є гнучкість системи: вона може адаптуватися до різних предметних галузей і розширюватися шляхом додавання нових шаблонів і сценаріїв [3].

Водночас існують і певні обмеження. Найважливіше з них пов'язане з неоднозначністю природної мови, коли агент може неправильно інтерпретувати опис користувача. Також виклики виникають у забезпеченні якості кінцевого результату, адже навіть синтаксично правильна модель може не відповідати реальній бізнес-логіці. Крім того, використання великих мовних моделей вимагає значних обчислювальних ресурсів, що може обмежувати застосування таких систем у локальних середовищах.

#### 1.4.5 Перспективи розвитку агентних технологій у BPMN-моделюванні

Подальший розвиток агентних систем у BPMN-моделюванні спрямований на створення інтерактивних середовищ, у яких користувач зможе спілкуватися з агентом у діалоговій формі. Така взаємодія дозволить системі уточнювати

незрозумілі частини опису, вносити правки у процес побудови та адаптувати модель до контексту конкретної організації. Перспективним напрямом є також інтеграція онтологій предметних областей, що дасть змогу агентам розуміти специфічні терміни та правила для кожної галузі.

Очікується, що інтелектуальні агенти з часом стануть невід'ємною частиною BPMS нового покоління. Вони не лише створюватимуть діаграми, але й здійснюватимуть оптимізацію бізнес-процесів, аналізуватимуть ефективність виконання, виявлятимуть вузькі місця та пропонуватимуть шляхи їх усунення. Такий підхід відкриває шлях до появи самонавчальних систем управління процесами, здатних еволюціонувати разом із бізнесом.

### 1.5 Постановка задачі дослідження

У сучасних умовах цифрової трансформації підприємств автоматизація бізнес-процесів набуває стратегічного значення. Моделювання у нотації BPMN стало ключовим інструментом візуального відображення процесів, який забезпечує єдине розуміння логіки діяльності між аналітиками, розробниками та управлінцями. Проте створення таких моделей залишається трудомістким завданням, що потребує високого рівня технічної підготовки та знання специфічних інструментів. З огляду на це, виникає необхідність у розробленні підходів, які б поєднували гнучкість моделювання з інтелектуальними можливостями систем штучного інтелекту [4].

Таким чином, дослідження спрямоване на розроблення підходу до моделювання бізнес-процесів, який використовує інтелектуальних агентів для аналізу текстових даних і побудови структурованих BPMN-моделей. Актуальність роботи полягає у прагненні поєднати традиційні методи процесного моделювання з сучасними когнітивними технологіями, що дозволяють скоротити час розробки моделей і підвищити точність опису бізнес-логіки.

Об'єктом дослідження є процес побудови бізнес-процесів у нотації BPMN у системах управління бізнес-процесами.

Предметом дослідження є методи автоматичного перетворення текстових описів бізнес-процесів у BPMN-моделі з використанням інтелектуальних агентів.

Метою дослідження є формування підходу до автоматизації створення BPMN-діаграм на основі аналізу природної мови з використанням інтелектуальних агентів.

Для досягнення поставленої мети необхідно вирішити такі основні завдання дослідження:

- здійснити аналіз сучасних методів моделювання бізнес-процесів і технологій автоматизації побудови BPMN-моделей;
- вивчити можливості застосування інтелектуальних агентів у процесному моделюванні;
- дослідити принципи обробки природної мови для виявлення логіки бізнес-процесів;
- визначити структуру системи, що поєднує аналіз тексту та генерацію BPMN-діаграми;
- провести експериментальне моделювання й оцінити ефективність запропонованого підходу.

Наукова новизна дослідження полягає в поєднанні аналізу природної мови з процесним моделюванням, що дає змогу створювати BPMN-моделі зі звичайних текстових описів без спеціальних технічних знань. Практичне значення полягає в тому, що запропонований підхід прискорює моделювання процесів, підвищує точність бізнес-описів і зменшує витрати часу на їх проектування [5].

Результати цього дослідження можуть бути використані в інформаційних системах управління бізнес-процесами, у розробленні корпоративних BPMS-рішень, а також у навчальних і дослідницьких цілях при вивченні дисциплін, пов'язаних із бізнес-аналітикою, системним аналізом і штучним інтелектом.

## 2 МОДЕЛІ ТА МЕТОДИ АВТОМАТИЗАЦІЇ ПОБУДОВИ BPMN-ДІАГРАМ

### 2.1 Модель обробки природної мови в задачах побудови BPMN

У процесі автоматизованого створення бізнес-процесів у нотації BPMN одним із центральних компонентів є модель обробки природної мови. Вона забезпечує перехід від неструктурованого текстового опису, який вводить користувач, до структурованих даних, що можуть бути використані для генерації BPMN-елементів. Саме завдяки застосуванню NLP відбувається перетворення людських фраз у зрозумілу для машини логіку, яка згодом відображається у вигляді процесної діаграми.

#### 2.1.1 Основні принципи обробки природної мови

Обробка природної мови як напрям досліджень спрямована на те, щоб навчити комп'ютер «розуміти» людську мову. Під «розумінням» у цьому контексті мається на увазі здатність системи ідентифікувати смислові структури, залежності між словами, типи дій, учасників подій і логічні зв'язки між ними. Для цього використовується багаторівнева архітектура, що включає:

- лексичний рівень, який відповідає за ідентифікацію слів, частин мови, морфологічних ознак;
- синтаксичний рівень, де визначається структура речення, порядок слів, залежності між підметом, присудком та обставинами;
- семантичний рівень, на якому аналізується значення висловлювань і виявляються дії, об'єкти та актори;
- прагматичний рівень, який дозволяє системі інтерпретувати текст з урахуванням контексту, цілі користувача та предметної області.

Для задачі побудови BPMN-діаграм усі ці рівні взаємопов'язані. Наприклад, речення «Клієнт оформлює замовлення, після чого система перевіряє

оплату» на лексичному рівні визначає ключові слова клієнт, оформлює, замовлення, перевіряє, оплату; на синтаксичному – виділяє дві дії (створення замовлення і перевірку оплати); на семантичному – встановлює послідовність цих дій, а на прагматичному – формує логічний зв’язок між актором (клієнтом) і системною дією (перевірка оплати) [6, 7].

### 2.1.2 Використання мовних моделей для семантичного аналізу

Сучасні системи NLP у задачах автоматизації моделювання бізнес-процесів спираються на великі мовні моделі, які навчені на величезних обсягах текстових даних. Їхня перевага полягає у здатності вловлювати контекст і розуміти складні зв’язки між словами. У рамках побудови BPMN це дозволяє не лише визначати дії, а й розрізняти ролі учасників, умови, паралельні процеси, виключення тощо.

Наприклад, якщо користувач описує: «Менеджер перевіряє заявку. Якщо вона неповна, система надсилає клієнту повідомлення, і процес повторюється» – мовна модель може інтерпретувати таку умовну конструкцію («якщо неповна – тоді повідомлення»), а також розпізнати цикл повторного виконання дії («процес повторюється»). Ці логічні взаємозв’язки згодом перетворюються на відповідні елементи BPMN: умовне розгалуження, подію надсилання повідомлення та позначку повторюваності.

Мовні моделі використовують підхід контекстного векторного представлення тексту [8]. Це означає, що кожне слово або фраза перетворюється на вектор чисел у багатовимірному просторі, де близькі за змістом слова мають схожі координати. Саме це дозволяє системі не лише розпізнавати слова, а й інтерпретувати їх значення у контексті конкретної задачі.

### 2.1.3 Етапи перетворення тексту у формальні структури

Процес обробки тексту для побудови BPMN-моделей зазвичай складається з кількох послідовних етапів.

Етап 1. Попередня обробка – очищення тексту, розбиття на слова та речення, лематизація.

Етап 2. Граматичний аналіз – визначення частин мови та синтаксичних залежностей.

Етап 3. Семантична інтерпретація – перетворення тексту на структурований опис із виділенням суб'єктів, дій, об'єктів, умов і послідовностей.

Етап 4. Формування логічної структури – побудова послідовності дій, переходів і залежностей, придатних для подальшого відображення у BPMN; результатом є набір структурованих сутностей, наприклад:

Лістинг 2.1 Реалізація перевірки відкриття сторінки:

```
{"actor": "client", "action": "create order"}  
{"actor": "system", "action": "verify payment", "condition": "after order created"}
```

Ці сутності є вхідними даними для наступного рівня – генератора BPMN-елементів, який уже формує XML-файли відповідно до нотації BPMN 2.0.

## 2.2 Формальна модель перетворення тексту в BPMN-структуру

У сучасних дослідженнях автоматизації моделювання бізнес-процесів ключову роль відіграє завдання перетворення природномовного опису процесу в формальну BPMN-структуру. Такий підхід дозволяє скоротити часові витрати на проектування, знизити ймовірність помилок при побудові моделей і забезпечити доступність інструментів BPMN для користувачів без глибоких

технічних знань. Формальна модель цього перетворення є теоретичним підґрунтям для реалізації інтелектуальних систем, які використовують методи обробки природної мови та великі мовні моделі.

Дослідження W. Li та C. Toxtli (2024) показало, що великі мовні моделі, зокрема GPT-4o [10], здатні автоматично формувати валідні BPMN-діаграми у форматі XML, придатні для імпорту в системи керування бізнес-процесами, такі як Camunda. Науковці експериментально довели, що генерація процесних моделей може здійснюватися навіть у режимі нульового навчання, тобто без попереднього навчання на спеціальних наборах BPMN-даних. Цей висновок підтверджує практичну доцільність побудови формальної моделі, яка забезпечує трансформацію тексту у структуровану нотацію BPMN.

### 2.2.1 Концептуальні основи формальної моделі

Будь-який текстовий опис бізнес-процесу містить ключові компоненти, які можна зіставити з елементами BPMN. У формальному поданні процес визначається як п'ятірка множин:

$$P = (S, A, C, E, F) , \quad (2.1)$$

де  $S$  – множина суб'єктів (акторів, виконавців процесу);

$A$  – множина дій (операцій, що виконуються в межах процесу);

$C$  – множина умов (логічних перевірок або розгалужень);

$E$  – множина подій (початкових, проміжних, завершальних);

$F$  – множина потоків, що визначають порядок виконання дій.

Це подання дозволяє інтелектуальній системі сприймати текст як набір семантичних одиниць, пов'язаних між собою логічними зв'язками. Кожен елемент тексту (наприклад, дієслово дії, підмет, умова, або подія) має своє відображення у відповідному BPMN-компоненті [11, 12].

Таким чином, формальна модель виступає своєрідним «каркасом» між людським описом і технічною структурою процесу (рис. 2.1).

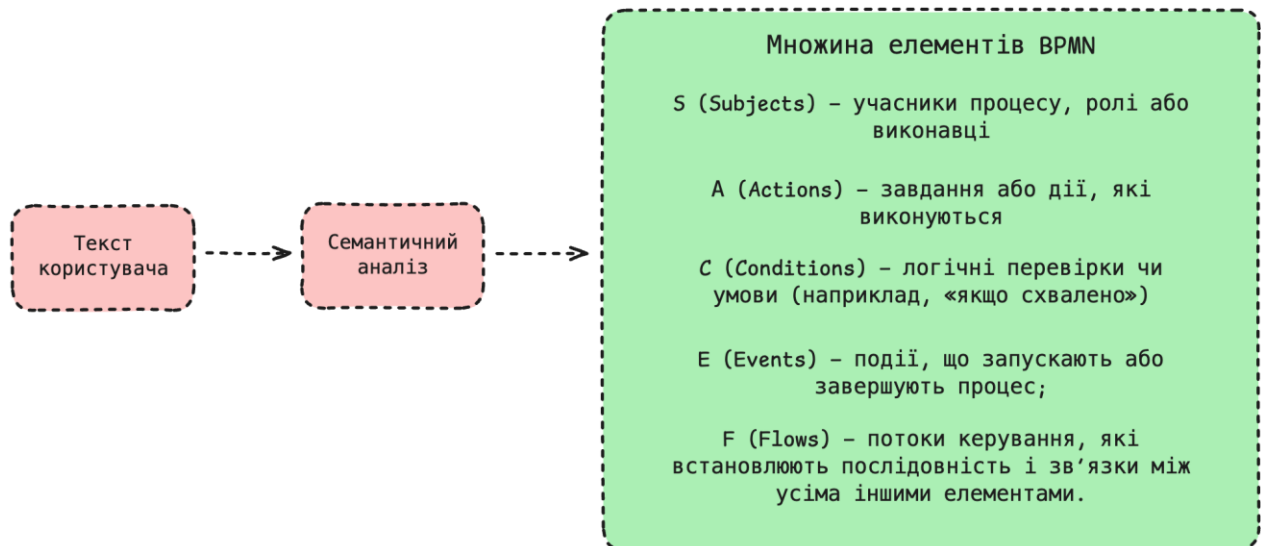


Рисунок 2.1 – Концептуальна схема відповідності між текстом користувача та множинами елементів BPMN

### 2.2.2 Семантичні правила класифікації елементів процесу

Ключовим завданням формальної моделі є встановлення правил, які визначають, як саме фрагменти тексту перетворюються на BPMN-елементи. Для цього використовується набір правил відображення, що мають вигляд:

$$r_i: \text{semantic pattern} \rightarrow \text{BPMN element type}, \quad (2.2)$$

де  $r_i$  – окреме правило відображення, що належить множині правил  $R$ ;

*semantic pattern* – семантичний шаблон або лінгвістична конструкція, яка містить ознаки певної дії, події, умови чи ролі в бізнес-процесі;

*BPMN element type* – тип елемента BPMN (наприклад: службове завдання, користувацьке завдання, виключне розгалуження, початкова подія, кінцева подія), який відповідає розпізаному семантичному шаблону.

