

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Інформаційних управляючих систем
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів Process Mining в задачах побудови і удосконалення бізнес процесів волонтерської організації
(тема)

Виконав:
студент 2 курсу, групи ІУСТм-21-1
Артем ГРИБЕНЮК
(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

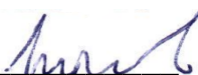
Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Інформаційні управляючі системи та технології
(повна назва освітньої програми)

Керівник професор каф. ІУС Сергій ЧАЛИЙ
(посада, власне ім'я, прізвище)

Допускається до захисту

Зав. кафедри


(підпис)

Костянтин ПЕТРОВ
(власне ім'я, прізвище)

2022 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Інформаційних управляючих систем

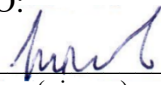
Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

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

Освітня програма Інформаційні управляючі системи та технології
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри 
(підпис)

« 21 » листопада 20 22 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Грибенюку Артему Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів Process Mining в задачах побудови і удосконалення бізнес процесів волонтерської організації
затверджена наказом університету від 14 листопада 2022 р. № 1490Ст
2. Термін подання студентом роботи до екзаменаційної комісії 14 12 2022 р.
3. Вихідні дані до роботи технології інтелектуального аналізу, технології управління процесам, методи аналізу та вдосконалення процесів, сучасні рішення програмної реалізації алгоритмів.
4. Перелік питань, що потрібно опрацювати в роботі виконати обстеження актуального стану волонтерської організації, провести аналіз та опис бізнес процесів організації, проаналізувати та порівняти методи аналізу та вдосконалення процесів, описати алгоритм Alpha-Miner методу, удосконалити метод Alpha-Miner для функціонування в інформаційній системі волонтерської організації, зробити опис технічних засобів і рішень для інформаційної системи, щоб реалізувати метод, експериментально перевірити удосконалений метод.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз задач управління процесами	10.10.2022	Виконано
2	Аналіз літератури та готових рішень інтелектуального аналізу процесів	11.10.2022 – 25.10.2022	Виконано
3	Постановка задачі	26.10.2022 – 3.11.2022	Виконано
4	Аналіз бізнес процесів волонтерської організації	4.11.2022 – 10.11.2022	Виконано
5	Дослідження методів аналізу та удосконалення процесів	11.11.2022 – 16.11.2022	Виконано
6	Реалізація удосконаленого Alpha-miner методу	17.11.2022 – 23.11.2022	Виконано
7	Експериментальна перевірка удосконаленого Alpha-miner методу	24.11.2022 – 27.11.2022	Виконано
8	Оформлення пояснювальної записки	28.11.2022 – 01.12.2022	Виконано
9	Оформлення графічної частини та презентаційних матеріалів захисту	02.12.2022 – 04.12.2022	Виконано
10	Представлення на рецензування	05.12.2022	Виконано
11	Представлення дипломного проекту в ЕК	14.12.2022	Виконано

Дата видачі завдання 21 листопада 2022 р.

Студент _____


(підпис)

Керівник роботи _____


(підпис)

професор каф. ІУС, Сергій ЧАЛИЙ
(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи містить: 85 с., 4 розділи, 35 рис., 3 табл., 35 джерел.

АНАЛІЗ, БІЗНЕС-ПРОЦЕСИ, ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ, ОРГАНІЗАЦІЯ, УПРАВЛІННЯ ПРОЦЕСАМИ, СХЕМА, PROCESS-MINING

У кваліфікаційній роботі виконано дослідження існуючих методів Process Mining в задачах побудови і удосконалення бізнес процесів волонтерської організації. У результаті проведеного аналізу було обрано, удосконалено та налагоджено конкретний алгоритм для розробки схематичної моделі процесів організації, що можуть бути використані при удосконаленні, або розширенні діяльності організації.

Об'єктом дослідження в рамках магістерської кваліфікаційної роботи є бізнес процеси волонтерської організації.

Предметом дослідження є методи удосконалення бізнес процесів з використанням Process Mining.

Результати роботи:

- обстеження актуального стану волонтерської організації;
- аналіз та опис бізнес процесів організації;
- аналіз та порівняння методів інтелектуального аналізу Process Mining;
- опис алгоритму методу Alpha-Miner;
- удосконалення да налагодження Alpha-Miner методу для функціонування в інформаційній системі волонтерської організації;
- опис технічних засобів і рішень для інформаційної системи, щоб реалізувати метод;
- експериментальна перевірка удосконаленого методу.

ABSTRACT

The explanatory note to the qualification work contains: 85 pages, 4 sections, 35 figures, 3 tables, 35 sources.

ANALYSIS, BUSINESS-PROCESSES, INTELLIGENT ANALYSIS, OPTIMIZATION, ORGANIZATION, PROCESS MANAGEMENT, PROCESS-MINING, SCHEMA

The study is about an overview of existing Process Mining methods in the building, analyzing and improving business processes of the chosen volunteer organization. Specific algorithm was selected and adjusted for building a model of the organization's processes. It can be used to improve or expand the organization's activities.

The objects of research of the master's qualification study are business processes of volunteer organization.

The subject of research is about methods of improving business processes using Process Mining.

Work results:

- examination of the current structure and state of the volunteer organization;
- description of business processes of the organization;
- analysis and comparison of process analytics methods – Process Mining;
- analysis of the process analytics method algorithm - Alpha-Miner;
- upgrade and setting up the Alpha-Miner method for functioning in the information system of a volunteer organization;
- description of the technical means and solutions for the information system to implement the method;
- experimental verification of the improved method.

ЗМІСТ

Календарний план	3
Зміст	6
Скорочення та умовні позначки	8
Вступ.....	9
1 Аналіз предметної області.....	11
1.1 Дослідження бізнес процесів волонтерської організації.....	11
1.2 Аналіз задач процесного управління.....	17
1.3 Дослідження методів інтелектуального аналізу процесів.....	19
1.4 Постановка задачі дослідження	25
2 Аналіз та удосконалення методу інтелектуального аналізу alpha-miner	27
2.1 Побудова моделі бізнес-процесу з використанням методу інтелектуального аналізу Alpha-miner	27
2.2 Удосконалення методу Alpha-miner для функціонування в інформаційній системі волонтерської організації.....	34
3 Технологічне забезпечення удосконаленого методу ALPHA-MINER в інформаційній системі волонтерської організації	38
4 Реалізація та експериментальна перевірка роботи удосконаленого методу alpha-miner.....	43
4.1 Реалізація удосконаленого методу Alpha-miner для інформаційної системи волонтерської організації	43
4.2 Підготовка вхідних тестових вхідних даних удосконаленого методу Alpha- miner для інформаційної системи волонтерської організації	48
4.3 Експериментальна перевірка удосконаленого Alpha-miner методу.....	52
Висновки	59
Перелік джерел посилання	60

Додаток А Графічний матеріал.....	64
-----------------------------------	----

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ІС – інформаційна система;

БД – база даних;

WF - Workflow nets;

СУБД – система управління базами даних;

CSV – Comma-Separated Values;

DFD – Data Flow Diagrams (діаграми потоків даних);

IDEF0 – методологія функціонального моделювання та графічна нотація.

ВСТУП

У зв'язку з ситуацією, що склалась у країні, кількість потребуючих людей стрімко зростає. Тому діяльність організацій з надання допомоги продуктами, ліками, товарами першої необхідності та їх доставки має дуже поширений попит. Процеси обліку людей та закладів, що потребують допомоги, компаній та окремих осіб, що бажають та можуть надати матеріальну чи фінансову допомогу, товарів та ліків, що поступають і передаються за потребами, планування і організація доставки отримувачам допомоги, а також усі супутні бізнес процеси діяльності волонтерських організацій є складними задачами.

На даний момент бізнес процеси багатьох новоутворених волонтерських організацій не є вдало побудованими, тому мають багато недоліків і потребують аналізу та оптимізації на основі даних, що отримані, або отримуються у результаті функціонування.

Існують різні технології аналізу для оптимізації бізнес процесів. Для обраної предметної області було обрано технологію Process Mining. Це технологія відновлення моделей реальних бізнес-процесів на основі даних, що зберігаються в інформаційних системах. Отримані результати допомагають побачити та виключити вузькі місця, трансформувати та оптимізувати роботу компанії.

Так як інформаційні системи мають різні підходи до зберігання даних, тому вказані методи не є добре налагодженими для інтегрування у різні ІС без попередньої перевірки, форматування та обробки вхідних даних.

В ході роботи необхідно провести аналіз та опис бізнес процесів волонтерської організації. Проаналізувати задачі управління процесами. Дослідити методи інтелектуального аналізу процесів Process Mining для вирішення задач. Обрати та виконати детальний опис кроків методу вирішення задачі аналізу та опису бізнес процесів інформаційної системи для подальшої оптимізації, а також удосконалити метод для функціонування в обраній

інформаційній системі волонтерської організації. Обрати технології для реалізації роботи алгоритму в інформаційній системі. Провести експериментальну перевірку удосконаленого методу та показати приклади роботи алгоритму. Порівняти роботу удосконаленого в роботі та стандартного Process Mining методу.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Дослідження бізнес процесів волонтерської організації

Об'єктом дослідження є бізнес процеси волонтерської організації. Організація надає волонтерську допомогу потребуючим людям з обмеженими можливостями, або людям у скрутному становищі. На даний момент має постійних спонсорів, а також певну базу постійних клієнтів.

Місія організації — створення зручного сервісу для допомоги людям, що потребують допомоги через певні обставини, проблеми зі здоров'ям, або через похилий вік та нездатність забезпечити себе.

На даному етапі розвитку організації функціонує інформаційна система, що автоматизує процес прийому та обліку замовлень на допомогу, процес комплектації та планування доставки, процес обліку клієнтів, меценатів та організацій.

Основним процесом діяльності організації є облік заявок та планування доставки. Відділами, що відповідають за даний процес є - відділ роботи з клієнтами, відділ планування та фінансів, а також відділ доставки.

Першим етапом процесу є приймання заявки, яка оброблюється та підтверджується працівниками відділу роботи з клієнтами та передаються до планово-фінансового відділу для обліку та опрацювання. Працівники планово-фінансового відділу приймають сформовані заявки та заносять до спеціального журналу обліку. Дані заявки розглядаються та узгоджуються з керівником – головним бухгалтером, приймається рішення щодо внесення замовлень до журналу підтверджених. Даний журнал використовується працівниками планово-фінансового відділу для планування доставки пакетів допомоги, що збираються за даними з журналу відділом доставки. Результатом процесу планування доставки є план на доставку, який використовується для доставки.

Для опису бізнес процесів в інформаційній системі, потоків даних бізнес процесів, а також для зображення та виявлення відносини між цими процесами

створено діаграму IDEF0[27-30] для опису основних бізнес процесів організації та декомпозицію узагальненої моделі опису основних бізнес процесів організації. Ці діаграми зображено на рисунках 1.1.1 – 1.1.2.

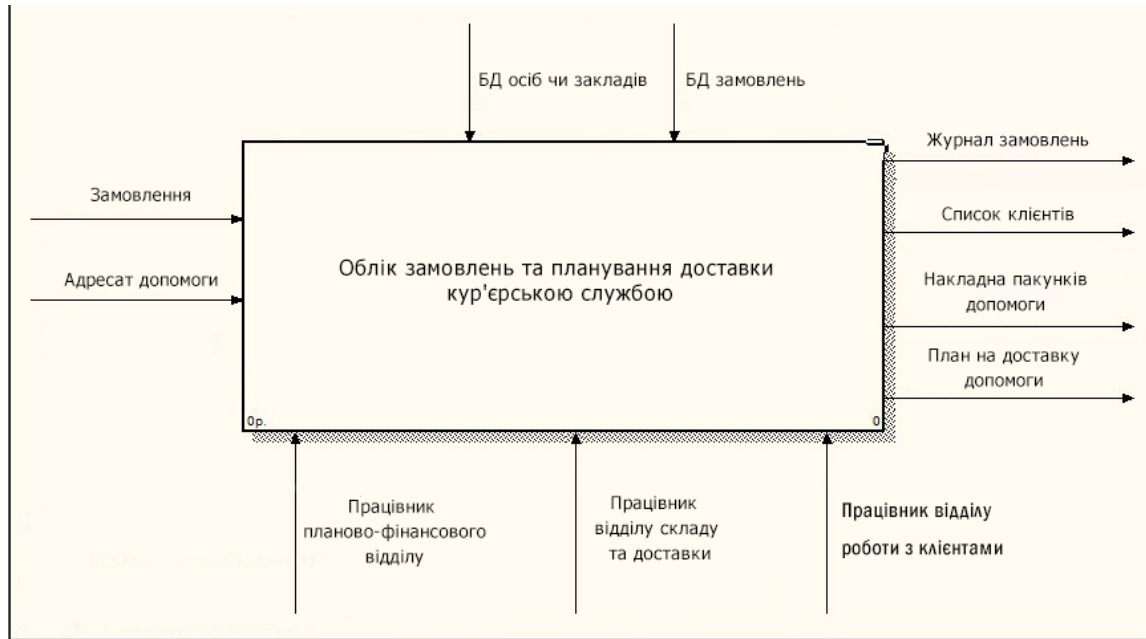


Рисунок 1.1.1 – Узагальнена модель опису основних бізнес процесів організації

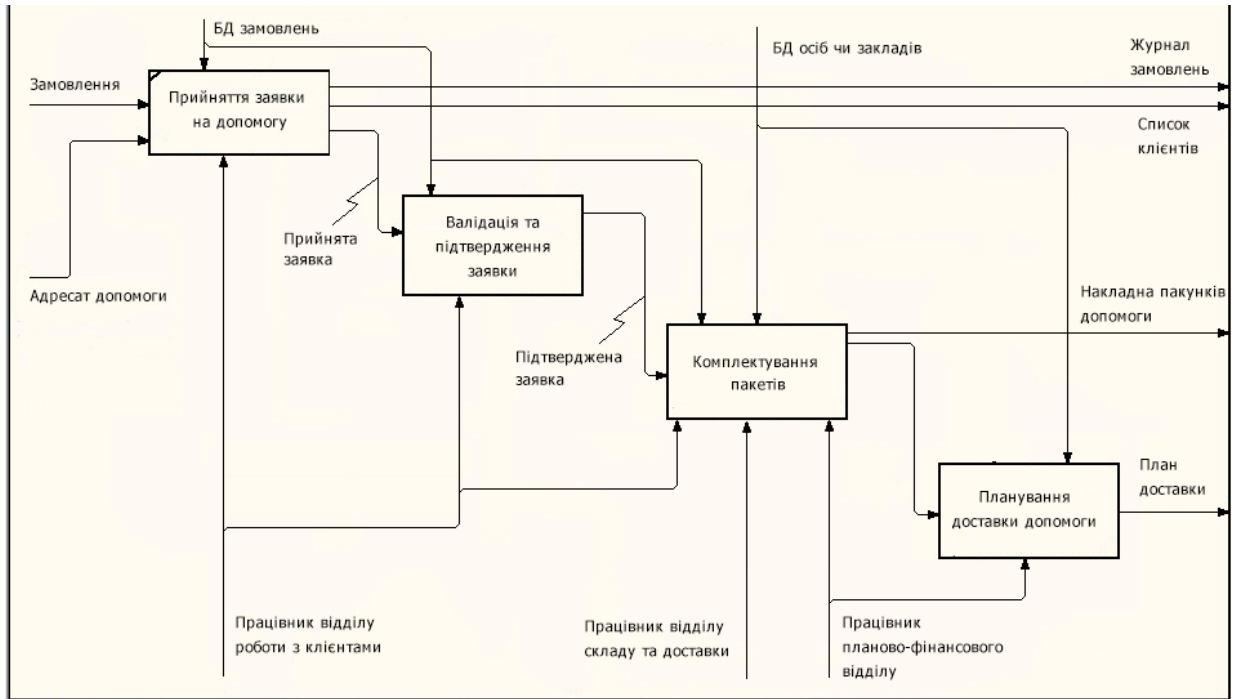


Рисунок 1.1.2 – Декомпозиція узагальненої моделі опису основних бізнес процесів організації

Для графічного зображення потоків даних було побудовано контекстну діаграму потоків даних[26]. Контекстна діаграма потоків даних зображена на рисунку 1.1.3.

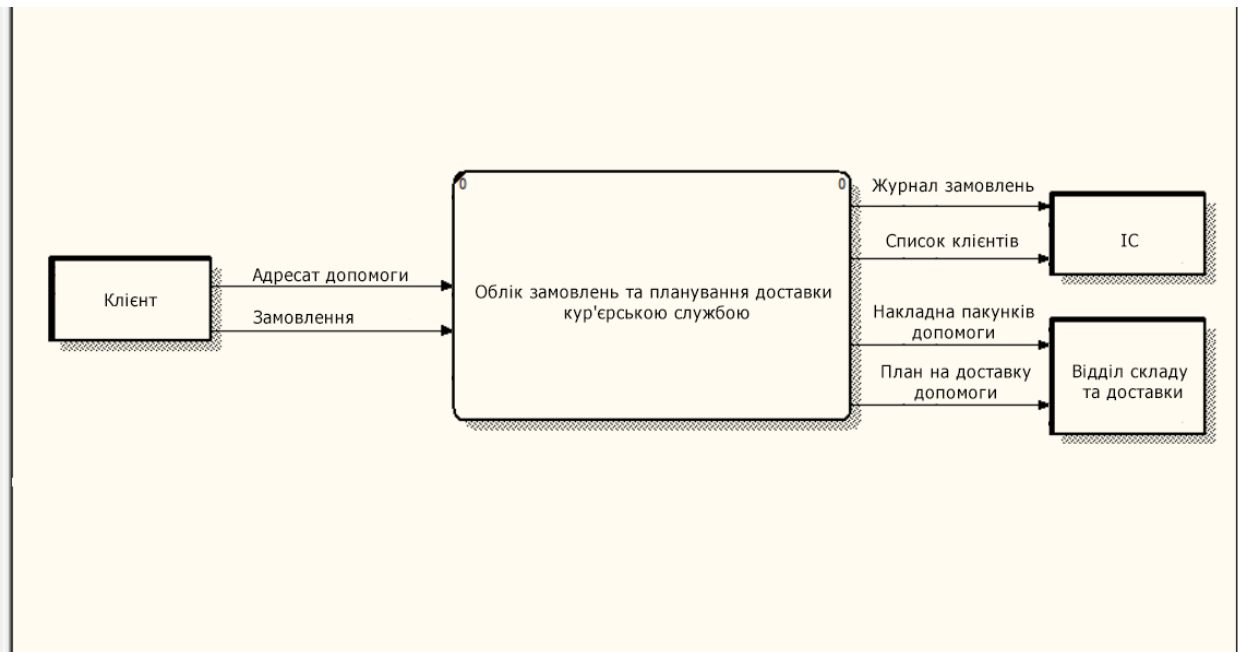


Рисунок 1.1.3 – Контекстна діаграма потоків даних основних бізнес процесів організації

Структура організації є лінійно-функціональна. Таку структуру можна охарактеризувати комбінацією лінійної та функціональної.

Лінійно-функціональна структура організації надає можливість управління процесами за лінійною схемою, а функціональні підрозділи допомагають лінійним керівникам для вирішення функцій. Керівники функціональних підрозділів безпосередньо впливають на виконавців.

Лінійно-функціональна структура є оптимальною, бо дана організація не занадто велика і відокремлення функціональних структур надає можливість ефективно автоматизувати обрані функції.

Організація формується з таких підрозділів та керівників:

- директор;
- планово-фінансовий відділ (керівник – головний бухгалтер);
- відділ роботи з клієнтами (керівник – адміністратор по роботі з клієнтами);

- відділ персонального обліку (керівник – завідуючий відділом персонального обліку);
- відділ рекрутингу (керівник – завідуючий відділом рекрутингу);
- інформаційний відділ (керівник – завідуючий інформаційним відділом);
- відділ складу та доставки (керівник – завідуючий відділом складу та доставки).

Структурну схему організації наведено на рисунку 1.1.4.

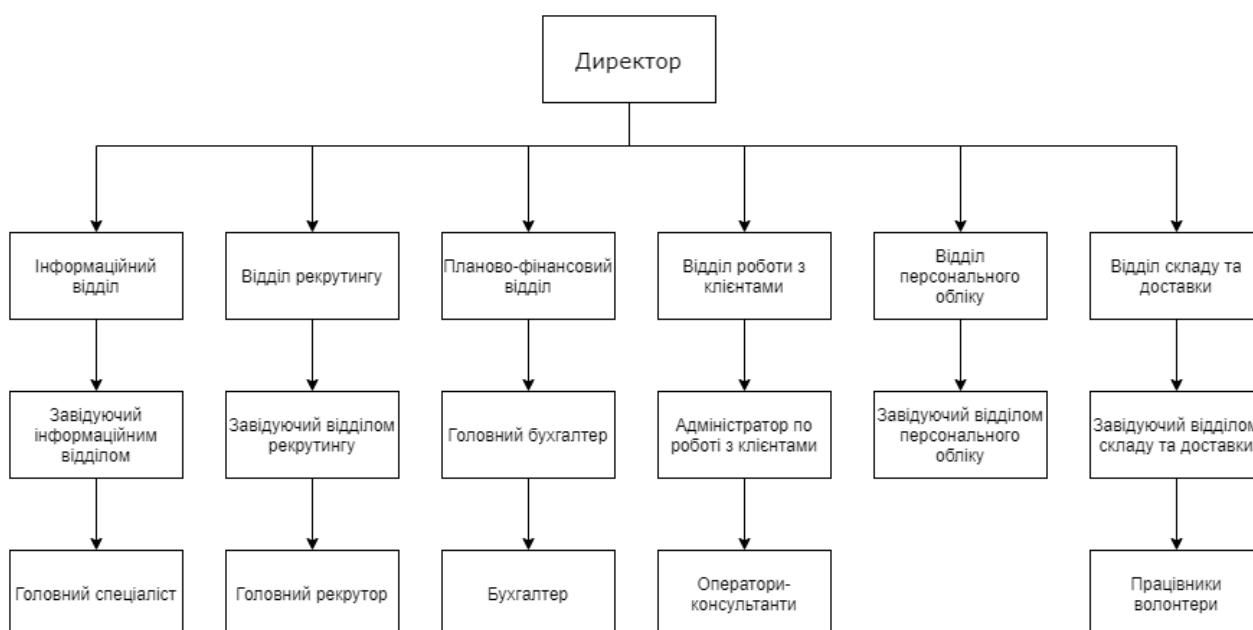


Рисунок 1.1.4 - Структурна схема організації

Директор контролює роботу усієї установи, приймає ключові рішення стосовно питань розвитку та напрямку роботи організації, перевіряє та аналізує звітність відділів.

Відділ роботи з клієнтами складається з керівника – адміністратора по роботі з клієнтами, а також з операторів-консультантів. Оператори консультанти допомагають створювати заявки на допомогу в телефонному режимі та в режимі онлайн консультанта, відповідають на питання від клієнтів системи. Також оператори-консультанти комунікують зі спонсорами, організаціями та людьми,

що хочуть допомогти організації фінансово та перенаправляють до інформаційного відділу. Зареєстровані заявки на допомогу підтверджуються операторами консультантами через систему та при необхідності по телефону. Адміністратор по роботі з клієнтами контролює роботу відділу, допомагає, при необхідності, операторам-консультантам і формує звіт роботи відділу.

Відділ планування та фінансів складається з керівника – головного бухгалтера та підлеглих. Головний бухгалтер розподіляє роботу між працівниками відділу та контролює виконання функцій відділу. Головним процесами відділу є керування фінансами компанії, процес комплектації, планування доставки допомоги.

Інформаційний відділ складається з керівника – завідуючого інформаційним відділом та підлеглих. Завідуючий разом з підлеглими організовують планові заходи щодо поширення інформації про необхідну допомогу та про діяльність організації в цілому. Комунікують зі спонсорами, організаціями та людьми, що хочуть допомогти організації фінансово та вирішують питання організації доставки допомоги. За списком необхідних для допомоги товарів виконується розсилка, або у телефонному режимі сповіщення постійних меценатів та спонсорів.

Відділ рекрутингу складається з завідуючого відділом рекрутингу та підлеглих працівників відділу. Працівники відділу рекрутингу відповідають за функцію пошуку та залучення людей волонтерів до непостійної діяльності, необхідної для реалізації доставки, або певної допомоги для потребуючих.

Відділ персонального обліку складається з однієї посади – завідуючий відділом персонального обліку. Задачі відділу: облік працівників, формування документу на відпустки, формування звіту обліку задіяних волонтерів.

Відділ складу та доставки складається з керівника – завідуючого відділом складу та доставки і підлеглих – тимчасових працівників-волонтерів. Задача відділу – приймати та розташовувати допомогу від спонсорів і меценатів на складі та за планами на доставку організовувати групування товарів зі складу та їх доставку потребуючим. Прийняті товари заносяться до БД в список доступних

товарів. Для товарів використаних у формуванні пакетів допомоги записи в БД оновлюються, а створені видані комплекти допомоги фіксуються новими записами.

1.2 Аналіз задач процесного управління

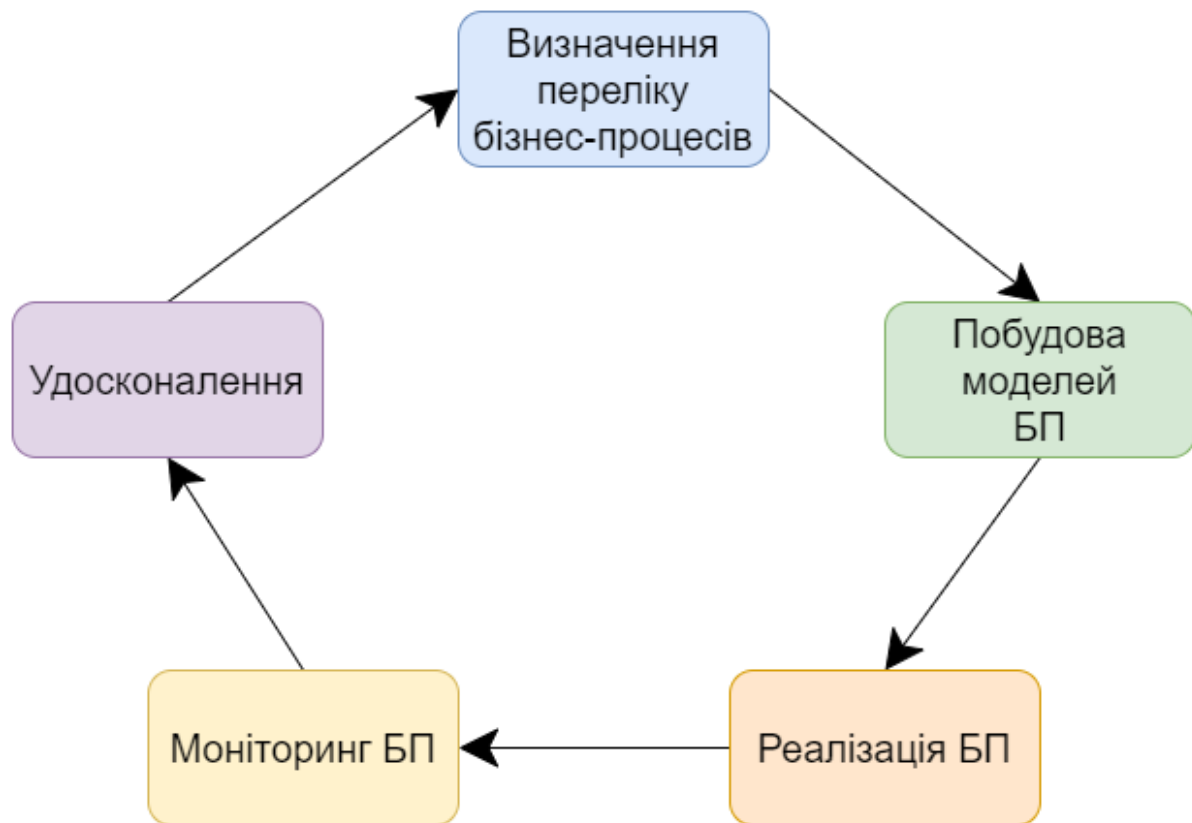
Управління процесами або управління бізнес-процесами — це організаційна дисципліна, яка надає інструменти та ресурси для аналізу, визначення, оптимізації, моніторингу та контролю бізнес-процесів, а також для вимірювання та підвищення ефективності взаємозалежних бізнес-процесів[19].

Управління процесами використовується організаціями, щоб визначити, як функціонує організація в цілому та як вона може працювати ефективніше.

Ось що зазвичай включає в себе управління процесами:

- покращення процесів, щоб робота могла виконуватися швидше;
- очищення будь-якого зайвого та марнотратного процесу;
- створення робочого середовища постійного вдосконалення.

Систематично впроваджуючи бізнес процеси, зменшується час та використання ресурсів, які витрачаються на виконання повторюваних завдань, і зводите до мінімуму помилки, спричинені неефективністю персоналу. Це також запобігає втраті даних і пропущеним крокам у процесі.



Рисунку 1.2.1 - П'ять ключових етапів життєвого циклу управління процесом

На рисунку 1.2.1 зображено п'ять ключових етапів життєвого циклу управління процесом.

Етап 1. Визначення переліку бізнес-процесів. На цьому етапі організація визначає процеси і аналізує, що хоче покращити.

Етап 2. Побудова моделей БП. На цьому наступному етапі поточний стан процесів має бути накреслений і розроблена ідеальна модель майбутнього управління процесом.