Навіть без спеціальних шаблонів моделі Gemini 2.5 Flash здатна розпізнавати такі семантичні структури та коректно відтворювати відповідні елементи BPMN. Це свідчить про наявність в мовній моделі базового розуміння процесної логіки, що дозволяє автоматично класифікувати події, завдання та умови у тексті [13].

### 2.2.3 Формування потоків керування

Після класифікації всіх елементів формується система зв'язків між ними – потоки керування. Вони визначають порядок виконання операцій і логіку переходів між завданнями, умовами та подіями:

$$F = \{(n_i, n_j) | n_i, n_j \in N, n_i \rightarrow n_j\}, \quad (2.3)$$

де  $N$  – множина усіх вузлів BPMN-моделі;

$n_i$  – окремий елемент моделі, що належить множині  $N$ ;

$n_j$  – елемент моделі, до якого здійснюється перехід від  $n_i$ ;

Цей етап є критично важливим, адже саме потоки визначають логічну цілісність процесу. У роботі зазначається, що навіть у згенерованих мовною моделлю BPMN-файлах найбільша кількість помилок припадає саме на неправильне формування потоків – зокрема, на дублювання з'єднань або відсутність виходу з вузла. Саме тому у формальній моделі особливу увагу приділяють перевірці напрямків і коректності зв'язків [14].

### 2.2.4 Валідація BPMN-моделі

Формальна перевірка є завершальним етапом побудови процесної моделі. Вона забезпечує відповідність структури BPMN основним правилам нотації та

вимогам до її виконання у BPMN-середовищах. Для цього перевіряються такі умови:

$$\forall n_i \in N, \exists in(n_i), out(n_i), \quad (2.4)$$

де  $N$  – множина всіх елементів процесу;

$n_i$  – окремий елемент моделі, що належить множині  $N$ ;

$in(n_i)$  – множина всіх вхідних потоків для елемента  $n_i$ ;

$out(n_j)$  – множина всіх вихідних потоків для елемента  $n_j$ .

Ця формула означає, що кожен елемент має вхід і вихід, а також що процес має єдиний початок і принаймні один кінець [15].

Дослідження показує, що валідація залишається одним із найважливіших критеріїв якості BPMN-моделей. У своїй роботі автори пропонують шість показників для оцінювання діаграм: коректність, повнота, деталізованість, якість позначення потоків, зрозумілість та коректність делегування.

Результати експериментів показали в таблиці 2.1, що GPT-4o продемонструвала високі результати за критерієм валідності та читабельності, але мала нижчі показники щодо повноти та правильності зв'язків.

Таблиця 2.1 – Показники якості згенерованих BPMN-діаграм у дослідженні

<b>Критерій</b>	<b>Середній показник</b>
Валідність	0,87
Повнота	0,65
Деталізація	0,72
Правильність стрілок	0,68
Читабельність	0,91
Делегування	0,74

## 2.3 Формалізація процесу побудови BPMN-діаграм за допомогою моделей обробки природної мови

Формалізація процесу автоматичної побудови BPMN-діаграм передбачає перехід від текстового опису до структурованої, машинно-читаної моделі процесу. На цьому етапі застосовуються алгоритми NLP [16, 17], методи інформаційного витягання, а також інтелектуальні моделі, здатні здійснювати семантичний аналіз контексту. Основна мета – забезпечити однозначну відповідність між мовними конструкціями користувача та елементами процесної моделі.

### 2.3.1 Етапи формалізації тексту бізнес-процесу

Процес формалізації (рис. 2.2) включає кілька взаємопов'язаних етапів.

Етап 1. синтаксичний аналіз, на цьому кроці текст розбивається на речення, частини мови та синтаксичні залежності. Використовуються моделі на зразок spaCy, Stanza або мовні моделі Google Gemini, які можуть ідентифікувати головні дієслова (дії), підмети (виконавців) і об'єкти (результати дії) [18].

Етап 2. семантичне розпізнавання, кожен компонент речення отримує семантичну мітку відповідно до типу BPMN-елемента (наприклад, задача, шлюз, подія). Наприклад, фраза «Клієнт заповнює заявку» – (користувацька задача), тоді як «Якщо дані некоректні» – (виключний шлюз).

Етап 3. логічне структурування, виявлені елементи об'єднуються у структуру з логічними зв'язками між подіями, задачами й умовами. створюються внутрішні моделі, що визначають послідовність виконання процесу.

Етап 4. формування BPMN-моделі, на цьому етапі формується XML-представлення діаграми згідно з нотацією BPMN 2.0. Кожен елемент має відповідний шаблон XML, у якому заповнюються атрибути.

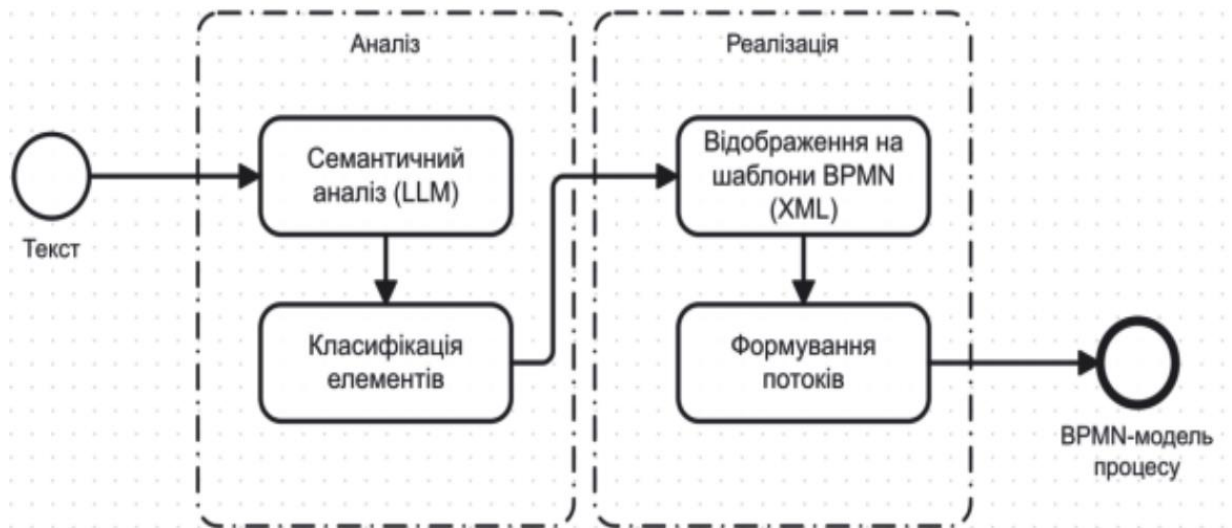


Рисунок 2.2 – Етапи формалізації текстового опису бізнес-процесу у BPMN-структуру

### 2.3.2 Використання моделей для контекстного розпізнавання

Моделі, зокрема OpenAI GPT, Gemini або Claude, демонструють здатність розуміти контекст тексту і логічно пов'язувати події. Це відкриває можливість не лише аналізувати окремі речення, а й формувати цілісну структуру процесу, ураховуючи залежності між його частинами (рис. 2.3) [19].

Мовна модель може, наприклад, розпізнати, що слова «після перевірки» означають завершення попереднього етапу й ініціацію наступного – що безпосередньо відповідає створенню послідовний потік у BPMN.

Такі моделі не потребують попереднього програмування правил – вони здатні самостійно будувати логічні ланцюги завдяки попередньому навчанні на великих текстових корпусах..

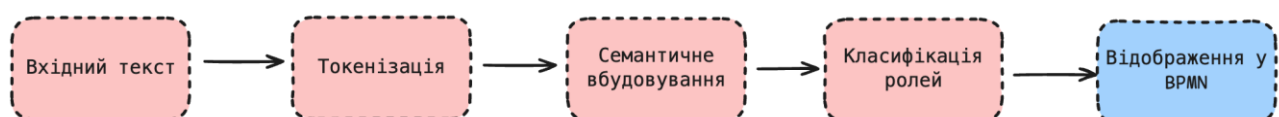


Рисунок 2.3 – Логічна модель обробки тексту мовною моделлю під час побудови BPMN

### 2.3.3 Формування потоків та залежностей між елементами

На фінальному етапі відбувається зв'язування елементів у послідовну логічну структуру.

Система встановлює напрямки виконання (старт – задача – подія – кінець), а також визначає умови переходу через виключні або паралельні шлюзи.

Наприклад, якщо текст містить конструкції «одночасно», «паралельно» – формується паралельний шлюз, а якщо – «у разі», «якщо» – створюється виключний шлюз.

Кожен такий зв'язок (рис. 2.4) формалізується у вигляді тегу `<bpml:sequenceFlow>` у XML-файлі. Це забезпечує повну структурну цілісність діаграми, придатну для імпорту у середовища на кшталт Camunda Modeler.

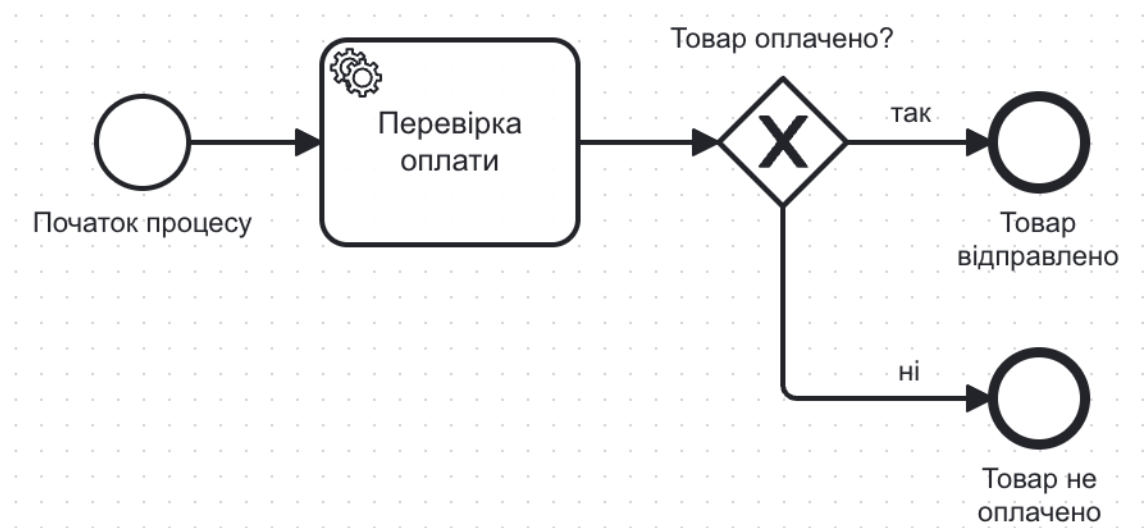


Рисунок 2.4 – Формування потоків між BPMN-елементами відповідно до логічних зв'язків тексту

## 2.4 Алгоритм перетворення текстового опису бізнес-процесу у BPMN-діаграму

Автоматичне створення BPMN-діаграми на основі текстового опису процесу – це складне завдання, яке поєднує методи обробки природної мови,

семантичного аналізу та генерації структурованих XML-моделей [20, 21]. Основною метою цього етапу є формування узгодженої логічної моделі процесу, що відповідає правилам BPMN 2.0 та може бути імпортована до системи управління бізнес-процесами, наприклад Camunda.

Процес побудови BPMN можна представити як послідовність взаємопов'язаних етапів, кожен із яких перетворює текст у дедалі більш структуровану форму, доки не буде отримано повноцінну модель процесу.