Етап 3. Реалізація БП. На етапі реалізації або виконання ідеальна модель інтегрується в бізнес. Для цього може знадобитися додавання технології, оновлення процедур або зміни ресурсів, навчання чи управління проектом.

Етап 4. Моніторинг БП. На етапі моніторингу реалізована ідеальна модель процесу вільно працює, поки виконується збір даних про ефективність моделі.

Етап 5. Удосконалення. На цьому етапі процес постійно вдосконалюється на основі інформації, зібраної на етапі моніторингу і в міру того, як бізнес-процеси змінюються з часом.

У зв'язку з ситуацією в країні, обрана для дослідження волонтерська організація була створена за дуже короткий проміжок часу. Через це, на даному етапі функціонування ІС волонтерської організації, бізнес процеси та в цілому організація роботи не є достатньо ефективною. Інформаційна система існує, але не має однозначного уявлення та досконалого бачення реальних взаємодій між процесами, тобто не має чіткої моделі поточного стану. У результаті не має можливості виявлення слабких місць, або напрямку удосконалення та росту.

Задача побудови і удосконалення бізнес-процесів волонтерської організації вирішується на основі порівняння поточної моделі процесів, тобто моделі, що відображає реальні процеси організації, та ідеальної моделі, що створена при проектуванні ІС[4]. За допомогою методів Process Mining потрібно виконати моделювання процесів, що відображатиме поточний стан. Створена в результаті роботи алгоритму модель буде використана для аналізу та удосконалення процесів організації.

1.3 Дослідження методів інтелектуального аналізу процесів

Для аналізу та оптимізації бізнес процесів волонтерської організації було обрано та досліджено методи інтелектуального аналізу Process Mining. Process Mining або процесна аналітика це методи і підходи, призначені для аналізу та вдосконалення бізнес-процесів в інформаційних системах на основі вивчення системних даних про виконані операції. Основною ідеєю методів є отримання знань про структуру та поведінки процесу з журналів подій, створених інформаційними системами під час функціонування.[31-34]



Рисунок 1.3.1 - Три основні типи процесної аналітики

На рисунку 1.3.1 зображено, що процесна аналітика встановлює зв'язки між фактичними процесами реального світу та даними, зібраними в журнал подій з одного боку та моделі процесів з іншого.

Сучасні інформаційні системи можуть реєструвати величезну кількість подій, які відбуваються в процесах. Вони зберігаються в різних таблицях баз даних, сервісах, звітах та інших. Однак скоріше за все ІС зберігають такі дані в неструктурованому вигляді. Наприклад, дані про події, розподілені по багатьох таблицях, або необхідні дані про події знаходяться в інших системах та підсистемах. У таких ситуаціях дані про подію існують, але для вилучення потрібні зусилля. Тому вилучення даних є невід'ємною частиною будь-якого процесу процесної аналітики.

Для структуризації даних про події використовуються журнали подій. Вони складаються з подій, які чітко визначені, стосуються точно одного випадку та однієї діяльності. При створенні журналу подій, можна створювати різні види залежно від бажаної мети дослідження та аналізу. Для збереження та організації журналів подій найбільш оптимальною технологією є СУБД.

За допомогою журналу подій можна проводити три типи аналітики: виявлення, перевірка відповідності і вдосконалення. На рисунку 1.3.1 також можна побачити ці три типи.

Виявлення - схеми процесу не існує і вона будується на основі журналу подій. Наприклад у результаті процесної аналітики виявляється та будується мережа Петрі.

Перевірка відповідності - схема процесу існує. Використовується схема (модель процесу), щоб перевірити, чи реальність відповідає цій схемі. Перевірка також може бути використана для виявлення відхилень, локалізації та пояснення цих відхилень і вимірювання їх виразності.

Покращення - схема процесу існує. Ця схема розширена новим аспектом або перспективою. Таким чином, ідея полягає в тому, щоб збагатити схему, наприклад, для виявлення вузьких місць.

Після аналізу діяльності волонтерської організації було виявлено, що організація не має повної картини того, як просуваються бізнес-процеси. Методи процесної аналітики можуть моделювати процеси якими вони є і допомогти користувачам зрозуміти їхню ситуацію, використовуючи журнали подій і дані в реальному часі. Також дослідження процесів дозволяє організації виявити, як фактичні процеси відрізняються від ідеальних. За допомогою інструментів аналізу процесу можна знайти невидимі відхилення процесів та їх частоту.

Існують різні алгоритми(методи) процесної аналітики, що надають змогу організації покращити ефективність та продуктивність обраного процесу, або підкреслити необхідність відповідності організації вимогам, правилам і нормам стосовно процесу. Ці алгоритми допомагають очищати необхідні дані та генерувати моделі процесів і мають різні сильні та слабкі сторони. Потрібно

вирішити, який алгоритм використовувати на основі даних про інформаційну систему.

Було досліджено декілька найбільш відомих та найбільш використовуваних алгоритмів:

Alpha-алгоритм. Alpha-алгоритм - це перший алгоритм, який допомагає створити міст між журналами подій або спостережуваними даними та моделями процесів. Alpha-алгоритм може будувати моделі процесів виключно на основі журналів подій, розуміючи причинно-наслідкові зв'язки між етапами процесів. Наприклад, в організаціях кілька дій можуть виконуватися паралельно, а не послідовно, щоб мінімізувати час виконання процесу. Alpha-алгоритм можна застосувати для виявлення паралельної активності, оскільки він розуміє події початку та кінця за допомогою подвійних позначок часу. На рисунку 1.3.2 Alpha-алгоритм знаходить процеси, що пов'язані з завершенням A - C, як окремі послідовності, а потім реструктурує їх разом у моделі за зв'язками, які вони мають.

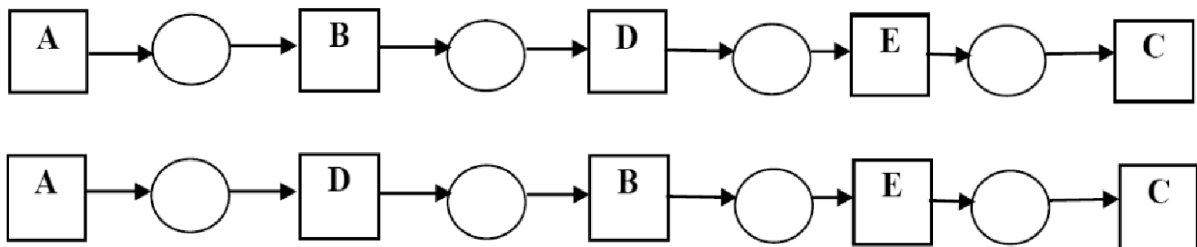


Рисунок 1.3.2 – Зображення двох послідовностей журналу подій

Для цього Alpha-алгоритм генерує:

- модель мережі Петрі, в якій усі переходи видимі, унікальні та відповідають класифікованим подіям;
- початкове маркування — описує статус моделі мережі Петрі на момент початку виконання;

– фінальне маркування — описує статус моделі мережі Петрі після завершення виконання.

Виконання процесу починається з подій, включених до початкової маркування і закінчується подіями, включеними до фінального маркування.

Alpha-алгоритм не розпізнає циклічних подій, або подій, що повторюються одна за одною. А також, він погано обробляє некоректні дані[12]. Некоректні дані – це нерелевантні або безглузді дані, які виникають через проблеми з якістю даних (наприклад, помилки введення даних, або їх неповнота) і негативно впливають на точність і простоту моделей процесу.

Евристичний алгоритм (Heuristic miner). Евристичний алгоритм – це стійкий до некоректних даних алгоритм, що застосовують для демонстрації поведінки процесу в погано структурованих даних. Алгоритм враховує лише порядок подій у процесі. Наприклад, він створює таблицю журналу подій із такими полями, як ідентифікатор процесу, ініціатор дії, позначка часу та дії, які розглядаються під час аналізу. Мітка часу дії використовується для обчислення порядку подій. Даний алгоритм включає три етапи:

- 1) Побудова графа залежностей.
- 2) Побудова вхідних і вихідних виразів для кожної діяльності.
- 3) Пошук залежності відносин на великій відстані один від одного.

Алгоритм враховує частоту подій. Можна описати лише події, які або виключно залежать одна від одної, або повністю незалежні.

Fuzzy miner. Алгоритм підходить для аналізу менш структурованих процесів, які демонструють велику кількість неструктурованої та конфліктної поведінки (наприклад, алгоритм перетворює спагетті-подібні моделі на більш стислі). Процеси можуть бути неструктурованими та суперечливими. Fuzzy miner застосовує різноманітні методи, такі як видалення неважливих країв, кластеризація високорельованих вузлів в одному вузлі та видалення ізольованих кластерів вузлів.

Індуктивний алгоритм (Inductive Miner). Алгоритм виявляє розвилки, які є умовами, або сполучними кроками між першою та кінцевою стадією процесу в

журналі. Існують різні типи розділень у журналі подій: послідовний, паралельний, одночасний, петля.

Після того, як індуктивний алгоритм визначає розвилку, він повторюється на дочірніх журналах подій, доки не знайде базовий варіант. Індуктивні моделі алгоритму дають унікальну позначку для кожного видимого переходу. Ці моделі використовують приховані переходи спеціально для розгалужень. На основі індуктивного алгоритму можна створити мережу Петрі або дерево процесів для відображення процесу.

В таблиці 1.3.1 зображено особливості та переваги розглянутих алгоритмів.

Таблиця 1.3.1 – Особливості та переваги досліджених алгоритмів Process Mining

Alpha-miner	<ul style="list-style-type: none"> - створення моделі процесів у вигляді мереж Петрі; - виявлення паралельної активності; - добре оброблює коректні дані; - найлегший алгоритм для початкового моделювання; - не оброблює циклічні події (дії, що повторюються).
Heuristic miner	<ul style="list-style-type: none"> - стійкий до некоректних даних; - події, які або виключно залежать одна від одної, або повністю незалежні; - враховує частоту подій.
Fuzzy miner	<ul style="list-style-type: none"> - аналіз менш структурованих процесів; - методи видалення неважливих країв, кластеризація високорельованих вузлів в одному вузлі та видалення ізольованих кластерів вузлів.
Inductive Miner	<ul style="list-style-type: none"> - виявляє розвилки, які є умовами, або сполучними кроками між першою та кінцевою стадією процесу;

	- побудова мережі Петрі або дерева процесів для відображення процесу.
--	---

1.4 Постановка задачі дослідження

У результаті аналізу існуючої інформаційної системи та стану організації було вирішено обрати для дослідження та налагодження метод процесної аналітики – Alpha-miner. Головними аспектами вибору є те, що алгоритм розуміє причинно-наслідкові зв'язки між етапами процесів[16]. Alpha-miner можна застосувати для виявлення паралельної активності, оскільки він розуміє події початку та кінця за допомогою подвійних позначок часу. В досліджуваній організації багато процесів може виконуватись паралельно, а не послідовно, щоб мінімізувати час виконання процесу. Саме тому для побудови моделі процесів волонтерської організації обрано Alpha-miner метод. Для початкового аналізу та побудови первинної моделі на основі добре структурованих даних найбільш підходящим методом є Alpha-miner. У майбутньому обраний метод також можна буде використовувати для інтеграції з іншими алгоритмами, або для порівняння результуючих моделей.

Метод інтелектуального аналізу процесів Alpha-miner дає можливість визначити логічні зв'язки між подіями БП, а також паралельне виконання дій БП, але не враховує зашумлених вхідних даних, що не дає можливість побудувати коректні моделі у випадку некоректних значень з журналу[31-34]. Alpha-miner метод використовує стандартизовані вхідні дані, в яких виділені окремі траси послідовностей [5]. При виконанні БП в екстремальних умовах, що, зокрема характерно для БП волонтерської організації, такі некоректні дані час від часу потрапляють до журналу подій. Використання некоректних даних може привести до помилок в моделі фактичного бізнес-процесу. Тому для побудови моделі БП методом Alpha-miner доцільно використати попередню обробку журналу подій та привести останній до стандартизованого вигляду.

Метою даної роботи є дослідження методів Process Mining для побудови й удосконалення бізнес процесів з урахуванням помилок в журналах подій.

Для досягнення мети, необхідно досліджувати наступні питання:

- дослідження бізнес процесів волонтерської організації;
- дослідження методів інтелектуального аналізу процесів Process Mining;
- аналіз методу інтелектуального аналізу Alpha-miner;
- удосконалення та налаштування методу Alpha-miner для роботи в інформаційній системі волонтерської організації;
- експериментальна перевірка удосконаленого методу Alpha-miner для побудови первинної моделі процесів волонтерської організації.

2 АНАЛІЗ ТА УДОСКОНАЛЕННЯ МЕТОДУ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ALPHA-MINER

2.1 Побудова моделі бізнес-процесу з використанням методу інтелектуального аналізу Alpha-miner

Alpha-алгоритм – це метод Process Mining, який дозволяє знаходити моделі процесів у вигляді мережі Петрі з журналу подій[34]. Ці моделі процесів також називають Workflow nets (WF). Відмінними характеристиками WF є:

- початковий стан - унікальний стан, який запускає всі ланцюги дій;
- кінцевий стан - унікальний стан, який закінчує всі ланцюжки дій;
- кожна окрема дія перебуває між початковим та кінцевим станом.

Приклад правильної WF схеми можна побачити на рисунку 2.1.1.

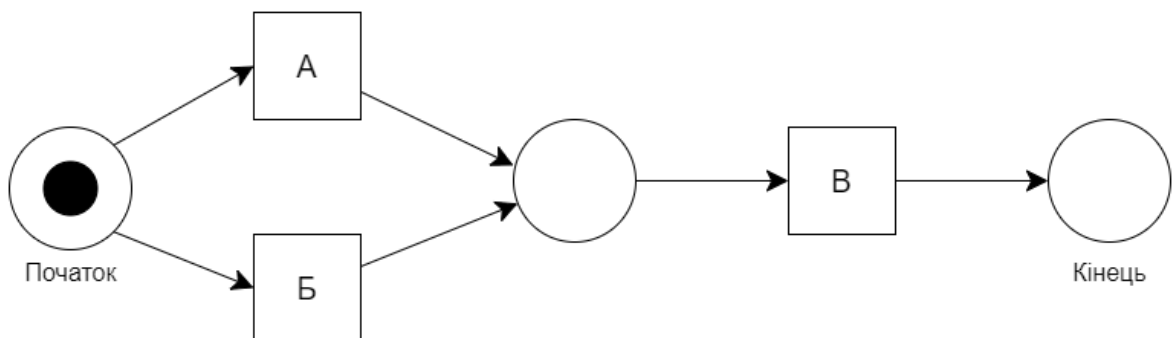


Рисунок 2.1.1 – Правильна схема WF у вигляді мережі Петрі

На рисунку 2.1.1 можна побачити, що у схеми є початковий стан, з якого починаються ланцюги дій. Дії позначено прямокутниками А, Б та В. Також можна побачити кінцевий стан і всі дії, що знаходяться між початком та кінцем мають закінчення.