#### 2.4.1 Попередня обробка тексту

Першим кроком у побудові BPMN-моделі є попередня обробка текстових даних.

На цьому етапі здійснюється нормалізація тексту – очищення від зайвих символів, поділ на речення, а також лематизація, яка дозволяє звести усі форми слова до початкової.

Так, наприклад, фрази «перевіряє», «перевірити», «перевірка» розпізнаються як одне базове поняття – перевірка [22].

Це важливо, адже для правильного визначення BPMN-елементів система має розуміти не лише синтаксичні, а й смислові відношення між словами.

На цьому етапі застосовуються інструменти NLTK, spaCy, або ж мовні моделі типу Gemini чи GPT, які здатні не лише аналізувати структуру речення, але й інтерпретувати його контекст.

#### 2.4.2 Визначення типів елементів процесу

Після попереднього аналізу здійснюється класифікація фрагментів тексту на відповідні типи BPMN-елементів.

Цей етап є ключовим, оскільки саме тут відбувається перехід від природної мови до формальної моделі. Наприклад:

- фрази, що описують дії людини, розпізнаються як користувацька задача («Клієнт заповнює анкету»);
- автоматизовані операції системи – як сервісна задача («Система перевіряє дані»);
- умовні вирази – як виключний шлюз («Якщо дані некоректні»);
- повідомлення – як задача повідомлення («Надсилається сповіщення клієнту»);
- завершення процесу – як кінцева подія.

Цей механізм ґрунтується на семантичних шаблонах, тобто на заздалегідь визначених структурах, які система зіставляє з текстом користувача (рис. 2.5).

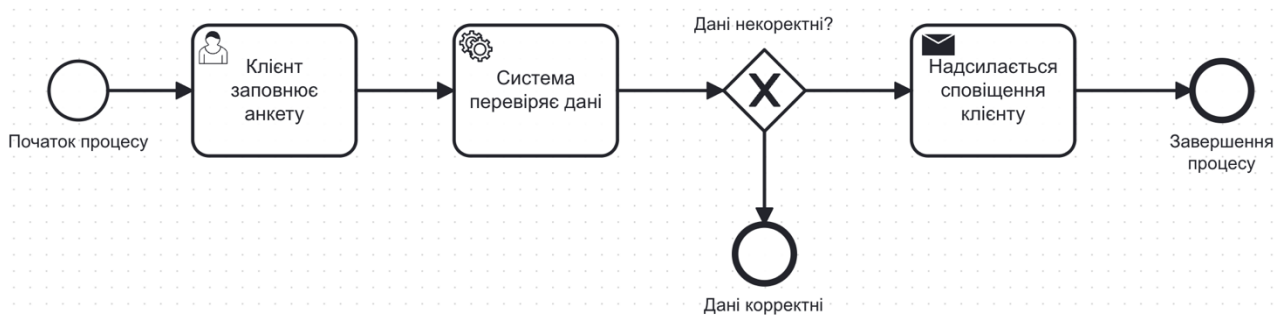


Рисунок 2.5 – Приклад семантичного зіставлення фраз із типами BPMN-елементів

### 2.4.3 Побудова логічної структури процесу

Після класифікації елементів необхідно визначити логічну послідовність їх виконання. Це завдання вирішується шляхом аналізу причинно-наслідкових зв'язків у тексті. Так, слова-маркери, як-от «після», «далі», «у разі», допомагають моделі встановити порядок етапів [22, 23].

На цьому кроці створюється граф процесу, де вершини – це елементи BPMN, а ребра – потоки, які визначають послідовність або розгалуження.

Мовна модель виконує роль аналітичного агента, що вибудовує ці зв'язки автоматично, враховуючи як синтаксис, так і контекст опису.

#### 2.4.4 Формування повної BPMN-діаграми

На фінальному етапі усі елементи об'єднуються в одну цілісну діаграму.

Система додає необхідні початкову подію, кінцеву подію та потік послідовності, перевіряє коректність з'єднань і формує остаточну XML-структуру.

Отриманий результат є валідним BPMN-документом, який можна відкрити у Camunda Modeler або безпосередньо запустити у Camunda.

Користувач може переглянути, змінити або доповнити діаграму, але основна робота з її побудови вже виконана автоматично.

### 2.5 Архітектура системи автоматичної генерації BPMN-діаграм

Архітектура системи автоматичної побудови BPMN-діаграм із текстових описів процесів є поєднанням технологій обробки природної мови, зберігання структурованих шаблонів і механізмів візуалізації процесів [24]. Основна ідея полягає в тому, щоб створити узгоджену модель, у якій інтелектуальний агент (мовна модель) виступає як аналітичний компонент, МСР-сервер – як сховище структур BPMN, а система управління бізнес-процесами – як середовище для подальшого виконання або редагування діаграм.

#### 2.5.1 Загальна структура архітектури

У загальному вигляді така архітектура складається з трьох взаємопов'язаних рівнів: інтерфейсного, аналітичного та системного. На інтерфейсному рівні користувач вводить текстовий опис процесу, який може бути як простим («Клієнт подає заявку – система перевіряє дані»), так і складним із умовами, паралельними діями чи таймерами.

Цей опис передається на аналітичний рівень, де здійснюється обробка тексту, визначення структурних зв'язків, класифікація фраз і перетворення їх у модель елементів BPMN.

Тут працює мовна модель, яка ідентифікує ролі, події, завдання, логічні розгалуження та залежності.

### 2.5.2 Логіка послідовної обробки даних

Архітектура функціонує за принципом послідовної обробки даних. Спочатку користувач вводить текст – це точка входу системи. Потім мовна модель проводить семантичний аналіз, виявляє структуру процесу й визначає, які BPMN-елементи відповідають кожному етапу опису [25].

Після цього система звертається до MCP-серверу, щоб отримати XML-шаблони відповідних елементів. Кожен шаблон заповнюється атрибутами (ID, назва, тип дії) і вставляється до єдиної структури BPMN.

Фінальний етап – візуалізація діаграми, яка може виконуватися у Camunda Modeler або у вебінтерфейсі з бібліотекою bpmn-js.

### 2.5.3 Роль мовної моделі у системі

У системі автоматичної побудови [26, 27] BPMN-діаграм ключову роль відіграє – велика мовна модель, яка виступає не просто генератором тексту, а когнітивним аналітичним агентом. Її головне завдання – глибоко зрозуміти зміст введеного користувачем тексту, розпізнати в ньому структуру дій, логічні залежності, умови, ролі та події, і на цій основі побудувати формалізовану модель процесу.

Мовна модель аналізує контекст не лише на рівні окремих слів, а й визначає логічну послідовність дій. Наприклад, у реченні «Після перевірки даних система надсилає клієнту сповіщення» модель розуміє, що дія «перевірка»

відбувається перед дією «надсилання» і створює між ними відповідний перехід у процесі. Це дозволяє автоматично будувати логіку процесу без втручання людини.

Особливістю мовних моделей є їхня здатність виконувати семантичну класифікацію ролей, тобто розуміти, хто саме здійснює певну дію — користувач, система чи сторонній учасник. Завдяки цьому модель може правильно визначити тип BPMN-елемента та виокремити системні й користувацькі дії, етапи очікування чи затримки, а також можливі альтернативні сценарії розвитку процесу.

Таким чином, мовна модель виступає як «мозок» усієї архітектури, який виконує аналіз природної мови, інтерпретує зміст і визначає структуру майбутньої моделі. Вона перетворює неструктурований текст у когнітивно зв'язану схему, що поєднує семантичну логіку з формальними правилами BPMN 2.0.

Крім того, мовна модель здатна адаптуватися до контексту предметної області. У фінансовій сфері вона виявляє терміни на кшталт «кредит», «рахунок», «перевірка клієнта» та коректно будує процеси на їх основі. У логістиці вона розпізнає етапи доставки, складські операції, транспортні події тощо. Така адаптивність робить систему універсальною та придатною до використання у будь-якому бізнес-середовищі.

Роль мовної моделі полягає також у створенні когнітивного мосту між людиною та системою BPMN. Користувач формулює завдання звичною мовою, не потребуючи технічних знань, а модель переводить його опис у формат, зрозумілий для BPMS-систем. Це значно спрощує комунікацію між аналітиками, розробниками та кінцевими користувачами [28].

У підсумку, мовна модель забезпечує інтелектуальне ядро системи, здатне не лише аналізувати текст, а й навчатися з часом, підвищуючи точність класифікації та якість побудови моделей. Це відкриває шлях до створення посправжньому «розумних» інструментів для автоматизованого бізнес-моделювання.

#### 2.5.4 МСР-сервер як джерело структурної істини

МСР-сервер є центральним компонентом архітектури, який забезпечує структурну коректність та узгодженість BPMN-моделей. Якщо мовна модель виступає «мозком», то МСР-сервер – це «пам'ять» і «база знань» системи, де зберігаються всі XML-шаблони BPMN-елементів, їх атрибути та можливі варіанти взаємозв'язків.

Сервер містить репозиторій стандартних XML-фрагментів, що відповідають нотації BPMN 2.0, наприклад: сервісна задача, користувацька задача, виключний шлюз, задача надсилання, подія-таймер, кінцева подія. тощо. Кожен елемент зберігається у форматі, який можна динамічно заповнювати через змінні. Це дає змогу формувати складні моделі шляхом складання готових блоків.

Коли мовна модель визначає тип елемента, вона надсилає запит до МСР-сервер у вигляді:

Лістинг 2.2 Приклад запиту від моделі до МСР-сервер:

```
{
  "action": "getElement",
  "params": { "element_type": "serviceTask", "camunda_version":
"Camunda8" }
}
```

У відповідь сервер повертає XML-фрагмент, який вставляється у спільну структуру BPMN-моделі. Наприклад:

Лістинг 2.3 Приклад відповіді МСР-сервера:

```
<bpmn:userTask          id="UserTask_123"          name="Перевірка
заявки"></bpmn:userTask>
```

Таким чином, MCP-сервер гарантує, що всі елементи відповідають єдиному синтаксису BPMN 2.0, незалежно від того, яка мовна модель або інтерфейс їх створює. Це дозволяє без проблем імпортувати згенеровані діаграми у Camunda, Zeebe, Flowable чи інші BPMS-системи.

Крім того, MCP-сервер виступає джерелом структурної істини. Він не лише забезпечує правильність формату, а й контролює зв'язність між елементами, щоб уникнути логічних помилок – наприклад, відсутності завершальної події або розриву потоку. Сервер може також перевіряти валідність моделей, попереджаючи користувача про порушення правил BPMN.

У майбутньому MCP-сервер може бути розширений підтримкою різних версій шаблонів для Camunda 7, Camunda 8 або Zeebe Cloud, а також зберігати варіативні елементи, що адаптуються до конкретних бізнес-доменів [29].

Таким чином, MCP-сервер є не просто сховищем даних, а архітектурним стрижнем, який забезпечує стабільність, відповідність стандартам і сумісність між усіма компонентами системи автоматичної генерації BPMN.

### 2.5.5 Гнучкість, масштабованість і адаптивність архітектури

Однією з найважливіших характеристик запропонованої архітектури є її гнучкість і масштабованість, що дозволяє системі ефективно адаптуватися до різних сценаріїв використання, типів бізнес-процесів і технологічних середовищ. Гнучкість проявляється у здатності системи змінювати окремі компоненти або модулі без порушення загальної логіки її роботи. Наприклад, мовну модель можна легко замінити на іншу – від OpenAI GPT-4 до Google Gemini або будь-якої спеціалізованої моделі, навченої для певного домену. Це забезпечує незалежність архітектури від конкретного постачальника ШІ-рішень та дозволяє обирати оптимальні інструменти під конкретні бізнес-вимоги.

Іншим важливим аспектом є масштабованість MCP-сервер, який відповідає за зберігання XML-шаблонів BPMN-елементів. Оскільки кількість типів елементів і складність процесів постійно зростають, система має бути

здатною обробляти значні обсяги запитів одночасно. Масштабування може здійснюватися горизонтально – шляхом розподілення навантаження між кількома вузлами бази даних, або вертикально – через збільшення потужності обчислювальних ресурсів. Завдяки цьому навіть при великій кількості користувачів система зберігає стабільність і швидкість формування діаграм.

Не менш важливою характеристикою є адаптивність архітектури. Завдяки динамічному контексту мовної моделі може налаштовувати свій стиль аналізу під різні галузі. У банківському секторі модель може фокусуватися на кредитних і фінансових процедурах, у логістиці – на потоках товарів і документації, а в медицині – на маршрутах пацієнтів і перевірках даних. Таким чином, система стає універсальною платформою, яка пристосовується до термінології, ролей та логіки конкретної сфери діяльності.