Існують чотири відносини порядку дії, які може виявити Alpha-miner для побудови схеми WF[16].

Пряма послідовність – позначається знаком “>”. Наприклад дія Б слідує за дією А позначається як “А > Б”.

Причинно-наслідковий зв'язок – позначається знаком “→”. Якщо дія Б слідує за дією А, але А ніколи не слідує за дією Б, то “А → Б”.

Паралельна послідовність – позначається знаком “||”. Якщо дія Б слідує за дією А, а також існує послідовність, коли А слідує за дією Б, то “А || Б”.

Відсутність послідовності – позначається знаком “#”. Якщо А ніколи не слідує за дією Б, а також Б ніколи не слідує за дією А, то “А # Б”.

Алгоритм Alpha-miner виконує вісім послідовних етапів, у результаті яких створюється саме модель процесів у вигляді WF схеми:

Нехай L це журнал подій над T. Тоді виконання Alpha-miner алгоритму $\alpha(L)$ можна описати такими етапами:

1. На першому етапі алгоритм визначає усі події з журналу подій.

$$T_L = \{t \in T \mid \exists \sigma \in L \ t \in \sigma\}, \quad (2.1)$$

2. Другий етап роботи алгоритму визначає усі можливі початкові події.

$$T_I = \{t \in T \mid \exists \sigma \in L \ t = first(\sigma)\}, \quad (2.2)$$

3. Третій етап визначає усі можливі кінцеві події.

$$T_O = \{t \in T \mid \exists \sigma \in L \ t = last(\sigma)\}, \quad (2.3)$$

4. На четвертому етапі алгоритм рахує усі можливі відношення за допомогою множин А та В. Усі події знаходяться у множині А та В. Події, що належать множині А та в множині В повинні бути незалежними від самих себе. Усі подій, що належать множині А повинні бути причинно пов'язані з подіями в В.

$$\begin{aligned} X_L = \{ (A, B) \mid & A \subseteq T_L \wedge A \neq \emptyset \wedge B \subseteq \\ & T_L \wedge B \neq \emptyset \wedge \forall a \in A \forall b \in B \ a \rightarrow_L b - \\ & \wedge \forall a_1, a_2 \in A \ a_1 \#_L a_2 \wedge \forall b_1, b_2 \in B \ b_1 \#_L b_2 \}, \end{aligned} \quad (2.4)$$

Схематично це зображено на рисунку 2.1.2.

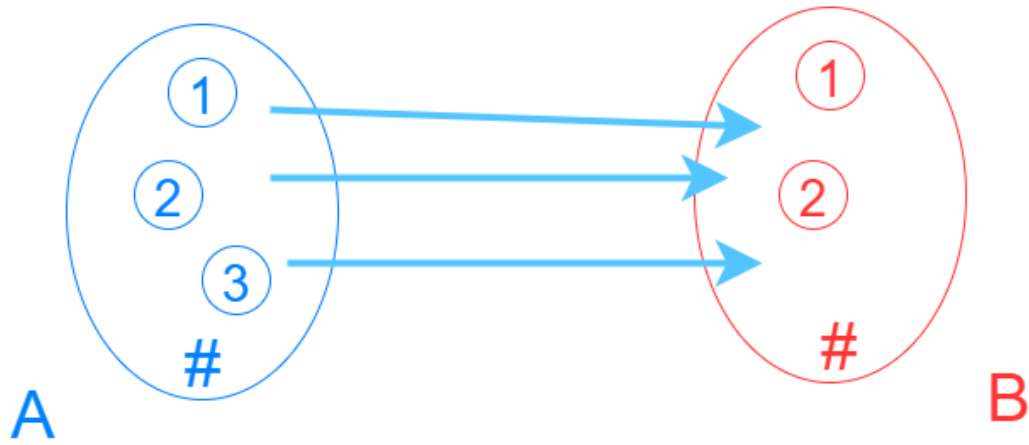


Рисунок 2.1.2 – Схематичне зображення відношень між подіями в множинах А і В

5. На п'ятому етапі видаляються всі не максимальні множини відносин.

$$Y_L = \{(A, B) \in X_L \mid \forall_{(A', B') \in X_L} A \subseteq A' \wedge B \subseteq B' \Rightarrow (A, B) = (A', B')\}, \quad (2.5)$$

6. Шостий етап розташовує всі множини відносин, що залишилися після п'ятого етапу на WF схемі та додає початковий і кінцевий стан.

$$P_L = \{p_{(A,B)} \mid (A, B) \in Y_L\} \cup \{i_L, o_L\}, \quad (2.6)$$

7. Сьомий етап додає зв'язки.

$$F_L = \{(a, p_{(A,B)}) \mid (A, B) \in Y_L \wedge a \in A\} \cup \{(p_{(A,B)}, b) \mid (A, B) \in Y_L \wedge b \in B\} \cup \{(i_L, t) \mid t \in T_I\} \cup \{(t, o_L) \mid t \in T_I\} \cup \{(t, o_L) \mid t \in T_o\}, \quad (2.7)$$

8. Останній етап повертає готову WF схему.

$$\alpha(L) = (P_L, T_L, F_L), \quad (2.8)$$

Для наглядного прикладу виконання етапів алгоритму було створено та представлено на рисунку 2.1.3 тестовий журнал подій L:

$$L = [(a,b,c,d)^3, (a,c,b,d)^2, (a,e,d)]$$

Рисунок 2.1.3 – Представлення тестового журналу подій L

Представлений журнал подій L складається з подій a, b, c, d, e. Журнал подій містить три екземпляри послідовності подій (a, b, c, d), два екземпляри послідовності подій (a, c, b, d) та один екземпляр (a, e, d). Етапи роботи алгоритму з тестовим журналом подій L:

1. На першому етапі алгоритм знаходить усі події – (a, b, c, d, e).
2. Знайти усі можливі початкові дії – (a).
3. Знайти усі можливі кінцеві дії – (d).
4. Для виконання четвертого етапу за допомогою чотирьох відносини порядку дії будується Footprint matrix. В таблиці 2.1.1 зображено Footprint matrix побудована для виконання четвертого етапу.

Таблиця 2.1.1 – Побудована Footprint matrix

L = [(a,b,c,d)^3, (a,c,b,d)^2, (a,e,d)]					
	a	b	c	d	e
a	#	→	→	#	→
b	←	#		→	#
c	←		#	→	#
d	#	←	←	#	←
e	←	#	#	→	#

За допомогою побудованої таблиці можна визначити можливі множини відношень: $(\{a\}, \{b\})$, $(\{a\}, \{c\})$, $(\{a\}, \{e\})$, $(\{b\}, \{d\})$, $(\{e\}, \{d\})$, $(\{a\}, \{b, e\})$, $(\{a\}, \{c, e\})$, $(\{b, e\}, \{d\})$, $(\{c, e\}, \{d\})$.

5. Видалення усіх не максимальних множини відносин. На рисунку 2.1.4 зображена декомпозиція складних множин:

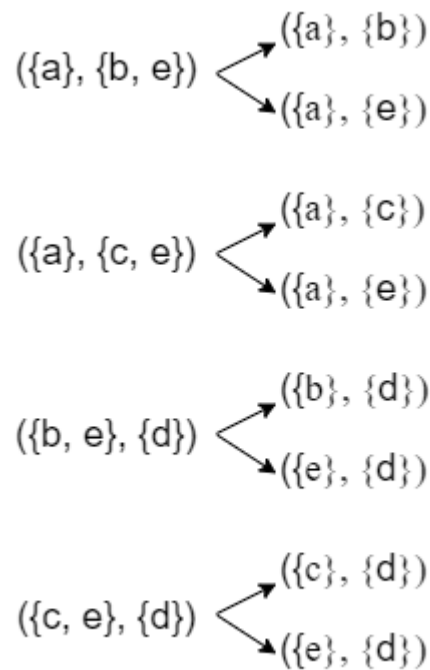


Рисунок 2.1.4 – Декомпозиція складних множин

У результаті декомпозиції виявлено, що множини $(\{a\}, \{b\})$, $(\{a\}, \{c\})$, $(\{a\}, \{e\})$, $(\{b\}, \{d\})$, $(\{c\}, \{d\})$, $(\{e\}, \{d\})$ можна видалити.

Множини відношень, що залишились після видалення: $(\{a\}, \{b, e\})$, $(\{a\}, \{c, e\})$, $(\{b, e\}, \{d\})$, $(\{c, e\}, \{d\})$.

6. На шостому етапі потрібно розташовувати всі множини відносин, що залишились після п'ятого етапу на WF схемі та додати початковий і кінцевий стан.

7. Сьомий етап додає зв'язки. На рисунку 2.1.5 зображена схема розташованих множин відносин із усіма зв'язками, де перехід $p1 = (\{a\}, \{b, e\})$, $p2 = (\{a\}, \{c, e\})$, $p3 = (\{b, e\}, \{d\})$, $p4 = (\{c, e\}, \{d\})$.

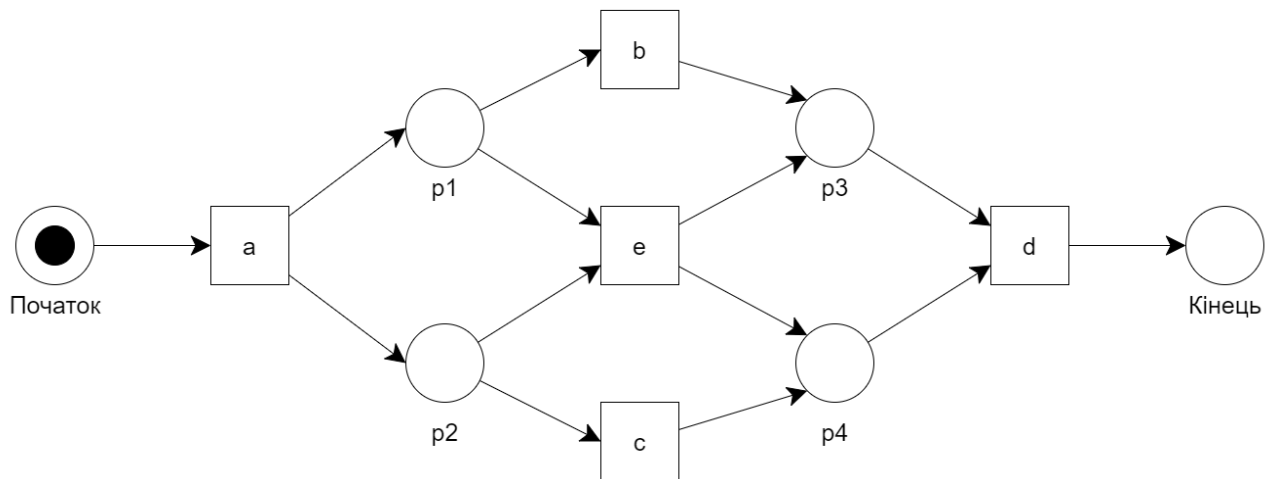


Рисунок 2.1.5 – Готова модель WF у вигляді мережі Петрі для тестового журналу подій L

8. Останній етап повертає готову модель WF, що зображена на рисунку 2.1.5.

Методи інтелектуального аналізу, зокрема Alpha-miner базуються на журналах подій. Журнали подій можна описати, як послідовність послідовностей подій. У більшості випадків, якщо процес в досліджуваній інформаційній системі підтримується будь-якою ІТ-системою, він створює певний журнал дій, виконаних користувачами. Наприклад, журнал може містити всі дії, які користувач виконав у програмі.

Щоб здійснити виявлення процесу, набір даних повинен обов'язково містити такі 3 типи інформації:

- ідентифікатор об'єкта дії (Case ID) — унікальний ідентифікатор екземпляру процесу. Ідентифікатор об'єкта дії має впорядкований набір пов'язаних із ним подій з позначкою часу. Можливо використовувати різні ідентифікатори (це залежить від варіанту використання);
- подія (Event) — крок процесу, будь-яка діяльність, яка є частиною процесу, який ми аналізуємо;

– позначка часу (Timestamp) — використовується для оцінки продуктивності та визначення порядку подій, може бути час, коли користувач увійшов/вийшов із даної події (або обох фактично).

Крім того, ми можемо включити більш детальну інформацію, таку як ресурси, країна, сегмент користувачів тощо. Використовуючи додаткову інформацію, ми можемо провести набагато більш детальний аналіз.

Робота Alpha-алгоритму в інформаційній системі волонтерської організації дасть змогу побудувати схеми робочих процесів за допомоги мережі Петрі. Розроблені схеми надають можливість аналізу і складання повної картини про фактичні процеси, що проходять в організації. Після аналізу та опису отриманих за допомогою Alpha-алгоритму схем, організація може удосконалювати та інтегрувати удосконалену схему в інформаційну систему для ліквідації виявлених слабких місць та підвищення ефективності процесів. Після впровадження удосконалень можливе повторне використання Alpha-алгоритму для перевірки та порівняння удосконаленої схеми та фактичної, отриманої в результаті повторного використання методу процесної аналітики. Таким чином, після впровадження методу, організація може постійно удосконалювати і перевіряти процеси.

Проблемою алгоритму у рамках бізнес процесів волонтерської організації є відсутність імпортування та попередньої обробки даних у вигляді, що сприймається алгоритмом, тому для впровадження Alpha-miner алгоритму необхідно удосконалити і налаштувати даний метод. На рисунку 2.1.6 схематично зображено етапи виконання Alpha-miner алгоритму для побудови моделі процесів волонтерської організації. Можна побачити, що є необхідність зовнішньої реалізації імпорту вхідних даних, їх обробки, видалення некоректних помилкових даних, сортування та форматування у вигляді, зрозумілий Alpha-miner алгоритмом.

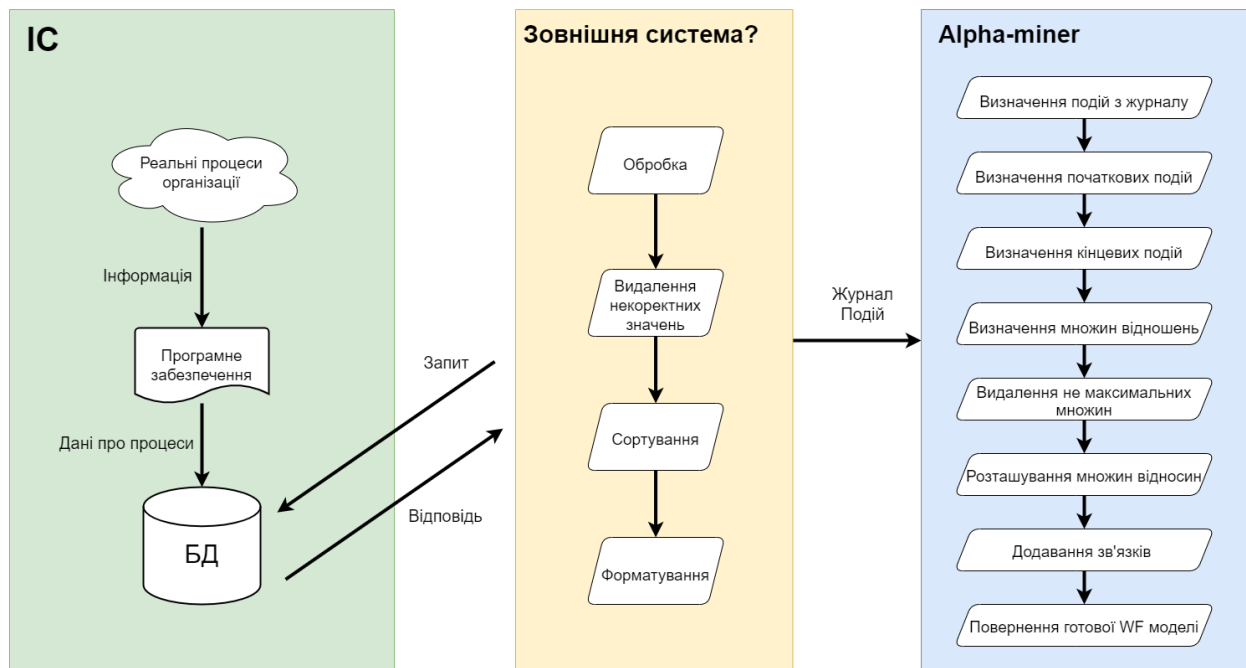


Рисунок 2.1.6 - Схема етапів виконання Alpha-miner алгоритму для побудови моделі процесів волонтерської організації

2.2 Удосконалення методу Alpha-miner для функціонування в інформаційній системі волонтерської організації.

Одним з головних компонентів для функціонування методів процесної аналітики та конкретно Alpha-miner алгоритму є вхідні дані. Вхідні дані, що можна використовувати у якості журналу подій зберігаються досліджуваною інформаційною системою волонтерської організації в реляційній БД. Для цього використовується система управління базами даних MySQL.

Після аналізу етапів роботи методу виникає проблема вибору, форматування та налагодження даних зі сховища інформаційної системи волонтерської організації, бо даний етап не є частиною виконання алгоритму.

Пропонованим рішенням проблеми є створення додаткового початкового етапу, що виконуватиме вибірку даних напряму з інформаційної системи,

виконуватиме парсинг, початкову обробку та налагоджування даних для подальшого використання в алгоритмі. Некоректні та циклічні події, які не сприймаються алгоритмом будуть видалені.

З використанням додаткового етапу зникає проблема підготовки даних за межами виконання методу. Виконання алгоритму буде розпочинатись із вводу даних для підключення до БД інформаційної системи. Після цього, за сформованим запитом на вибірку даних, буде виконано підключення до БД та імпорт необхідних даних. Отриману вибірку даних буде оброблено та форматовано у зрозумілий для алгоритму вид. Усі вхідні дані, що вводить користувач, потрібно перевірити та сповістити зрозумілими повідомленнями, якщо виникли помилки в роботі етапів методу. Удосконалений метод знаходить та вирішує помилки вхідних даних для Alpha-miner з журналу подій, такі, як помилкові дубльовані події та циклічні події, тобто ті самі події, що повторюються один за одним для унікального ідентифікатора Case ID. Також вхідний журнал подій потрібно форматувати та відсортувати для виділення трас послідовностей подій.

На рисунку 2.2.1 схематично зображено етапи виконання удосконаленого Alpha-miner алгоритму для побудови моделі процесів волонтерської організації. Тепер немає необхідності зовнішньої реалізації імпорту вхідних даних, їх обробки, видалення некоректних даних, сортування та форматкування у вигляд, зрозумілий Alpha-miner алгоритмом, все це реалізовано як додатковий початковий етап методу. Тепер запит на вибірку даних з інформаційної системи конфігуруються та виконується безпосередньо методом.

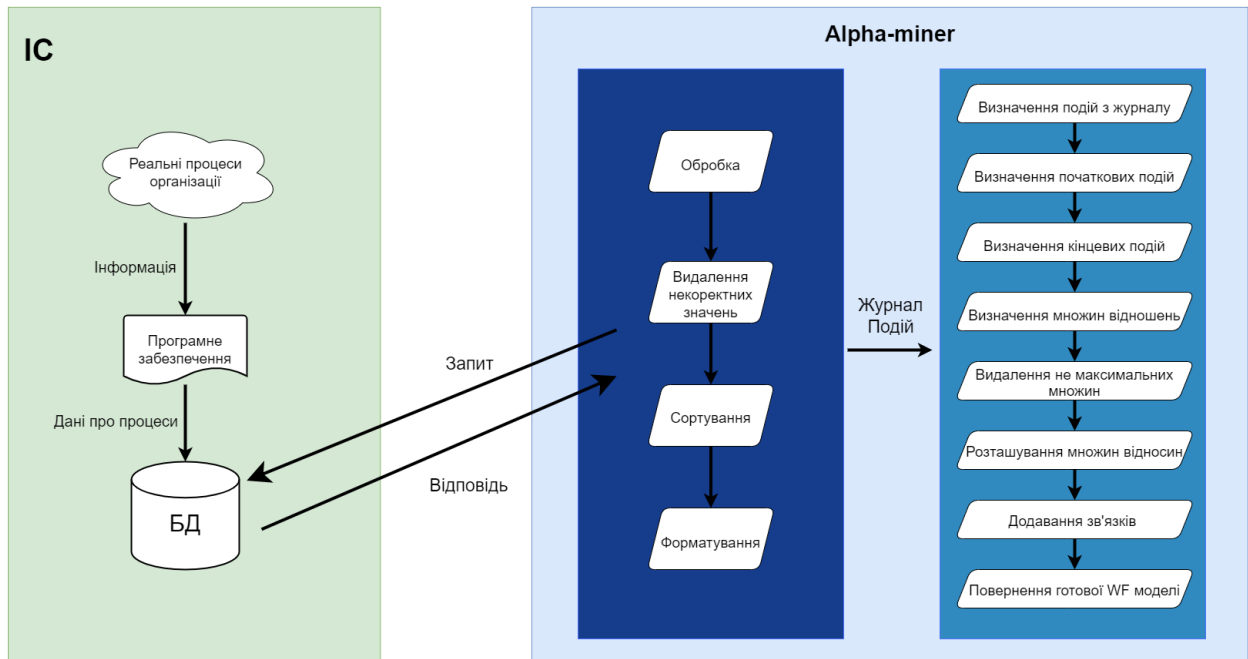


Рисунок 2.2.1 - Схема етапів виконання удосконаленого Alpha-miner алгоритму для побудови моделі процесів волонтерської організації

На схемі етапів виконання удосконаленого Alpha-miner алгоритму темно синім кольором виділено новий етап запити до БД та попередньої обробки, форматування та сортування вхідних даних.

У результаті удосконалення Alpha-miner алгоритму можна виділити додатковий початковий етап виконання з такими кроками:

1. Першим кроком додаткового етапу є виконання запити до БД інформаційної системи волонтерської організації для відбору вхідних даних журналу подій.
2. Другим кроком є обробка вхідних даних та виділення усіх подій журналу за заданий інтервал часу.

$$L^{(1)} = \{e_i : (\forall i) t_{\min} < t_i < t_{\max}\},$$

де t_i - момент виникнення події e_i ;

(t_{\min}, t_{\max}) - інтервал часу для відбору подій. (2.9)

3. Третім кроком є видалення усіх подій, що дублюються.

$$L^{(2)} \subseteq L^{(1)},$$

$$L^{(2)} = \{e_i : (\forall i \neq m) e_i \neq e_m\}, \quad (2.10)$$

4. Четвертий крок видаляє циклічних подій (події, що повторюються одна за одною).

$$L^{(3)} \subseteq L^{(2)},$$

$$L^{(3)} = \{e_i : \forall (e_i, e_m) \forall (e_k, e_l) (e_i \neq e_k \vee e_m \neq e_l)\}, \quad (2.11)$$

5. П'ятий крок виконує сортування для виділення трас послідовностей подій.

$$L^{(4)} = \{\langle e_1, \dots, e_i, \dots, e_I \rangle, \dots, \langle e_1, \dots, e_j, \dots, e_J \rangle\}, \quad (2.12)$$

6. У шостому кроці виконується форматування отриманих даних до типу, що сприймається Alpha-miner алгоритмом.

3 ТЕХНОЛОГІЧНЕ ЗАБЕЗПЕЧЕННЯ УДОСКОНАЛЕНОГО МЕТОДУ ALPHA-MINER В ІНФОРМАЦІЙНІЙ СИСТЕМІ ВОЛОНТЕРСЬКОЇ ОРГАНІЗАЦІЇ

Запропонована інформаційна технологія удосконаленого Alpha-miner методу призначена для побудови моделей процесів волонтерської організації.

В якості вхідних даних використовується журнал подій, що відображає записи про реальні події зареєстровані під час виконання процесів організації. Технологічні етапи удосконаленого Alpha-miner методу зображені на рисунку 3.1.

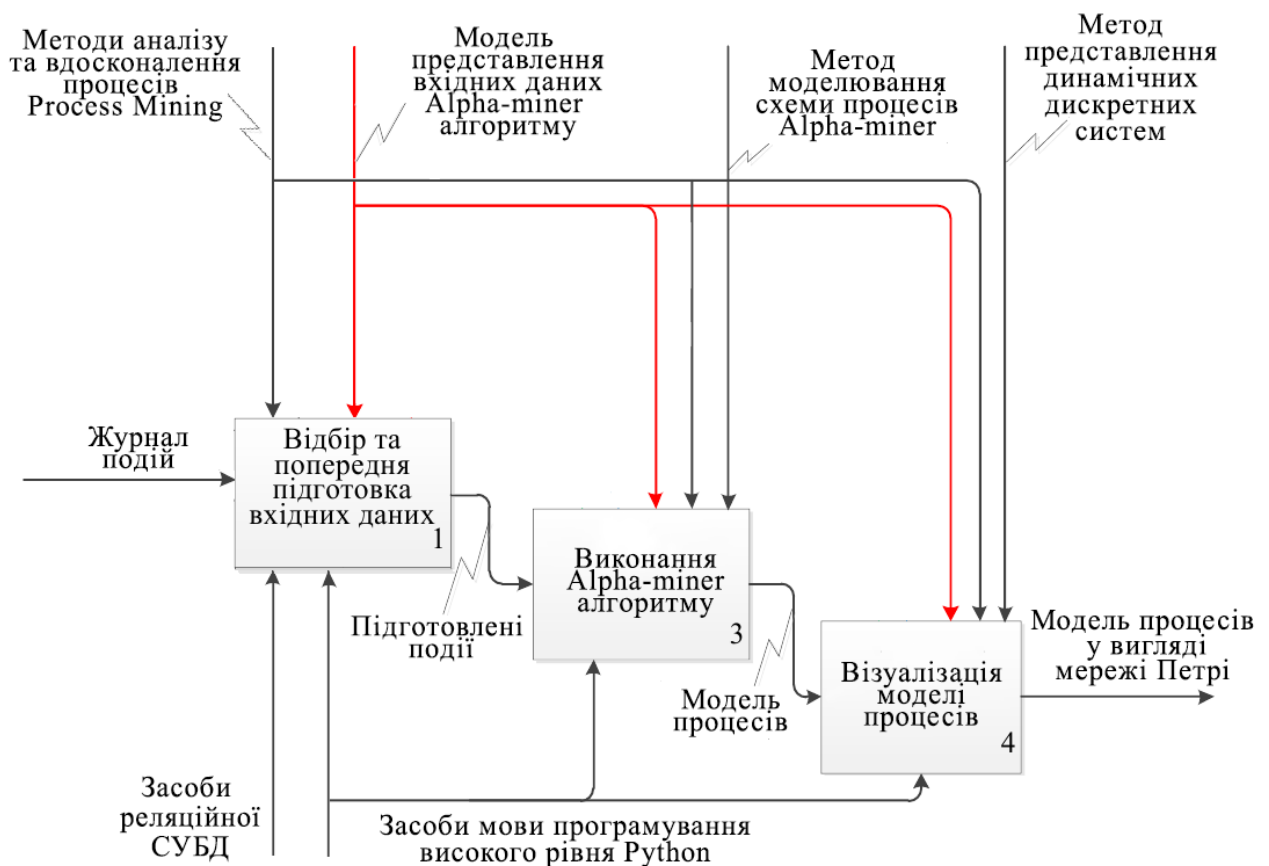


Рисунок 3.1 – Технологічні етапи удосконаленого Alpha-miner методу

На першому етапі формується журнал подій з урахуванням обов'язкових даних:

- ідентифікатор об'єкта дії (Case ID);
- подія (Event) ;
- позначка часу (Timestamp).

Дані відбираються засобами реляційної СУБД, а також за допомогою інструментів мови програмування високого рівня Python. Відібрані дані обробляються та формуються за допомогою моделі представлення вхідних даних Alpha-miner алгоритму, що використовується у наступних етапах. Відібрані дані оброблюються з використанням моделі представлення двовимірних табличних даних. За допомогою моделі виконується пошук та видалення помилок вхідних даних, таких, як дубльовані та циклічні події. Відформатовані дані сортуються для виділення трас послідовностей подій. У результаті етапу вихідними даними є підготовлені для виконання аналізу події.

На другому етапі підготовлені дані використовуються методом моделювання схеми процесів Alpha-miner для аналізу процесів та створення моделі. На даному етапі виконуються описані кроки виконання класичного Alpha-miner алгоритму:

1. Визначення усіх подій.
2. Визначення усіх можливих початкових дій.
3. Визначення усіх можливих кінцевих дій.
4. Визначення усіх можливих множин відносин.
5. Видалення усіх не максимальних множини відносин.
6. Розташування множин відносин.
7. Додавання зв'язків.
8. Повернення готової моделі процесів.

На останньому четвертому етапі готова модель процесів використовується для візуалізації за допомогою методу представлення динамічних дискретних систем - мережі Петрі.

Після дослідження методів Process Mining було проведено пошук та аналіз існуючих рішень реалізації саме алгоритму Alpha-miner. Найбільш підходящим рішенням для реалізації роботи алгоритму в інформаційній системі було обрано бібліотеку Python – PM4Py[18].

Python — інтерпретована об'єктно-орієнтована мова програмування високого рівня. Має строго-динамічну типізацію. В Python підтримуються різні парадигми програмування: об'єктно-орієнтована, функціональна, процедурна та аспектно-орієнтована.