Гнучкість архітектури також проявляється у модульності – можливості незалежного розвитку компонентів. Наприклад, фронтенд-інтерфейс, створений на базі Next.js, може оновлюватися або змінювати дизайн без необхідності втручання у логіку генерації BPMN [30]. Так само оновлення шаблонів у MCP-сервер або введення нових типів елементів не потребує зміни мовної моделі. Цей підхід суттєво спрощує підтримку системи, полегшує інтеграцію з новими сервісами та скорочує час адаптації під потреби користувачів.

Окремої уваги заслуговує можливість розгортання системи у хмарному середовищі. У такій конфігурації всі її компоненти – мовна модель, MCP-сервер і модуль генерації BPMN – можуть функціонувати як окремі мікросервіси, що взаємодіють через інтерфейс додатка. Це відкриває перспективу масштабування не лише з погляду продуктивності, а й географічного розподілення – коли запити користувачів обробляються найближчим до них регіональним сервером, мінімізуючи затримки. Хмарна архітектура також забезпечує високу відмовостійкість: у разі збоїв одного з компонентів інші продовжують роботу без зупинки всієї системи.

### 3 РЕАЛІЗАЦІЯ СИСТЕМИ АВТОМАТИЧНОГО МОДЕЛЮВАННЯ БІЗНЕС-ПРОЦЕСІВ

#### 3.1 Загальна характеристика системи

Розроблена система автоматичного моделювання бізнес-процесів у нотації BPMN з використанням інтелектуальних агентів є комплексним програмним рішенням, що поєднує сучасні методи обробки природної мови, семантичного аналізу та формального моделювання процесів. Її головною метою є спрощення побудови бізнес-процесів – замість ручного створення діаграм користувач описує процес природною мовою, а система автоматично формує структуровану BPMN-модель, придатну для подальшої роботи в BPMS-середовищі, зокрема в Camunda [31, 32].

Реалізація системи базується на тривірневій архітектурі (рис. 3.1), що складається з інтерфейсного, аналітичного та системного рівнів, які взаємодіють між собою за принципом послідовного обміну даними. На інтерфейсному рівні користувач вводить текстовий опис процесу через вебінтерфейс, після чого запит передається до аналітичного рівня, де обробляється мовною моделлю. Вона виконує семантичний аналіз, визначає логічну структуру опису, класифікує дії, події, учасників і залежності між ними. Потім результат надсилається до системного рівня, де MCP-сервер підставляє відповідні XML-шаблони BPMN-елементів і формує повну діаграму процесу. Основні вимоги до системи передбачали:

- здатність приймати текст природною мовою українською або англійською;
- підтримку стандарту BPMN 2.0 для сумісності з іншими BPMS-платформами;
- можливість інтеграції з мовними моделями (Gemini, GPT-4, Claude тощо);
- швидкість генерації моделі не більше 5 секунд для процесів середньої складності;

– збереження згенерованих діаграм у форматі .bpmn для подальшого імпорту до Camunda Modeler.

Загальна структура системи включає такі основні компоненти:

– користувацький інтерфейс: забезпечує взаємодію з користувачем, введення текстового опису, візуалізацію BPMN-діаграми через бібліотеку bpmn-js, а також можливість завантаження отриманої моделі у вигляді файлу;

– серверна частина: реалізує логіку обробки запитів, комунікацію з мовною моделлю та MCP-сервером, обробку XML-фрагментів і формування BPMN-моделі;

– база даних: містить таблиці зі збереженими шаблонами BPMN-елементів, параметрами процесів і історією запитів користувачів;

– MCP-сервер: відповідає за зберігання та надання структурованих XML-шаблонів елементів BPMN;

– API мовної моделі: виконує функцію аналітичного агента, який проводить семантичний аналіз введеного тексту, ідентифікує сутності та визначає структуру процесу [33, 34];

– Camunda: використовується для тестування та подальшого виконання побудованих моделей, що дає змогу перевіряти їхню функціональність у реальному BPMN-середовищі.

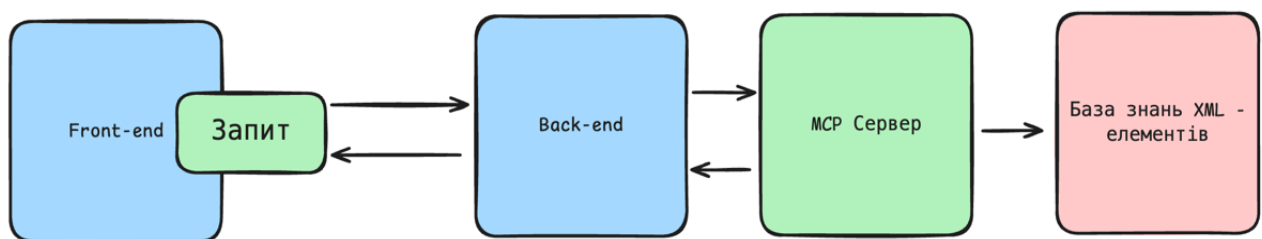


Рисунок 3.1 – Інтерфейс користувача системи з відображенням згенерованого процесу

### 3.2 Реалізація серверної частини

Серверна частина системи є центральним компонентом, який забезпечує зв'язок між усіма рівнями архітектури – користувацьким інтерфейсом, мовною моделлю, базою даних і МСР-сервер. Вона виконує роль «посередника» між аналітичним та системним рівнями, приймаючи запит користувача, координуючи процес генерації BPMN-діаграми та повертаючи результат у вигляді готового XML-файлу [35].

Розробка серверної частини була реалізована мовою Go з використанням фреймворку Gin. Вибір цієї технології обумовлений високою продуктивністю, простотою паралельної обробки запитів і зручним механізмом маршрутизації. Архітектура бекенду є модульною, що дозволяє легко розширювати або замінювати окремі частини системи – наприклад, логіку взаємодії з моделлю або механізм роботи з базою даних.

Основними функціональними модулями серверної частини є такі компоненти:

- маршрутизатор: обробляє зовнішні запити з інтерфейсу користувача та скеровує їх до відповідних керувачів;
- керувач: реалізує основну логіку генерації BPMN-діаграм;
- сервісний рівень: відповідає за взаємодію з мовною моделлю та сервером шаблонів;
- рівень сховища даних: забезпечує роботу з базою даних, зокрема з таблицями `s8_bpmn_elements` і `s8_bpmn_root`;
- журналізатор: фіксує всі етапи обробки запиту для подальшого аналізу;
- завершальний етап роботи: після формування діаграми користувач може відкрити її у Camunda Modeler для редагування або запустити виконання в хмарному середовищі Camunda.

Кожен компонент системи реалізує чітко визначену функцію, що забезпечує модульність і масштабованість рішення. Наприклад, мовна модель може бути замінена без потреби зміни решти архітектури – достатньо адаптувати запити до нового API. Аналогічно, МСР-сервер можна розширювати, додаючи

шаблони для нових типів елементів BPMN або підключати альтернативні джерела даних.

### 3.2.1 Структура запиту генерації BPMN

Процес взаємодії між клієнтською частиною та сервером починається з моменту, коли користувач надсилає текстовий опис бізнес-процесу через інтерфейс системи. Цей опис є вхідними даними, на основі яких відбувається подальший аналіз, класифікація елементів і побудова BPMN-діаграми.

Тіло запиту має формат JSON і містить лише один основний параметр – текстовий опис процесу, який користувач вводить природною мовою. На цьому етапі сервер отримує вхідні дані, виконує їхню базову перевірку та передає на подальшу обробку до модуля аналітики, де вже залучається мовна модель.

Лістинг 3.1 Приклад генерування BPMN-діаграми:

```
func (c *BpmnController) GenerateBpmn(ctx *gin.Context) {
    var req GenerateRequest
    if err := ctx.ShouldBindJSON(&req); err != nil {
        ctx.JSON(http.StatusBadRequest, gin.H{"error": "Invalid request"})
        return
    }
    result, err := c.service.GenerateProcess(req.Prompt)
    if err != nil {
        ctx.JSON(http.StatusInternalServerError, gin.H{"error": err.Error()})
        return
    }
    ctx.JSON(http.StatusOK, gin.H{"xml": result})
}>
```

### 3.2.2 Взаємодія з API мовної моделі

Компонент сервісу відповідає за формування запитів до мовної моделі та обробку отриманих результатів.

Він формує спеціальний системний `prompt`, який визначає правила побудови ВРМН [36, 37], та відправляє користувацький текст для аналізу. Мовна модель повертає список об'єктів із типами елементів і вказівкою, як вони повинні бути пов'язані між.

Отриманий результат проходить етап очищення, заміни спеціальних символів та перетворюється у придатний для використання формат.

Лістинг 3.2 Фрагмент взаємодії з API мовної моделі:

```
func (s *LLMService) GenerateElements(prompt string) ([]Element, error) {
    payload := map[string]string{"prompt": prompt}
    body, _ := json.Marshal(payload)
    resp, err := http.Post(LLM_API_URL, "application/json",
bytes.NewBuffer(body))
    if err != nil {
        return nil, err
    }
    defer resp.Body.Close()

    var result LlmResponse
    json.NewDecoder(resp.Body).Decode(&result)
    return result.Elements, nil
}
```

### 3.2.3 Взаємодія з MCP-сервер

MCP-сервер є ключовим компонентом архітектури, який відповідає за отримання, зберігання та повернення XML-шаблонів елементів BPMN. Він виступає проміжним шаром між мовною моделлю, яка визначає логічну структуру процесу, та модулем генерації, який формує фінальний XML-файл. Його роль полягає в тому, щоб гарантувати, що кожен елемент діаграми відповідає стандарту BPMN 2.0 і може бути виконаний у BPMS-середовищі, наприклад, Camunda 8.

Після того як мовна модель завершує аналіз тексту, вона формує структурований опис бізнес-процесу у вигляді списку елементів – подій, завдань, розгалужень, таймерів тощо. Кожен із цих елементів має свій тип (наприклад: початкова подія, користувацьке завдання, виключне розгалуження) та зв'язки, які визначають послідовність їх виконання. Щоб побудувати повноцінну BPMN-модель, система має отримати XML-фрагменти, що відповідають кожному типу елементів.

Саме тут виконується запит до MCP-сервер [38, 39], який виступає у ролі «джерела структурної істини». Він містить у собі репозиторій шаблонів, де кожен елемент описано у форматі XML. Для звернення до сервера мовна модель або проміжний шар системи надсилає запит у форматі JSON. У відповідь сервер повертає XML-шаблон, який відповідає запитаному типу елемента.

Отриманий фрагмент вставляється у спільну структуру моделі, яку система формує поступово – елемент за елементом. Кожен XML-файл має стандартну структуру з унікальним ID, атрибутами та зв'язками з іншими компонентами процесу.

На цьому етапі важливою особливістю є уніфікація шаблонів. MCP-сервер підтримує однакову структуру для всіх типів елементів, що дає змогу автоматично з'єднувати їх між собою без додаткового ручного коригування. Завдяки цьому система може легко комбінувати прості шаблони у складні BPMN-моделі.

Крім основного методу отримання елемента, MCR-сервер підтримує й інші види запитів. Зокрема, він може виконувати функцію `getFlow`, яка відповідає за генерацію зв'язків між елементами; функцію `getEvent`, що створює події з визначеними умовами або таймерами; а також функцію `getTemplateList`, яка надає перелік доступних типів елементів для подальшого формування моделі.

Ще однією перевагою MCR-сервер є його масштабованість. У міру розширення бази шаблонів він може зберігати альтернативні версії XML-фрагментів для різних BPMN-платформ. Це забезпечує сумісність між різними системами управління процесами та дозволяє гнучко адаптувати архітектуру під вимоги конкретного середовища.

#### 3.2.4 Робота з базою даних

Для забезпечення стабільності, масштабованості та швидкої обробки запитів у системі використовується база даних PostgreSQL [40], яка виконує роль сховища шаблонів BPMN-елементів, а також історії користувацьких запитів і результатів генерації діаграм. Такий підхід дає змогу зменшити навантаження на мовну модель, забезпечуючи повторне використання вже готових структур і швидкий доступ до необхідних елементів без додаткового запиту на аналіз тексту.

У структурі системи головною таблицею є `s8_bpmn_elements`, яка містить опис усіх стандартних елементів BPMN 2.0. Кожен запис у цій таблиці зберігає як метадані (тип, опис, атрибути), так і безпосередньо XML-шаблон, що використовується для складання фінальної BPMN-моделі.

Таблиця `s8_bpmn_elements` містить такі поля:

- поле `element_type`: тип елемента (наприклад, завдання, подія, розгалуження, послідовний перехід);
- поле `xml_snippet`: XML-фрагмент, що містить шаблон цього елемента для подальшого вставлення у модель;
- поле `description`: короткий текстовий опис призначення елемента;

– поле `attributes`: набір параметрів у форматі JSON, у якому можуть зберігатися додаткові характеристики.

Лістинг 3.3 SQL-запит для отримання BPMN-елемента:

```
SELECT xml_snippet  
FROM c8_bpmn_elements  
WHERE element_type = 'serviceTask'  
LIMIT 1;
```

Такий запит дозволяє швидко отримати XML-шаблон для потрібного елемента без залучення мовної моделі. Наприклад, якщо в процесі генерації мовна модель визначає, що у тексті користувача згадується дія на кшталт “система перевіряє дані заявки”, вона надсилає запит на MCP-сервер або безпосередньо до бази даних, щоб отримати шаблон для елемента типу користувацької задачі.

Завдяки цій структурі база даних виконує роль локального кешу знань BPMN, що значно знижує кількість повторних обчислень. У разі повторного звернення з аналогічним запитом система не викликає мовну модель повторно, а просто отримує збережений XML-фрагмент.

### 3.2.5 Узагальнення результатів реалізації серверної частини

Розроблена серверна частина системи є ключовим компонентом архітектури автоматичної генерації BPMN-діаграм. Вона забезпечує стабільність, масштабованість та узгодженість усіх етапів обробки – від моменту отримання текстового запиту користувача до формування готового XML-файлу BPMN.

Серверна частина реалізовано з використанням фреймворку Gin (Go) [41], що гарантує високу продуктивність і низький час відгуку навіть при одночасній

обробці великої кількості запитів. Його структура побудована за принципами REST API.

Ключовою перевагою реалізованого підходу є можливість інтеграції з різними мовними моделями – GPT, Gemini або Claude. Це стало можливим завдяки уніфікованому протоколу взаємодії, який використовує спільний формат запитів і відповідей у JSON. Таким чином, система не прив’язана до конкретного провайдера, що значно підвищує її гнучкість та життєздатність у довгостроковій перспективі.

Лістинг 3.4 Код відправлення запиту на аналіз текстової частини

```
func (b BpmnGenerator) Generate(ctx context.Context, prompt string) (string,
error) {
    var opts []ai.GenerateOption

    opts = append(opts,
        ai.WithPrompt(prompt),
        ai.WithSystem(SYSTEM_PROMPT),
        ai.WithTools(b.tools...),
        ai.WithMaxTurns(15),
    )

    resp, err := genkit.Generate(ctx, b.g, opts...)
    if err != nil {
        slog.Info("error generating bpmn:", "error", err)
        return "", err
    }

    slog.Info("generated bpmn:", "response", resp)
    result := b.cleanText(resp.Message.Content[0].Text)
    return result, nil
}
```

З точки зору масштабованості, сервер підтримує горизонтальне масштабування – можна запускати кілька інстансів для обробки запитів у паралельному режимі. Це особливо важливо у середовищах, де одночасно працюють десятки або сотні користувачів. Використання PostgreSQL як бази даних забезпечує надійне зберігання та швидкий доступ до шаблонів BPMN-елементів, а також до історії згенерованих моделей.

### 3.3 Реалізація клієнтської частини

Клієнтська частина системи була розроблена з метою забезпечення інтуїтивної взаємодії користувача з модулем генерації BPMN-діаграм. Вона реалізує повний цикл роботи – від введення опису бізнес-процесу у вигляді тексту до відображення сформованої BPMN-моделі у візуальному форматі.

Основу інтерфейсу складає React-компонент [42], який керує станом застосунку, відправляє запити до серверної частини та відображає отримані результати у вигляді діаграми.

#### 3.3.1 Опис інтерфейсу користувача

Користувацький інтерфейс (рис. 3.2) системи розроблено таким чином, щоб забезпечити максимально просту, інтуїтивно зрозумілу взаємодію користувача з функціоналом генерації BPMN-діаграм. Основна мета інтерфейсу – дати змогу користувачеві у зручний спосіб описати бізнес-процес природною мовою та миттєво отримати його графічне відображення у форматі BPMN.

Інтерфейс складається з трьох основних функціональних зон, які взаємопов'язані між собою та формують єдиний робочий простір.

Поле введення тексту це центральний елемент взаємодії користувача з системою. У ньому користувач формулює опис процесу природною мовою, без необхідності використовувати технічну термінологію чи знання BPMN.

Наприклад, користувач може ввести простий опис: «Клієнт оформлює заявку. Система перевіряє дані та надсилає підтвердження.» Після натискання кнопки надсилання система передає цей текст для обробки мовною моделлю, яка аналізує зміст, визначає ключові етапи, інтерпретує ролі учасників та встановлює логічні зв'язки між окремими діями. Мовна модель перетворює вихідний текст на структурований набір сутностей, які у подальшому використовуються для побудови BPMN-діаграми.

Кнопка надсилання запиту слугує тригером запуску всього процесу обробки введеної інформації. Після натискання відбувається виклик спеціально налаштованої функції, що формує запит до серверної частини через REST API. На бекенді текст проходить через модуль попередньої обробки, який очищує його від зайвих символів, виконує токенизацію та підготовку до подальшої інтерпретації. Далі сервер формує запит до мовної моделі, отримує структурований опис процесу та повертає його клієнтському застосунку.

Отримані результати можуть бути автоматично візуалізовані у вигляді BPMN-діаграми або представлені як набір кроків процесу, які користувач може переглянути, відредагувати та підтвердити. Таким чином, навіть короткий текстовий опис перетворюється на формальну модель бізнес-процесу завдяки взаємодії інтерфейсу, серверної логіки та системи обробки природної мови.

У момент натискання кнопка змінює свій стан: вона може ставати неактивною або показувати індикатор завантаження, що сигналізує про початок генерації діаграми. Фактично ця кнопка виконує роль не лише засобу надсилання даних, а й елемента керування процесом. Вона блокує повторні натискання, доки попередній запит не буде опрацьовано, після успішного створення BPMN-діаграми змінює свій статус на «Готово» або «Переглянути результат», а у разі виникнення помилки відображає відповідне повідомлення з поясненням та рекомендацією повторити спробу.

Describe the process below to generate BPMN

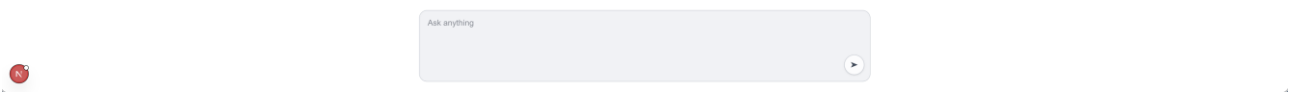


Рисунок 3.2 – Інтерфейс початкового екрану користувача

### 3.3.2 Механізм взаємодії з сервером

Взаємодія між клієнтською та серверною частинами реалізована за допомогою функції `fetch()`, що виконує HTTP POST-запит.

Користувацький запит містить JSON-об'єкт із ключем `prompt`, який передає опис процесу (рис. 3.3) мовною моделлю для подальшого аналізу.

Лістинг 3.6 Код відправлення запиту до серверної частини

```
const handleSubmit = async (e: React.FormEvent) => {
  e.preventDefault();
  setLoading(true);

  const res = await fetch("http://localhost:8081/bpmn/generate", {
    method: "POST",
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify({ prompt: description }),
  });
```

```

const data = await res.json();
setResult(data.bpmn_xml);
setLoading(false);
};

```

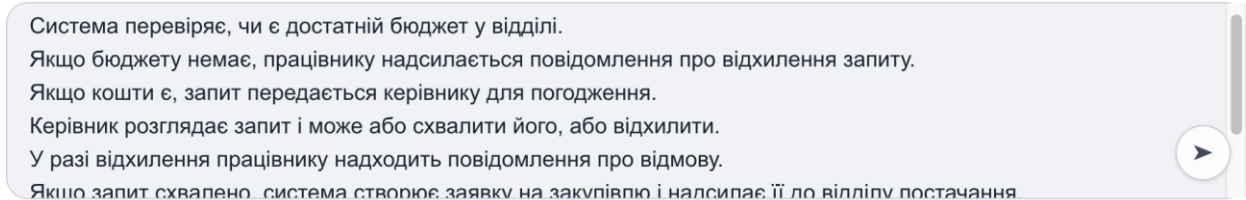


Рисунок 3.3 – Опис процесу у textarea для створення BPMN

### 3.4 Тестування та обґрунтування результатів

Етап тестування є критично важливою частиною розробки системи автоматичної генерації BPMN-діаграм [43], оскільки саме він дозволяє оцінити якість реалізованих алгоритмів, стабільність роботи архітектури та точність перетворення природномовних описів у формалізовані BPMN-моделі.

Метою тестування було підтвердити працездатність створеного рішення, визначити ступінь відповідності згенерованих діаграм стандарту BPMN 2.0, а також виявити межі ефективності застосованих методів у різних сценаріях використання.

#### 3.4.1 Методика тестування

Тестування проводилося у кілька етапів, що охоплюють усі основні складові системи: від роботи з мовною моделлю до формування готового XML-файлу.

Основна методика базувалася на принципах функціонального тестування, валідації результатів та оцінки продуктивності. Для забезпечення

відтворюваності результатів усі тести виконувались у контрольованому середовищі з фіксованими параметрами системи.

У межах дослідження було визначено кілька ключових напрямків перевірки системи. Насамперед оцінювалася коректність синтаксису BPMN XML, що передбачало перевірку відповідності згенерованих діаграм стандарту BPMN 2.0 та можливість їх безпомилкового відкриття у Camunda Modeler. Далі аналізувалася семантична відповідність, тобто правильність логічної побудови процесу, включно з коректними зв'язками між подіями, шлюзами та завданнями. Окрему увагу приділяли швидкодії системи, вимірюючи час відповіді сервера за різної довжини текстових описів. Також перевірялася стійкість і масштабованість, що включало тестування роботи під час одночасного надходження великої кількості запитів. Окрім цього, проводилося тестування взаємодії з MCR-сервером і базою даних для оцінки швидкості доступу до шаблонів BPMN та стабільності системи під час багатопотокових звернень.

### 3.4.2 Тестові сценарії

Для перевірки функціональності було розроблено чотири тестові сценарії, які охоплювали як прості, так і складні бізнес-процеси.

У процесі тестування системи автоматичної генерації BPMN-діаграм було проведено серію експериментів, спрямованих на оцінку здатності моделі правильно інтерпретувати текстові описи бізнес-процесів різного рівня складності та перетворювати їх у формалізовані BPMN-моделі. Кожен сценарій був побудований таким чином, щоб максимально відтворити реальні умови використання системи та продемонструвати її ефективність у різних ситуаціях: від простого лінійного опису до розгалужених процесів із кількома умовами та альтернативними шляхами виконання.

Перший сценарій відтворював найпростіший тип бізнес-процесу – лінійну послідовність кроків. У ньому користувач створює заявку, система перевіряє введені дані, після чого формується та надсилається підтвердження. Основною

метою сценарію було перевірити здатність моделі правильно інтерпретувати базову черговість дій, визначити типи завдань (user task, service task, send task) та сформувати коректні послідовні зв'язки між ними. За результатами тестування модель складалася з трьох ключових елементів, об'єднаних потоками послідовності, і була згенерована без помилок. Отримана діаграма (рис. 3.4) повністю відповідала очікуваному результату, що підтвердило точність роботи системи у випадку найпростішого варіанту бізнес-опису та продемонструвало її здатність коректно формувати базові BPMN-моделі.