PM4Py — це бібліотека Python, яка підтримує найсучасніші алгоритми інтелектуального аналізу процесів на Python. Програмний продукт розроблений Інститутом прикладних інформаційних технологій Фраунгофера (FIT). Це є “Open Source” проект і дозволяється до використання в академічних та дослідницьких цілях і промислових проектах. Проект знаходиться у вільному доступі в репозиторії GitHub та має ліцензію “GNU General Public License v3.0”. На рисунку 3.2 зображені особливості ліцензії використання бібліотеки.


 pm4py/pm4py-core is licensed under the GNU General Public License v3.0 <small>Permissions of this strong copyleft license are conditioned on making available complete source code of licensed works and modifications, which include larger works using a licensed work, under the same license. Copyright and license notices must be preserved. Contributors provide an express grant of patent rights.</small>	Permissions ✓ Commercial use ✓ Modification ✓ Distribution ✓ Patent use ✓ Private use	Limitations ✗ Liability ✗ Warranty	Conditions ① License and copyright notice ① State changes ① Disclose source ① Same license
---	---	---	---

Рисунок 3.2 - Особливості ліцензії використання бібліотеки PM4Py

На рисунку 3.3 зображені опис, посилання на документацію, приклад використання та встановлення бібліотеки.

pm4py

pm4py is a python library that supports (state-of-the-art) process mining algorithms in python. It is open source (licensed under GPL) and intended to be used in both academia and industry projects. pm4py is a product of the Fraunhofer Institute for Applied Information Technology.

Documentation / API

The full documentation of pm4py can be found at <http://pm4py.org/>

First Example

A very simple example, to whet your appetite:

```
import pm4py

if __name__ == "__main__":
    log = pm4py.read_xes('<path-to-xes-log-file.xes>')
    net, initial_marking, final_marking = pm4py.discover_petri_net_inductive(log)
    pm4py.view_petri_net(net, initial_marking, final_marking, format="svg")
```

Installation

pm4py can be installed on Python 3.7.x / 3.8.x / 3.9.x / 3.10.x by invoking: `pip install -U pm4py`

Release Notes

To track the incremental updates, please refer to the *CHANGELOG* file.

Рисунок 3.3 - Опис, посилання на документацію, приклад використання та встановлення бібліотеки PM4Py

ІС волонтерської організації використовує у якості сховища та керування даними СУБД MySQL.

Для попередньої обробки, сортування та конвертації даних використано модулі “pandas” та “numpy”.

На рисунку 3.4 зображено схему використаних технологій для реалізації удосконаленого методу Alpha-miner для роботи в ІС волонтерської організації.

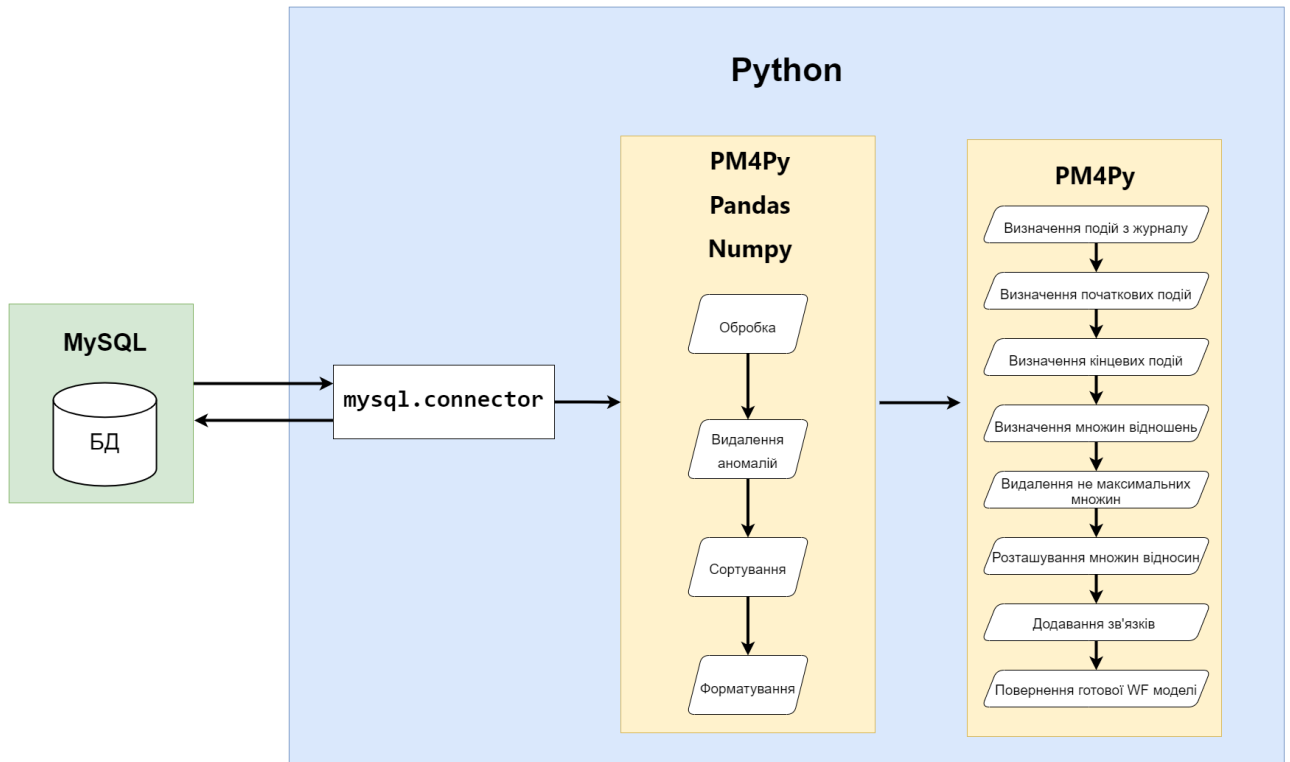


Рисунок 3.4 - Схема використаних технологій для реалізації удосконаленого методу Alpha-miner

4 РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА РОБОТИ УДОСКОНАЛЕНОГО МЕТОДУ ALPHA-MINER

4.1 Реалізація удосконаленого методу Alpha-miner для інформаційної системи волонтерської організації

Для реалізації удосконаленого методу Alpha-miner для інформаційної системи волонтерської організації було визначено задачі, що потребують експериментальної перевірки. Перш за все потрібно перевірити функціонал доступу до БД для відбору вхідних даних журналу подій. Користувач має можливість вводу даних для підключення до БД. Якщо введено невірні дані, система повинна відобразити повідомлення про помилку та запропонувати повторно ввести дані. Якщо помилка підключення виникла після третьої спроби, система повинна завершити роботу. Етапи роботи методу повинні відображатись повідомленнями для користувача. Після успішного виконання програма повинна вивести на екран результуючу модель процесів, розроблену на основі вхідного журналу подій. Для перевірки додаткового етапу удосконаленого методу потрібно виконати тестовий запуск роботи алгоритму з підготовленими тестовими даними, що містять некоректні дані. Тестовий вхідний журнал подій повинен містити циклічні події, а також події, що дублюються. Очікуваним результатом запуску удосконаленого методу є правильно побудована модель процесів, з урахуванням зашумлених даних. Для оцінки та порівняння удосконаленого та стандартного методів потрібно виконати роботу стандартного Alpha-miner алгоритму над тими ж даними з помилковими подіями.

Реалізація, відладка та експериментальний запуск програми було виконано за допомогою середовища Visual Studio Code.

Спочатку потрібно імпортувати необхідні для функціонування програми бібліотеки та інструменти. На рисунку 4.1.1 зображено імпорт бібліотек та модулів для вводу, модифікації та конвертації даних.

```
# hidden input for password
from getpass import getpass

# data modification and convertation
import pandas as pd
import numpy as np
from pm4py.objects.conversion.log import converter as log_converter
from pm4py.objects.log.util import dataframe_utils
```

Рисунок 4.1.1 - Імпорт бібліотек та модулів для модифікації та конвертації даних

Імпортовано бібліотеку Pandas, яку використано для конвертації вхідних даних з БД інформаційної системи у внутрішній об'єкт програми типу Dataframe. Імпортовано модуль converter бібліотеки PM4Py для перетворення об'єкту, що містить вхідні дані типу Dataframe, до типу EventLog. Саме тип EventLog приймається методом алгоритму. Модуль numpy імпортовано для роботи з внутрішнім станом типу Dataframe. Останній модуль для конвертації даних це dataframe_utils. Використовується для перетворення даних про час події в тип timestamp.

Для роботи Alpha-miner алгоритму імпортовано модуль algorithm бібліотеки PM4Py. На рисунку 4.1.2 зображено імпорт модулю алгоритму.

```
# process mining - alpha_miner module
from pm4py.algo.discovery.alpha import algorithm as alpha_miner
```

Рисунок 4.1.2 - Імпорт модулю Alpha-miner алгоритму

Для підключення до БД волонтерської організації імпортовано модуль “mysql.connector”. На рисунку 4.1.3 зображено імпорт модулю підключення до БД.

```
# db connection
import mysql.connector as connection
```

Рисунок 4.1.3 - Імпорт модулю підключення до БД

Для візуалізації отриманих даних у результаті роботи методу у вигляді мережі Петрі імпортовано модуль “pm4py.visualization.petri_net”. На рисунку 4.1.4 зображено імпорт модулю для візуалізації моделі.

```
# vizualization
from pm4py.visualization.petri_net import visualizer as pn_visualizer
```

Рисунок 4.1.4 – Імпорт модулю для візуалізації моделі

Робота програми починається з імпорту запиту до БД інформаційної системи, що знаходиться у вхідному файлі “sql_input.sql”. Запит зчитується з файлу та оброблюється для виконання. Наступним кроком користувач вводить дані для підключення до БД інформаційної системи. Якщо користувач увів некоректні або невірні дані для підключення, або виникла проблема при створенні підключення, то програма сповістить користувача та запропонує ввести дані ще раз. Усього користувач має три спроби. Якщо за три спроби програма не зможе створити підключення, то програма завершиться. Після успішного підключення до БД програма посилає запит з вхідного файлу та перетворює відповідь у дані типу DataFrame. Після перетворення підключення до БД закривається. Для логування дані, що прийшли після запиту, зберігаються у форматі CSV. Далі програма видаляє події, що помилково дублюються, а також циклічні події. Після обробки конвертує та сортує дані про час події в типі timestamp. Готові дані перетворюються у зрозумілий алгоритму об’єкт представлення журналу подій типу EventLog. Об’єкт журналу подій передається

на опрацювання Alpha-miner алгоритму, яких повертає мережу Петрі, початкове та кінцеве маркування. Ці результуючі дані представляються для користувача за допомогою візуалізатора. На рисунку 4.1.5 зображений лістинг усіх імпортованих модулів програми.

```
import sys

# db connection
import mysql.connector as connection

# hidden input for password
from getpass import getpass

# data modification and convertation
import pandas as pd
import numpy as np
from pm4py.objects.conversion.log import converter as log_converter
from pm4py.objects.log.util import dataframe_utils

# process mining - alpha_miner module
from pm4py.algo.discovery.alpha import algorithm as alpha_miner

# vizualization
from pm4py.visualization.petri_net import visualizer as pn_visualizer
```

Рисунок 4.1.5 - Лістинг усіх імпортованих модулів програми

На рисунку 4.1.6 зображено частину коду програми, що виконує обробку запиту до БД з вхідного файлу, обробку введених користувачем даних для підключення до БД, а також спробу підключення та надсилання запиту на вхідні дані до БД інформаційної системи волонтерської організації.

```

# get sql query from input file
with open("sql_input.sql") as file:
    sqlQueryRows = [line.rstrip() for line in file]
sqlQuery = " ".join(str(x) for x in sqlQueryRows)

# create db connection
for attempt in range(3):
    inputHost = input("host: ")
    dbName = input("database: ")
    userName = input("user name: ")
    password = getpass("password: ")
    try:
        print("Connecting to db.")
        mydb = connection.connect(host=inputHost, database = dbName, user=userName, passwd=password, use_pure=True)
        print("Connected.")
        break
    except Exception as e:
        if attempt == 2:
            print("Error in db conection.")
            sys.exit()
        else:
            print(e)
            print("Error in db conection. Try again")
            continue

# get db data from response as Dataframe
df = pd.read_sql(sqlQuery, mydb)
mydb.close()

# input data logging
df.to_csv("input_data")

```

Рисунок 4.1.6 – Частина коду програми, що виконує запит на вхідні дані до БД

На рисунку 4.1.7 зображено частину коду програми, що виконує початкову обробку і конвертацію даних до стандартизованого та зрозумілого для Alpha-miner алгоритму типу. Після форматування даних починається виконання алгоритму над вхідними даними. Alpha-miner алгоритм оброблює журнал подій та передає результуючі дані для візуалізація фінальної моделі WF у вигляді мережі Петрі.

```

# input data converting and processing
print()
print("Processing input data..")
df = dataframe_utils.convert_timestamp_columns_in_df(df)
df = df.sort_values(['case:concept:name', 'time:timestamp'])
df = df.drop_duplicates(subset=['concept:name', 'case:concept:name', 'time:timestamp'], keep='last')
df = df.reset_index(drop=True)
seq_duplicates = []
lastIndex = len(df.index) - 1
for ind in df.index:
    if ind == lastIndex:
        break
    if (np.where(df['case:concept:name'][ind] == df['case:concept:name'][ind+1], True, False)
        and np.where(df['concept:name'][ind] == df['concept:name'][ind+1], True, False)):
        seq_duplicates.append(ind)
df = df.drop(seq_duplicates)
df = df.reset_index(drop=True)
df = df.sort_values('time:timestamp')

# Dataframe to EventLog
log = log_converter.apply(df)

# data processing by algorithm
# input - EventLog
# output - PetriNet, Initial marking, Final marking
print()
print("Running Alpha-miner..")
net, initial_marking, final_marking = alpha_miner.apply(log)

# vizualization of PetriNet
gviz = pn_visualizer.apply(net, initial_marking, final_marking)
pn_visualizer.view(gviz)

```

Рисунок 4.1.7 – Частина коду програми, що виконує обробку вхідних даних, запуск Alpha-miner алгоритму та візуалізацію результату

4.2 Підготовка вхідних тестових вхідних даних удосконаленого методу Alpha-miner для інформаційної системи волонтерської організації

Потрібно розглянути існуючу схему сховища даних та вибрати які саме дані можна експортувати для алгоритму.