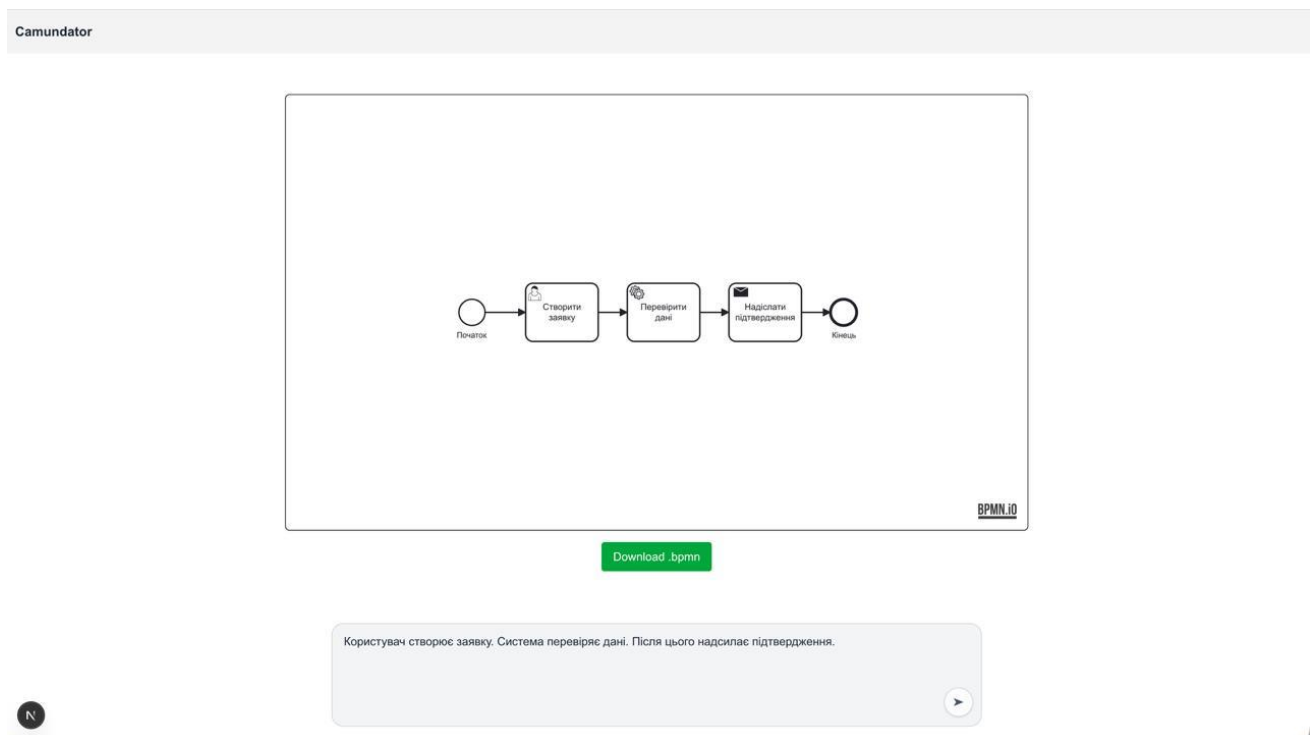


Рисунок 3.4 – Результат генерації простого послідовного процесу

Другий сценарій був спрямований на перевірку здатності системи працювати з процесами, у яких окремі дії виконуються паралельно. Текстовий опис передбачав, що після перевірки даних система має виконати два незалежні завдання: надіслати повідомлення клієнту та зберегти заявку у базі даних. Такі процеси є поширеними в реальних бізнесових сценаріях, оскільки дозволяють пришвидшити обробку інформації та уникнути зайвих затримок. Тому важливо було оцінити, чи здатна мовна модель правильно інтерпретувати логіку паралельного виконання й вибрати відповідний BPMN-елемент – паралельний шлюз.

Під час тестування перевірялося не лише виявлення паралельної структури, а й коректність побудови обох гілок, синхронізація потоків та дотримання стандартів BPMN 2.0. Отримана діаграма (рис. 3.5) містила два окремі потоки, які запускалися одночасно та завершувалися у відповідному місці злиття, що забезпечувало логічну узгодженість процесу. Такий результат засвідчив, що модель здатна правильно інтерпретувати незалежні дії, визначати моменти розгалуження та об'єднання, а також формувати коректні паралельні структури навіть у випадках, коли відповідна логіка явно не вказана у тексті.

Таким чином, другий сценарій продемонстрував, що система може обробляти не лише послідовні, а й складніші процеси з паралельними гілками, що є критично важливим для побудови реалістичних BPMN-моделей у практичних бізнесових задачах.

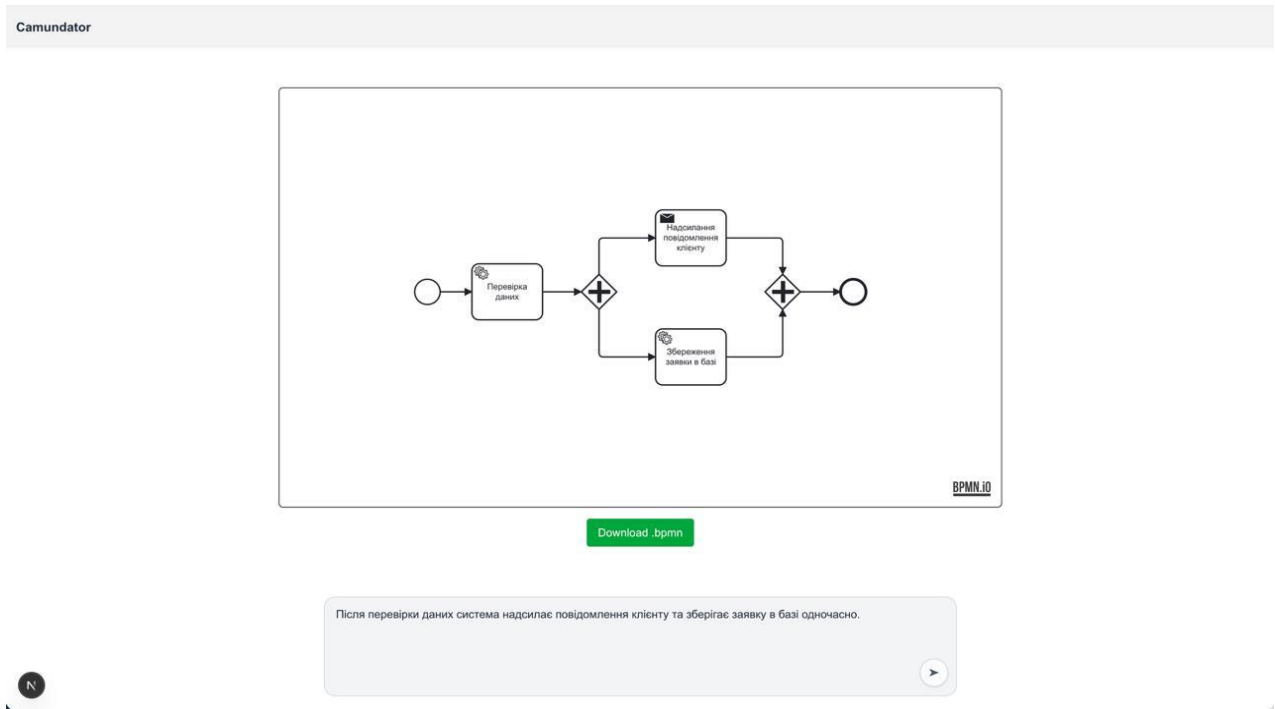


Рисунок 3.5 – Процес із паралельними потоками виконання

У третьому сценарії було змодельовано процес із таймером, у якому після перевірки заявки система повинна почекати визначений проміжок часу, а вже потім надіслати повідомлення клієнту. Такий тип процесів поширений у реальних бізнесових застосуваннях, де важливо враховувати часові затримки, нагадування або відкладені дії. Основною метою було перевірити, чи зможе мовна модель коректно інтерпретувати часову логіку та сформувати елемент, що

відповідає проміжній події очікування з таймером. У результаті система успішно згенерувала BPMN-діаграму (рис. 3.6) із вбудованим таймером, що підтвердило її здатність автоматично створювати процеси, у яких присутні часові умови та відкладені дії.

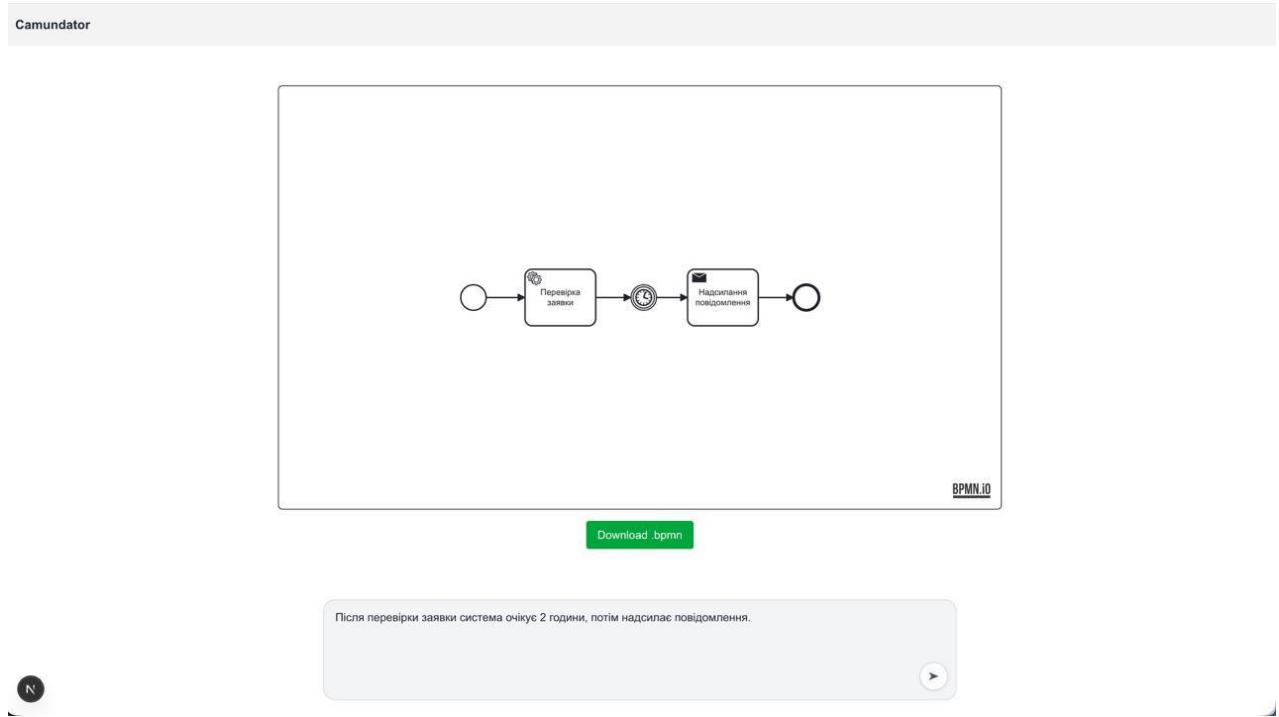
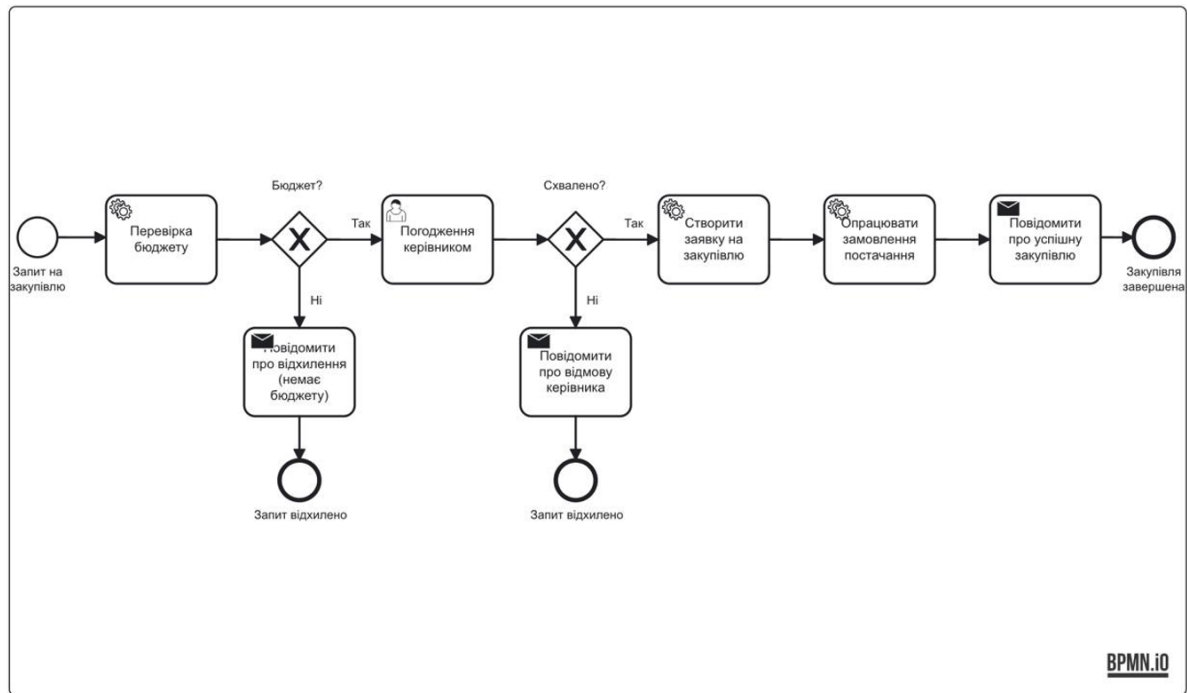


Рисунок 3.6 – Процес із таймером

Четвертий сценарій моделював складний бізнес-процес із декількома виключеннями. У цьому випадку система мала розпізнати ситуації відмови оплати, недоступності товару або затримки доставки. Залежно від умови, процес переходив до відповідної гілки – надсилання повідомлення клієнту, повторної обробки замовлення чи ескалації менеджеру. Створена діаграма (рис. 3.7) містила кілька типів подій та відображала реалістичну логіку корпоративного процесу з винятковими ситуаціями. Приклад запиту: «Система перевіряє, чи є достатній бюджет у відділі. Якщо бюджету немає, працівнику надсилається повідомлення про відхилення запиту. Якщо кошти є, запит передається керівнику для погодження. Керівник розглядає запит і може або схвалити його, або відхилити. У разі відхилення працівнику надходить повідомлення про відмову. Якщо запит схвалено, система створює заявку на закупівлю і надсилає її до відділу постачання. Відправляє повідомлення про успішну закупівлю». Результати описані у таблиці 3.1.



Download .bpmn

Рисунок 3.7 – Складний бізнес-процес із декількома виключеннями

Для кожного сценарію перевірялись такі параметри:

- час генерації BPMN-моделі;
- коректність структури XML;
- наявність усіх очікуваних елементів;
- відсутність помилок при відкритті в Camunda Modeler.

Таблиця 3.1 – Результати тестування функціональних сценаріїв

№	Тип процесу	Кількість елементів	Час генерації (с)	Результат
1	Лінійний	5	4	Відмінно
2	3 розгалуженням	8	6	Відмінно
3	3 таймером	10	15	Відмінно
4	Складний бізнес-процес	9	17	Відмінно

### 3.4.3 Перевірка помилок

Окремою частиною тестування є валідація згенерованої BPMN-діаграми в Camunda Modeler.

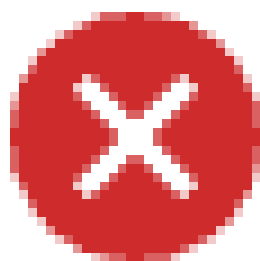
Після генерації XML-моделі її імпортують у Camunda Modeler, який автоматично виконує синтаксичну перевірку відповідності нотації BPMN 2.0 і виявляє структурні помилки.

Camunda Modeler підтримує два основних типи повідомлень про проблеми:

- критичні помилки, позначені червоним маркером на рисунку 3.8 а). Такі помилки роблять модель недійсною та унеможливають виконання або розгортання процесу в Camunda. Наприклад: не зазначено тип події, некоректні посилання у переходах, повторювані ідентифікатори елементів.;

- попередження, що позначаються жовтим маркером на рисунку 3.8 б), вони не блокують виконання, але вказують на потенційні проблеми якості моделі, які бажано виправити (наприклад, відсутність назви задачі).

Таким чином, перевірка у Camunda Modeler дає змогу впевнитися, що згенерована мовною моделлю BPMN-структура є не лише логічно правильною, але й відповідає формальним вимогам стандарту, що робить її придатною для виконання.



(а)



(б)

Рисунок 3.8 – Типи помилок BPMN 2.0:

(а) критична помилка; (б) зауваження.

До основних типів помилок, які перевіряються під час тестування, належать:

- відсутність обов’язкових атрибутів елементів;
- некоректні або відсутні входи/виходи елементів;
- помилки типу «звисяючі потоки» – коли потік не з’єднано з елементом;
- некоректне використання шлюзів;
- дублікати ID;
- невідповідність XML схемі BPMN 2.0.

Camunda Modeler не повинен показувати критичних помилок у вкладці «Problems». Усі згенеровані елементи мають бути валідними і правильно відображатися у візуальному редакторі (рис. 3.9).

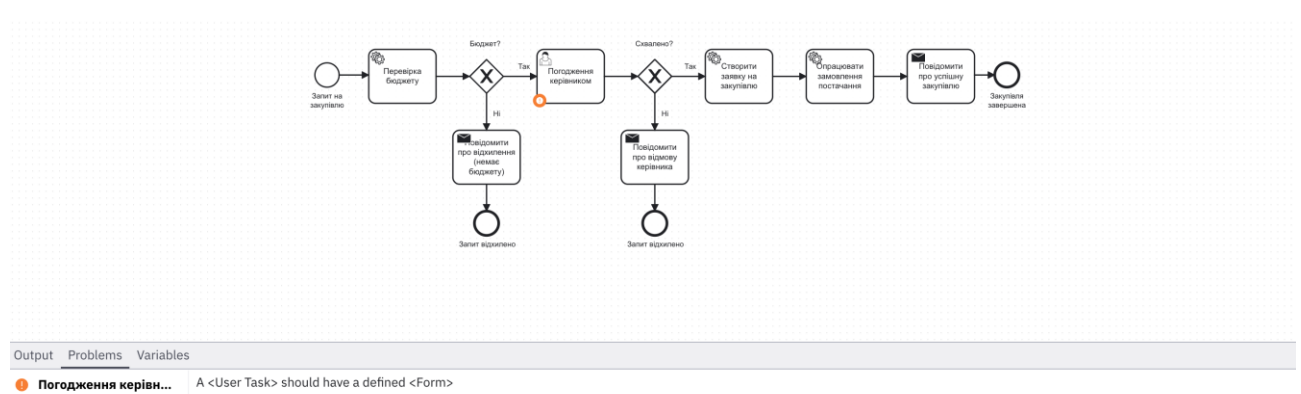


Рисунок 3.9 – Демонстрація відсутності критичних помилок згенерованої BPMN

#### 3.4.4 Порівняння мовних моделей

У межах дослідження було проведено експериментальне порівняння кількох сучасних мовних моделей, здатних виконувати трансформацію природномовного опису процесу у формальну BPMN-модель. Метою порівняння було визначити, яка з моделей найточніше відтворює структуру

бізнес-процесу, найменше допускає семантичних і структурних помилок, та забезпечує синтаксичну коректність XML-діаграми.

Для цього був сформований тестовий датасет, який складався з трьох основних компонентів: промпту користувача, що містив текстовий опис бізнес-процесу; еталонної BPMN-діаграми, вручну побудованої автором як точна модель процесу; а також згенерованої діаграми, отриманої як результат роботи системи на основі заданого запиту.

Такий підхід дозволив оцінити не лише правильність структурного складу діаграми, але й ступінь наближеності згенерованої моделі до реального процесу.

Було сформовано 7 тестових сценаріїв (табл. 3.2), які охоплюють різні категорії бізнес-процесів:

- прості лінійні процеси;
- процеси з умовними розгалуженнями;
- процеси з паралельним виконанням;
- процеси з таймерами;
- процеси з обробкою помилок і відмов;
- процеси зі структурами повідомлень.

Таблиця 3.2 – Сформовані запити для дослідження

Категорія бізнес-процеса	Запит
1	2
Простий лінійний	Процес починається з отримання заявки. Після цього система перевіряє передані дані, визначає їх коректність та повноту. Якщо перевірка проходить успішно, заявка переходить на етап затвердження відповідальним співробітником. У разі потреби її можуть повернути на уточнення або доопрацювання.

Продовження таблиці 3.2

1	2
Паралельне виконання	Після перевірки даних система одночасно надсилає повідомлення клієнту та зберігає заявку в базі. Після завершення обох дій процес завершується.
Процес з таймером	Після надсилання рахунку, система чекає 7 днів на оплату. Якщо оплата не надійшла, процес переходить до відправлення нагадування.
Обробка помилок	При обробці платежу може виникнути помилка. У разі помилки процес має перейти до ручної перевірки.
Обробка відмов	При поданні заявки є 5 днів на підтвердження документів. Якщо час вичерпано, заявка автоматично скасовується.
Структури повідомлень	Система надсилає запит на інформацію до зовнішнього сервісу і чекає на відповідь перед продовженням.

Проведене дослідження мало на меті порівняльний аналіз чотирьох ключових великих мовних моделей – Gemini 2.5 Pro, GPT-4.1, Gemini 2.5 Flash та GPT-4.1-Mini – щодо їхньої здатності генерувати коректні та семантично відповідні діаграми бізнес-процесів у нотації BPMN на основі текстових описів. Аналіз базувався на 7 тестових сценаріях, які охоплювали повний спектр складності BPMN-конструкцій (лінійні процеси, умовні та паралельні шлюзи, обробка подій та повідомлень).

Оцінювання ефективності моделей проводилося за трьома основними критеріями:

- структурна коректність: відсоток кейсів, де згенерована діаграма відповідає синтаксису BPMN без критичних помилок;
- семантична відповідність: кількісний показник, що відображає точність відтворення логіки, описаної у запиті;
- схожість зі зразком: кількісний показник, що відображає ступінь візуальної та логічної близькості до еталонної діаграми.

Результати порівняння представлені у таблиці 3.3.

Таблиця 3.3 – Порівняння моделей за запитами

<b>Модель</b>	<b>Структурна коректність</b>	<b>Семантична відповідність</b>	<b>Схожість зі зразком</b>	<b>Загальний бал</b>
Gemini 2.5 Pro	100%	0.95	0.93	9.6 / 10
GPT-4.1	100%	0.92	0.91	9.4 / 10
Gemini 2.5 Flash	95%	0.88	0.85	8.9 / 10
GPT-4.1-Mini	83%	0.74	0.78	7.8 / 10

Аналіз результатів генерації BPMN-діаграм, отриманих від моделей Gemini 2.5 Pro, GPT-4.1, Gemini 2.5 Flash та GPT-4.1-Mini, підтвердив пряму залежність ефективності від архітектурної потужності, але виявив критичну перевагу моделей Gemini в контексті семантичної точності та економічної доцільності.

Дві флагманські моделі, Gemini 2.5 Pro та GPT-4.1, продемонстрували найвищу якість, досягнувши 100% структурної коректності, що є необхідною умовою для подальшої автоматизації та виконання процесів. Проте, за показником семантичної відповідності (точного відтворення бізнес-логіки), Gemini 2.5 Pro виявив найглибше розуміння складних вимог, перевершивши GPT-4.1. Ця перевага вказує на здатність Gemini 2.5 Pro точніше інтерпретувати нюанси взаємодії елементів, особливо у сценаріях з обробкою помилок та

паралельним виконанням, що визначає його як найбільш точний інструмент для генерації критично важливих BPMN-діаграм.

Аналіз моделей, оптимізованих за вартістю та швидкістю, виявив значну економічну перевагу Gemini 2.5 Flash. Ця модель досягла структурної коректності 95% та семантичної відповідності 0.88, що є відмінним результатом для економічного сегменту. Зокрема, Gemini 2.5 Flash значно перевищила показники GPT-4.1-Mini, доводячи, що вона успішно справляється з переважною більшістю бізнес-сценаріїв. Показник 95% структурної коректності свідчить про високу надійність, тоді як його економічна ефективність робить Gemini 2.5 Flash найкращим вибором з точки зору цінності.

У проєктах, орієнтованих на масштабування та бюджетну ефективність, де невеликий відсоток помилок може бути виправлений ручною валідацією, Gemini 2.5 Flash є технічно та економічно обґрунтованою рекомендацією, на відміну від моделей меншого розміру, наприклад GPT-4.1-Mini, які несуть значний ризик критичних структурних помилок. Таким чином, для завдань, що вимагають стовідсоткової точності, слід обирати Gemini 2.5 Pro, тоді як Gemini 2.5 Flash є оптимальним рішенням для масового моделювання при контролі витрат.

## ВИСНОВКИ

Таким чином, у кваліфікаційній роботі досліджено підходи до побудови бізнес-процесів у нотації BPMN із використанням інтелектуальних агентів та мовних моделей, а також розроблено архітектуру системи, здатної автоматично генерувати BPMN-діаграми на основі текстового опису процесу природною мовою.