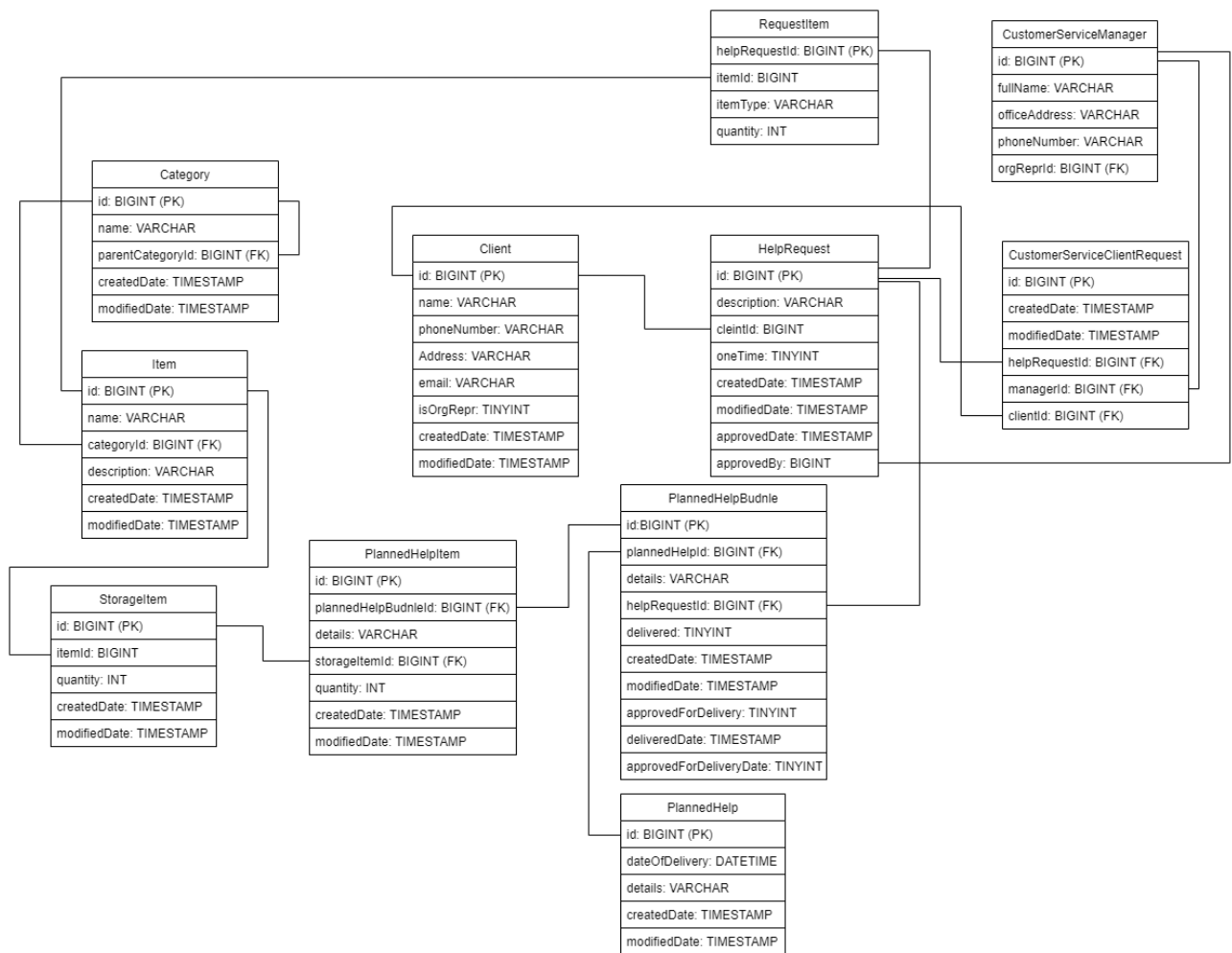


Рисунок 4.2.1 – Частина фізичної схеми БД інформаційної системи волонтерської організації

На рисунку 4.2.1 зображено частину фізичної схеми БД, що використовується інформаційною системою волонтерської організації. З цих даних можна зробити вибірку тих даних, що можна використовувати у вигляді журналу подій для процесної аналітики. У подальшому інтегруванні та використанні алгоритму можна робити біль масштабні та різноманітні вибірки та для демонстрації у даній роботі було визначено тестовий вигляд даних для аналізу. Для представлення Case ID вирішено використовувати “Client.id” та унікальний ідентифікатор заявки на допомогу “HelpRequest.id”. У якості подій створено такі позначки: “register help request” – подія створення заявки на допомогу, “register customer service procession” – подія реєстрації запиту до

менеджера по роботі з клієнтами, “approve help request” - підтвердження заявки на допомогу менеджером, “create planned help bundle” – створення плану на допомогу для клієнта, “add item to help bundle” – додати компонент допомоги до пакунку, “plan date of help bundle delivery” – планування дати доставки пакунку допомоги, “update description to help bundle” – додавання опису та приміток для доставки допомоги, “approve help bundle” – підтвердження готовності доставки, “help bundle delivered” – підтвердження доставки допомоги потребуючим. А також вирішено використовувати відповідні позначки часу (Timestamp) для подій: “HelpRequest.createdDate” – позначка часу створення заявки на допомогу, “CustomerServiceClientRequest.createdDate” – час запиту до сервісу по роботі з клієнтами, “HelpRequest.helpRequestApprovedDate” – час підтвердження заявки на допомогу, “PlannedHelpBundle.createdDate” – час створення плану на допомогу, “PlannedHelp.createdDate” – час створення дати доставки допомоги, “PlannedHelp.modifiedDate” – час оновлення опису та приміток для доставки допомоги, “PlannedHelpItem.createdDate” – час додавання предмету до пакунку допомоги “PlannedHelpBundle.approvedForDeliveryDate”- час підтвердження готовності доставки, “PlannedHelpBundle.deliveredDate” – час доставки допомоги потребуючим.

Результуючі дані повинні обов’язково мати заголовки: “case:concept:name” – Case ID, “concept:name” – Event та “time:timestamp” – Timestamp. На рисунку 4.2.2 зображений тестовий варіант журналу подій. Синім кольором виділено події, що повторюються один за одним (циклічні події) для ідентифікатора “5#1”. Жовтим кольором виділено циклічні події для ідентифікатора “2#1”. Червоним кольором позначено некоректні дубльовані події. Програма повинна знайти та вирішити усі ці помилки вхідного журналу подій.

case:concept:name	concept:name	time:timestamp
1#1	register help request	2022-12-01 14:32:15+01:00
1#1	approve help request	2022-12-01 14:35:22+01:00
3#1	register help request	2022-12-01 14:42:15+01:00
4#1	register help request	2022-12-01 14:42:15+01:00
5#1	register help request	2022-12-01 14:42:15+01:00
6#1	register help request	2022-12-01 14:42:15+01:00
3#1	approve help request	2022-12-01 14:50:22+01:00
4#1	approve help request	2022-12-01 14:50:22+01:00
5#1	approve help request	2022-12-01 14:50:22+01:00
6#1	approve help request	2022-12-01 14:50:22+01:00
2#1	register help request	2022-12-01 15:07:37+01:00
1#1	create planned help bundle	2022-12-01 15:08:52+01:00
3#1	create planned help bundle	2022-12-01 15:08:52+01:00
4#1	create planned help bundle	2022-12-01 15:08:52+01:00
5#1	create planned help bundle	2022-12-01 15:08:52+01:00
6#1	create planned help bundle	2022-12-01 15:08:52+01:00
2#1	register customer service procession	2022-12-01 15:09:01+01:00
4#1	add item to help bundle	2022-12-01 15:15:06+01:00
5#1	update description to help bundle	2022-12-01 15:15:06+01:00
6#1	add item to help bundle	2022-12-01 15:15:06+01:00
5#1	update description to help bundle	2022-12-01 15:16:06+01:00
3#1	update description to help bundle	2022-12-01 15:20:06+01:00
4#1	update description to help bundle	2022-12-01 15:20:06+01:00
5#1	add item to help bundle	2022-12-01 15:20:06+01:00
6#1	plan date of help bundle delivery	2022-12-01 15:20:06+01:00
2#1	approve help request	2022-12-01 15:29:01+01:00
1#1	plan date of help bundle delivery	2022-12-01 15:30:06+01:00
1#1	add item to help bundle	2022-12-01 15:39:06+01:00
1#1	update description to help bundle	2022-12-01 15:49:06+01:00
2#1	create planned help bundle	2022-12-01 15:50:52+01:00
2#1	plan date of help bundle delivery	2022-12-01 16:20:06+01:00
3#1	plan date of help bundle delivery	2022-12-01 16:20:06+01:00
4#1	plan date of help bundle delivery	2022-12-01 16:20:06+01:00
5#1	plan date of help bundle delivery	2022-12-01 16:20:06+01:00
6#1	update description to help bundle	2022-12-01 16:20:06+01:00
3#1	add item to help bundle	2022-12-01 16:25:06+01:00
1#1	approve help bundle	2022-12-01 16:50:06+01:00
2#1	update description to help bundle	2022-12-01 16:50:06+01:00
2#1	add item to help bundle	2022-12-01 16:55:06+01:00
3#1	approve help bundle	2022-12-01 16:55:06+01:00
4#1	approve help bundle	2022-12-01 16:55:06+01:00
5#1	approve help bundle	2022-12-01 16:55:06+01:00
6#1	approve help bundle	2022-12-01 16:55:06+01:00
2#1	add item to help bundle	2022-12-01 16:56:06+01:00
2#1	approve help bundle	2022-12-01 17:00:06+01:00
1#1	help bundle delivered	2022-12-01 19:10:06+01:00
1#1	help bundle delivered	2022-12-01 19:10:06+01:00
2#1	help bundle delivered	2022-12-01 19:20:06+01:00
3#1	help bundle delivered	2022-12-01 19:30:06+01:00
4#1	help bundle delivered	2022-12-01 19:40:06+01:00
5#1	help bundle delivered	2022-12-01 19:40:06+01:00
6#1	help bundle delivered	2022-12-01 19:40:06+01:00

Рисунок 4.2.2 - Тестовий варіант журналу подій

4.3 Експериментальна перевірка удосконаленого Alpha-miner методу

Спочатку користувач вносить дані для підключення до тестової БД. Після успішного підключення до БД виконується первинна обробка та конвертація даних для алгоритму. Підготовлені дані передаються для роботи Alpha-miner алгоритму. На рисунку 4.3.1 зображено введення даних користувачем та вивід на консоль звіту виконання етапів програми.

```
host: localhost
database: test
user name: root
password:
Connecting to db.
Connected.

Processing input data...

Running Alpha-miner...
```

Рисунок 4.3.1 - Введення даних користувачем та вивід на консоль звіту виконання етапів програми

Якщо користувач увів пароль з помилкою, то система відобразить повідомлення про заборону доступу до БД. На рисунку 4.3.2 зображено вивід на консоль невдалої спроби доступу до БД через невірний введений пароль користувача.

```

host: localhost
database: test
user name: root
password:
Connecting to db.
1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
Error in db conection. Try again

```

Рисунок 4.3.2 - Вивід на консоль невдалої спроби доступу до БД

Під час проведення експериментальної перевірки удосконаленого Alpha-miner методу додано логування стану журналу подій на різних етапах роботи методу. На рисунку 4.3.3 зображено стан журналу подій на початку роботи методу.

case:concept:name	concept:name	time:timestamp
1#1	register help request	2022-12-01 14:32:15+01:00
1#1	approve help request	2022-12-01 14:35:22+01:00
3#1	register help request	2022-12-01 14:42:15+01:00
4#1	register help request	2022-12-01 14:42:15+01:00
5#1	register help request	2022-12-01 14:42:15+01:00
6#1	register help request	2022-12-01 14:42:15+01:00
3#1	approve help request	2022-12-01 14:50:22+01:00
4#1	approve help request	2022-12-01 14:50:22+01:00
5#1	approve help request	2022-12-01 14:50:22+01:00
6#1	approve help request	2022-12-01 14:50:22+01:00
2#1	register help request	2022-12-01 15:07:37+01:00
1#1	create planned help bundle	2022-12-01 15:08:52+01:00
3#1	create planned help bundle	2022-12-01 15:08:52+01:00
4#1	create planned help bundle	2022-12-01 15:08:52+01:00
5#1	create planned help bundle	2022-12-01 15:08:52+01:00
6#1	create planned help bundle	2022-12-01 15:08:52+01:00
2#1	register customer service procession	2022-12-01 15:09:01+01:00
4#1	add item to help bundle	2022-12-01 15:15:06+01:00
5#1	update description to help bundle	2022-12-01 15:15:06+01:00
6#1	add item to help bundle	2022-12-01 15:15:06+01:00
5#1	update description to help bundle	2022-12-01 15:16:06+01:00
3#1	update description to help bundle	2022-12-01 15:20:06+01:00
4#1	update description to help bundle	2022-12-01 15:20:06+01:00
5#1	add item to help bundle	2022-12-01 15:20:06+01:00
6#1	plan date of help bundle delivery	2022-12-01 15:20:06+01:00
2#1	approve help request	2022-12-01 15:29:01+01:00
1#1	plan date of help bundle delivery	2022-12-01 15:30:06+01:00
1#1	add item to help bundle	2022-12-01 15:39:06+01:00
1#1	update description to help bundle	2022-12-01 15:49:06+01:00
2#1	create planned help bundle	2022-12-01 15:50:52+01:00
2#1	plan date of help bundle delivery	2022-12-01 16:20:06+01:00
3#1	plan date of help bundle delivery	2022-12-01 16:20:06+01:00
4#1	plan date of help bundle delivery	2022-12-01 16:20:06+01:00
5#1	plan date of help bundle delivery	2022-12-01 16:20:06+01:00
6#1	update description to help bundle	2022-12-01 16:20:06+01:00
3#1	add item to help bundle	2022-12-01 16:25:06+01:00
1#1	approve help bundle	2022-12-01 16:50:06+01:00
2#1	update description to help bundle	2022-12-01 16:50:06+01:00
2#1	add item to help bundle	2022-12-01 16:55:06+01:00
3#1	approve help bundle	2022-12-01 16:55:06+01:00
4#1	approve help bundle	2022-12-01 16:55:06+01:00
5#1	approve help bundle	2022-12-01 16:55:06+01:00
6#1	approve help bundle	2022-12-01 16:55:06+01:00
2#1	add item to help bundle	2022-12-01 16:56:06+01:00
2#1	approve help bundle	2022-12-01 17:00:06+01:00
1#1	help bundle delivered	2022-12-01 19:10:06+01:00
1#1	help bundle delivered	2022-12-01 19:10:06+01:00
2#1	help bundle delivered	2022-12-01 19:20:06+01:00
3#1	help bundle delivered	2022-12-01 19:30:06+01:00
4#1	help bundle delivered	2022-12-01 19:40:06+01:00
5#1	help bundle delivered	2022-12-01 19:40:06+01:00
6#1	help bundle delivered	2022-12-01 19:40:06+01:00

Рисунок 4.3.3 - Стан журналу подій на початку роботи методу

На рисунку 4.3.4 зображено стан журналу подій після обробки і сортування.