У ході дослідження вирішено такі завдання:

- проведено аналіз сучасних підходів до моделювання бізнес-процесів, визначено їхні переваги та обмеження, а також охарактеризовано роль BPMN як стандарту для уніфікованого відображення бізнес-логіки;

- вивчено методи обробки природної мови та особливості застосування великих мовних моделей для семантичного аналізу текстів у контексті побудови процесів;

- розроблено алгоритм перетворення текстових описів у BPMN-модель, що включає етапи токенізації, класифікації ролей, формування потоків і побудови XML-структури;

- спроектовано архітектуру системи автоматичної генерації BPMN-діаграм, яка складається з трьох рівнів: інтерфейсного, аналітичного та системного (MCP-сервер, Camunda);

- реалізовано серверну частину для обробки запитів, взаємодії з мовною моделлю та побудови BPMN XML, а також клієнтську частину для введення тексту, візуалізації результатів і завантаження згенерованих діаграм;

- проведено тестування системи на чотирьох типах сценаріїв – лінійних, паралельних, умовних і часових, що підтвердило коректність побудови BPMN, синтаксичну валідність XML і точність інтерпретації текстових інструкцій.

Результатом роботи стала повнофункціональна система автоматичної генерації BPMN-діаграм, яка забезпечує перетворення текстових описів у формальні моделі процесів, сумісні з Camunda Modeler. Це дозволяє значно скоротити час проєктування бізнес-процесів, зменшити кількість ручних дій аналітиків і підвищити точність документування процесної логіки.

Наукова новизна проведеної роботи полягає у створенні архітектури, що поєднує мовну модель, MCP-сервер та BPMN-середовище в єдину інтегровану систему, здатну виконувати когнітивний аналіз тексту та формувати структуровані XML-моделі. Практична цінність полягає у можливості застосування розробленої системи для автоматизації побудови бізнес-процесів у корпоративних BPMS-середовищах, зокрема Camunda 8, що відкриває нові перспективи для розвитку інтелектуальних систем процесного моделювання та цифрової трансформації підприємств.

На основі порівняльного аналізу ефективності мовних моделей, визначено, що вибір моделі для інтегрованої системи повинен ґрунтуватися на пріоритетах проекту: для завдань, що вимагають абсолютної точності у генерації складних BPMN-діаграм, слід обирати модель Gemini 2.5 Pro. Вона досягла найвищого загального балу, забезпечуючи ідеальну структурну коректність та найкращу семантичну відповідність, що є критично важливим для процесів, які безпосередньо виконуються в BPMS. Натомість, для проектів, орієнтованих на масштабування та бюджетну ефективність, рекомендовано обирати модель Gemini 2.5 Flash. Ця модель забезпечує високу якість при значно менших операційних витратах, що робить її оптимальним економічно обґрунтованим рішенням для масового моделювання, тоді як моделі з нижчою точністю, як-от GPT-4.1-Mini, не можуть бути рекомендовані через високий ризик критичних структурних помилок.

Результати роботи апробовано у вигляді тез доповідей, представлених на VIII Міжнародній студентській науковій конференції «Тренди та перспективи розвитку мультидисциплінарних досліджень» [44] та на X Міжнародній науково-практичній конференції «Current Issues, Modern Achievements and Innovations of Science» [45].

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Творошенко, І.С. (2021). Технології прийняття рішень в інформаційних системах: навч. посібник. Харків: ХНУРЕ. С. 17-20.
2. Yakovleva, O., Kovtunenکو, A., Liubchenko, V., Honcharenko, V., & Kobylin, O. (2023, April). Face Detection for Video Surveillance-based Security System. In COLINS (3) (pp. 69-86).
3. Кобилін, О.А., & Творошенко, І.С. (2021). Методи цифрової обробки зображень: навч. посібник. Харків: ХНУРЕ.
4. Гороховатський В.О., Творошенко І.С. (2022) Аналіз багатовимірних даних за описом у формі множини компонент: монографія. Харків: ХНУРЕ, 124 с.
5. Lyashenko, V., Lyubchenko, V., Mohammad, A., Alveera, K., & Kobylin, O. (2016). The methodology of image processing in the study of the properties of fiber as a reinforcing agent in polymer compositions. С.15-20.
6. Mustafa, S. K., Kopot, M., Ahmad, M. A., Lyubchenko, V., & Lyashenko, V. (2020). Interesting applications of mobile robotic motion by using control algorithms, С. 5-12.
7. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Al-Dhaifallah M. (2022) Classification of Images Based on a System of Hierarchical Features, *Computers, Materials & Continua*, 72(1), pp. 1785-1797.
8. Гороховатський В., Передрій О., Творошенко І., Марков Т. (2023) Матриця відстаней для множини компонентів структурного опису як інструмент для створення класифікатора зображень, *Сучасні інформаційні системи*, 7(1), С. 5-13.
9. Gorokhovatskyi, V., Tvoroshenko, I., Kobylin, O., & Vlasenko, N. (2023). Search for visual objects by request in the form of a cluster representation for the structural image description, *Advances in Electrical and Electronic Engineering*, 21(1), pp. 19-27.

10. Li, Z., Yang, Y., & Wang, J. (2022). Automatic Process Model Generation Using Large Language Models. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(5), 1234–1246.
11. Pomazan, V., Tvoroshenko, I., & Gorokhovatskyi, V. (2023). Development of an application for recognizing emotions using convolutional neural networks, *International Journal of Academic Information Systems Research*, 7(7).
12. Tvoroshenko I., and Gorokhovatskyi V. (2022) The Application of Hybrid Intelligence Systems for Dynamic Data Analysis, *International Journal of Engineering and Information Systems*, 6(2), pp. 40-48.
13. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., Gadetska S., and Al-Dhaifallah M. (2023) Statistical data analysis models for determining the relevance of structural image descriptions, *IEEE Access*, vol. 11.
14. Gorokhovatskyi V., Tvoroshenko I., and Yakovleva O. (2024) Transforming image descriptions as a set of descriptors to construct classification features, *Indonesian Journal of Electrical Engineering and Computer Science*, 33(1), pp. 113-125.
15. Lyubchenko, V., Veretelnyk, K., Kots, P., & Lyashenko, V. (2024). Digital image segmentation procedure as an example of an NP-problem.
16. Gorokhovatskyi, V., Tvoroshenko, I., Yakovleva, O., Hudáková, M., & Gorokhovatskyi, O. (2024). Application a committee of Kohonen neural networks to training of image classifier based on description of descriptors set. *IEEE Access*, 12, 73376-73385.
17. Gorokhovatskyi, V., Chmutov, Y., Tvoroshenko, I., & Kobylín, O. (2025). Reducing computational costs by compressing the structural description in image classification methods. *Advanced Information Systems*, 9(1), 5-12.
18. Daradkeh, Y. I., Gorokhovatskyi, V., Tvoroshenko, I., & Zeghid, M. (2024). Improving the Effectiveness of Image Classification Structural Methods by Compressing the Description According to the Information Content Criterion. *Computers, Materials & Continua*, 80(2).
19. Pomazan V., Tvoroshenko I., and Gorokhovatskyi V. (2023) Handwritten character recognition models based on convolutional neural networks, *International Journal of Academic Engineering Research*, 7(9).

20. Tvoroshenko I., Gorokhovatskyi V., Kobylin O., and Tvoroshenko A. (2023) Application of deep learning methods for recognizing and classifying culinary dishes in images, *International Journal of Academic and Applied Research*, 7(9), pp. 57-70.

21. Yakovleva, O., Matúšová, S., Tvoroshenko, I., & Isaiev, Y. (2024). Visitor counting based on video stream analysis from surveillance cameras to solve various business problems. *Verejná správa a regionálny rozvoj ekonómia, manažment a marketing*, XX (1), 67-87.

22. Yakovleva, O., Matúšová, S., Liubchenko, V., Maksimov, H. (2025). Innovative solutions based on AI in education, on the example of generating multimodal lecture notes. *Scientific Journal of Bratislava University of Economics and Management «Public Administration and Regional Development, Economics, Management and Marketing»*, vol. 21, no. 1, pp.140–163.

23. Yakovleva, O., Matúšová, S., Talakh, V. (2025, February 14). Gradio and Hugging capabilities for developing research AI applications. *Proceedings of the VII International Scientific and Practical Conference «Scientific practice: modern and classical research methods»*, Boston, USA.

24. Yakovleva, O., Matúšová, S., Koshel, V. (2025, February 21). Implementation of AI approaches in current tools for managing image collections to improve the search capabilities. *Proceedings of the IV Correspondence International Scientific and Practical Conference «Science in motion: classic and modern tools and methods in scientific investigations»* in *Periodical International scientific journal «Grail of science»*. Vinnytsia, Ukraine - Vienna, Austria. Vol. 49.

25. Cherednichenko, O., Yanholenko, O., Iakovleva, O., & Kustov, O. (2014). Models of Research Activity Measurement: Web-based monitoring implementation. *Information Systems: Education, Applications, Research*, 75–87.

26. Cherednichenko, O., Yanholenko, O., Liutenko, I., & Iakovleva, O. (2013). Monitoring and Evaluation Problems in Higher Education Comprehensive Assessment Framework Development. In *CSEU* (pp. 455-460).

27. Yanholenko, O., Cherednichenko, O., Yakovleva, O., Arkatov, D. (2020). A Model for Estimating the Security Level of Mobile Applications: a Fuzzy Logic Approach. In *IntelITSIS*.

28. Klyuchka, Y. A., Cherednichenko, O. Y., Vasylenko, A. V., & Yakovleva, O. V. (2017). Forecasting the results of football matches on the internet based information. Bulletin of National Technical University "KhPI". Series: System Analysis, Control and Information Technologies, 0(55), 51–59.

29. Gorokhovatskyi V., Tvoroshenko I. (2023) Identification of visual objects by the search request. International scientific symposium «INTELLIGENT SOLUTIONS-S». Computational intelligence (results, problems and perspectives). Decision making theory: proceedings of the international symposium, September 28, 2023, Kyiv-Uzhorod, Ukraine.

30. Silver, B. (2021). BPMN Method and Style: A Levels-Based Methodology for BPM Process Modeling and Improvement (3rd ed.). Cody-Cassidy Press.

31. Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. A. (2018). Fundamentals of Business Process Management (2nd ed.). Springer.

32. Camunda. (2024). Camunda Platform 8 Documentation. URL: <https://docs.camunda.io/> (Дата звернення 02.11.2025).

33. Marrella, A., Mecella, M., & Russo, A. (2022). Cognitive BPM: Combining AI and BPM for Adaptive Process Management. Information Systems, 104, 101919. <https://doi.org/10.1016/j.is.2021.101919>

34. Google Cloud. (2024). Vertex AI and Generative AI Models Overview. Retrieved from <https://cloud.google.com/vertex-ai> (Дата звернення 02.11.2025).

35. OpenAI. (2024). GPT-4 Technical Report. Retrieved from <https://openai.com/research/gpt-4> (Дата звернення 02.11.2025).

36. Ghidini, C., et al. (2020). Intelligent Process Automation: Combining Artificial Intelligence and Process Management. Journal of Artificial Intelligence Research, 68, 875–921.

37. Dalle Carbonare, D., Dragoni, A. F., & Marrella, A. (2021). Intelligent Agents for Process Automation: A Review. Journal of Systems and Software, 177, 110953.

38. Mendling, J., Pentland, B. T., & Recker, J. (2020). Building a Complementary Agenda for Business Process Management and Digital Innovation. European Journal of Information Systems, 29(3), 208–219.

39. IBM. (2023). Intelligent Automation: The Next Evolution of Business Process Management. IBM Corporation.
40. Fernandez, H., & Reijers, H. A. (2021). Natural Language Processing in BPM: A Systematic Literature Review. *Business & Information Systems Engineering*, 63(3), 241–259.
41. PostgreSQL Global Development Group. (2024). PostgreSQL 16 Documentation. Retrieved from <https://www.postgresql.org/docs/> (Дата звернення 02.11.2025).
42. GitHub. (2024). bpmn-js Library by bpmn.io. URL: <https://github.com/bpmn-io/bpmn-js> (Дата звернення 02.11.2025).
43. Golang Foundation. (2024). The Go Programming Language Specification. URL: <https://go.dev/doc/> (Дата звернення 02.11.2025).
44. Ardagna, D., Pernici, B., & Zengin, A. (2020). Adaptive and Cognitive Business Process Management in the Era of AI. *ACM Computing Surveys*, 53(6), C. 1–36.
45. Іткін Д. О., Любченко В. А. (2025). Інтелектуальне моделювання бізнес-процесів у BPMN. У *Proceedings of the X International Scientific and Practical Conference «Current Issues, Modern Achievements and Innovations of Science»*. С. 46-47.
46. Іткін Д. О., Любченко В. А. (2025). Моделювання бізнес-процесів у BPMN з використанням інтелектуальних агентів. «Тренди та перспективи розвитку мультидисциплінарних досліджень». С. 87-88.