case:concept:name	concept:name	time:timestamp
1#1	register help request	2022-12-01 14:32:15+01:00
1#1	approve help request	2022-12-01 14:35:22+01:00
1#1	create planned help bundle	2022-12-01 15:08:52+01:00
1#1	plan date of help bundle delivery	2022-12-01 15:30:06+01:00
1#1	add item to help bundle	2022-12-01 15:39:06+01:00
1#1	update description to help bundle	2022-12-01 15:49:06+01:00
1#1	approve help bundle	2022-12-01 16:50:06+01:00
1#1	help bundle delivered	2022-12-01 19:10:06+01:00
1#1	help bundle delivered	2022-12-01 19:10:06+01:00
2#1	register help request	2022-12-01 15:07:37+01:00
2#1	register customer service procession	2022-12-01 15:09:01+01:00
2#1	approve help request	2022-12-01 15:29:01+01:00
2#1	create planned help bundle	2022-12-01 15:50:52+01:00
2#1	plan date of help bundle delivery	2022-12-01 16:20:06+01:00
2#1	update description to help bundle	2022-12-01 16:50:06+01:00
2#1	add item to help bundle	2022-12-01 16:55:06+01:00
2#1	add item to help bundle	2022-12-01 16:56:06+01:00
2#1	approve help bundle	2022-12-01 17:00:06+01:00
2#1	help bundle delivered	2022-12-01 19:20:06+01:00
3#1	register help request	2022-12-01 14:42:15+01:00
3#1	approve help request	2022-12-01 14:50:22+01:00
3#1	create planned help bundle	2022-12-01 15:08:52+01:00
3#1	update description to help bundle	2022-12-01 15:20:06+01:00
3#1	plan date of help bundle delivery	2022-12-01 16:20:06+01:00
3#1	add item to help bundle	2022-12-01 16:25:06+01:00
3#1	approve help bundle	2022-12-01 16:55:06+01:00
3#1	help bundle delivered	2022-12-01 19:30:06+01:00
4#1	register help request	2022-12-01 14:42:15+01:00
4#1	approve help request	2022-12-01 14:50:22+01:00
4#1	create planned help bundle	2022-12-01 15:08:52+01:00
4#1	add item to help bundle	2022-12-01 15:15:06+01:00
4#1	update description to help bundle	2022-12-01 15:20:06+01:00
4#1	plan date of help bundle delivery	2022-12-01 16:20:06+01:00
4#1	approve help bundle	2022-12-01 16:55:06+01:00
4#1	help bundle delivered	2022-12-01 19:40:06+01:00
5#1	register help request	2022-12-01 14:42:15+01:00
5#1	approve help request	2022-12-01 14:50:22+01:00
5#1	create planned help bundle	2022-12-01 15:08:52+01:00
5#1	update description to help bundle	2022-12-01 15:15:06+01:00
5#1	update description to help bundle	2022-12-01 15:16:06+01:00
5#1	add item to help bundle	2022-12-01 15:20:06+01:00
5#1	plan date of help bundle delivery	2022-12-01 16:20:06+01:00
5#1	approve help bundle	2022-12-01 16:55:06+01:00
5#1	help bundle delivered	2022-12-01 19:40:06+01:00
6#1	register help request	2022-12-01 14:42:15+01:00
6#1	approve help request	2022-12-01 14:50:22+01:00
6#1	create planned help bundle	2022-12-01 15:08:52+01:00
6#1	add item to help bundle	2022-12-01 15:15:06+01:00
6#1	plan date of help bundle delivery	2022-12-01 15:20:06+01:00
6#1	update description to help bundle	2022-12-01 16:20:06+01:00
6#1	approve help bundle	2022-12-01 16:55:06+01:00
6#1	help bundle delivered	2022-12-01 19:40:06+01:00

Рисунок 4.3.4 - Стан журналу подій після обробки і сортування

На рисунку 4.3.5 зображено стан журналу подій після видалення некоректних вхідних даних. Можна побачити, що після проведеного етапу в журналі не залишилось циклічних подій, подій що повторюються одна за одною, а також некоректних дубльованих даних.

case:concept:name	concept:name	time:timestamp
1#1	register help request	2022-12-01 14:32:15+01:00
1#1	approve help request	2022-12-01 14:35:22+01:00
1#1	create planned help bundle	2022-12-01 15:08:52+01:00
1#1	plan date of help bundle delivery	2022-12-01 15:30:06+01:00
1#1	add item to help bundle	2022-12-01 15:39:06+01:00
1#1	update description to help bundle	2022-12-01 15:49:06+01:00
1#1	approve help bundle	2022-12-01 16:50:06+01:00
1#1	help bundle delivered	2022-12-01 19:10:06+01:00
2#1	register help request	2022-12-01 15:07:37+01:00
2#1	register customer service procession	2022-12-01 15:09:01+01:00
2#1	approve help request	2022-12-01 15:29:01+01:00
2#1	create planned help bundle	2022-12-01 15:50:52+01:00
2#1	plan date of help bundle delivery	2022-12-01 16:20:06+01:00
2#1	update description to help bundle	2022-12-01 16:50:06+01:00
2#1	add item to help bundle	2022-12-01 16:56:06+01:00
2#1	approve help bundle	2022-12-01 17:00:06+01:00
2#1	help bundle delivered	2022-12-01 19:20:06+01:00
3#1	register help request	2022-12-01 14:42:15+01:00
3#1	approve help request	2022-12-01 14:50:22+01:00
3#1	create planned help bundle	2022-12-01 15:08:52+01:00
3#1	update description to help bundle	2022-12-01 15:20:06+01:00
3#1	plan date of help bundle delivery	2022-12-01 16:20:06+01:00
3#1	add item to help bundle	2022-12-01 16:25:06+01:00
3#1	approve help bundle	2022-12-01 16:55:06+01:00
3#1	help bundle delivered	2022-12-01 19:30:06+01:00
4#1	register help request	2022-12-01 14:42:15+01:00
4#1	approve help request	2022-12-01 14:50:22+01:00
4#1	create planned help bundle	2022-12-01 15:08:52+01:00
4#1	add item to help bundle	2022-12-01 15:15:06+01:00
4#1	update description to help bundle	2022-12-01 15:20:06+01:00
4#1	plan date of help bundle delivery	2022-12-01 16:20:06+01:00
4#1	approve help bundle	2022-12-01 16:55:06+01:00
4#1	help bundle delivered	2022-12-01 19:40:06+01:00
5#1	register help request	2022-12-01 14:42:15+01:00
5#1	approve help request	2022-12-01 14:50:22+01:00
5#1	create planned help bundle	2022-12-01 15:08:52+01:00
5#1	update description to help bundle	2022-12-01 15:16:06+01:00
5#1	add item to help bundle	2022-12-01 15:20:06+01:00
5#1	plan date of help bundle delivery	2022-12-01 16:20:06+01:00
5#1	approve help bundle	2022-12-01 16:55:06+01:00
5#1	help bundle delivered	2022-12-01 19:40:06+01:00
6#1	register help request	2022-12-01 14:42:15+01:00
6#1	approve help request	2022-12-01 14:50:22+01:00
6#1	create planned help bundle	2022-12-01 15:08:52+01:00
6#1	add item to help bundle	2022-12-01 15:15:06+01:00
6#1	plan date of help bundle delivery	2022-12-01 15:20:06+01:00
6#1	update description to help bundle	2022-12-01 16:20:06+01:00
6#1	approve help bundle	2022-12-01 16:55:06+01:00
6#1	help bundle delivered	2022-12-01 19:40:06+01:00

Рисунок 4.3.5 - Стан журналу подій після видалення некоректних вхідних даних

Дані з рисунку 4.3.5 форматуються у зрозумілий для Alpha-miner алгоритму вигляд та передаються для роботи алгоритму.

Результуючу модель процесів волонтерської організації, що побудована удосконаленим Alpha-miner алгоритмом з тестового журналу подій зображено на рисунках 4.3.6(А) - 4.3.6(Б).

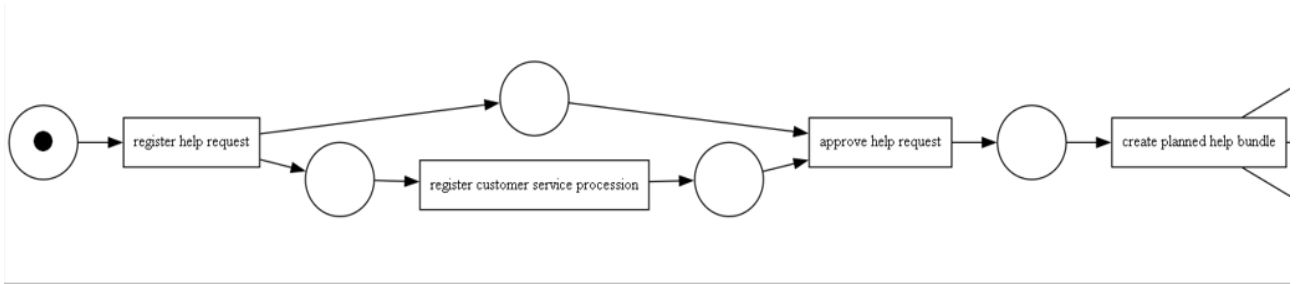


Рисунок 4.3.6(А) – Модель процесів волонтерської організації, що побудована удосконаленим Alpha-miner алгоритмом

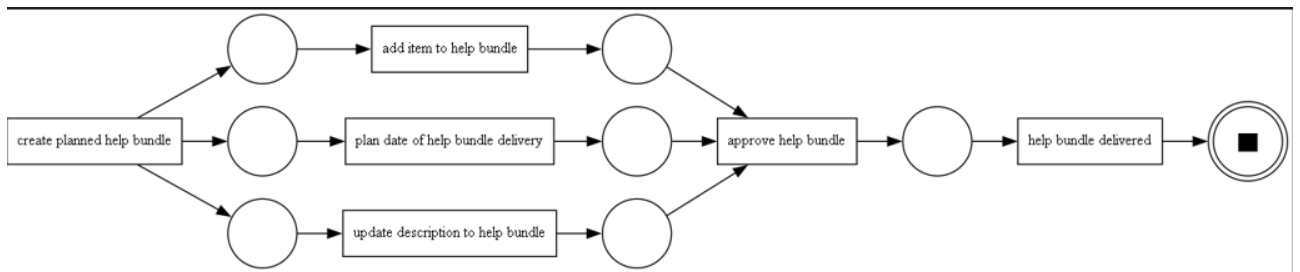


Рисунок 4.3.6(Б) – Модель процесів волонтерської організації, що побудована удосконаленим Alpha-miner алгоритмом

На вихідній моделі процесів волонтерської організації можна побачити, що незважаючи на помилкові події, які представлені у журналі подій, схема являється коректною та виконана за правилами Workflow nets.

Для оцінки та порівняльного аналізу результату роботи реалізованого удосконаленого Alpha-miner методу було побудовано модель процесів, що є результатом виконання існуючого стандартного Alpha-miner алгоритму. Використано журнал подій зображений на рисунку 4.2.2 з усіма циклічними подіями, а також подіями, що дублюються.

Результуючу модель процесів волонтерської організації, що побудована стандартним Alpha-miner алгоритмом з тестового журналу подій зображено на рисунках 4.3.7(А) - 4.3.7(Б).

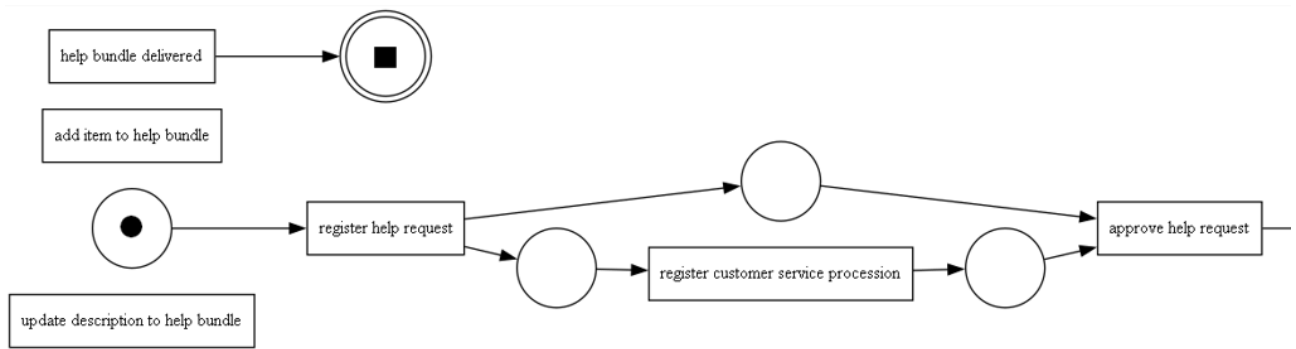


Рисунок 4.3.7(А) – Модель процесів волонтерської організації, що побудована стандартним Alpha-miner алгоритмом

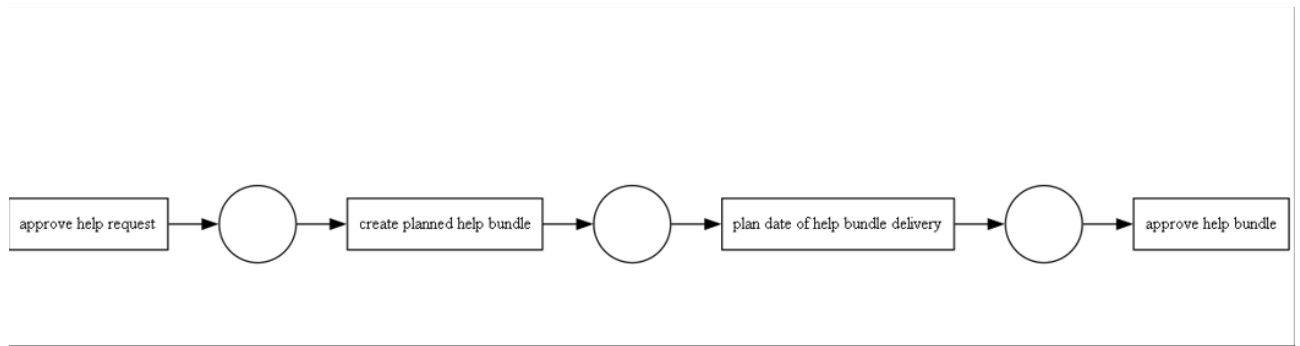


Рисунок 4.3.7(Б) – Модель процесів волонтерської організації, що побудована стандартним Alpha-miner алгоритмом

На рисунку 4.3.7(А) можна побачити, що події, які були присутні у некоректних даних журналу подій, побудовані з помилками. Такі помилки виникають через те, що Alpha-miner неправильно інтерпретує циклічні події, а також події, що повторюються.

В таблиці 4.3.1 зображено порівняння роботи удосконаленого та стандартного Alpha-miner методу. Модель, що побудована за допомогою стандартного методу не відповідає правилам WF. Удосконалений метод виконав аналіз тестового журналу подій без помилок.

Таблиця 4.3.1 – Порівняння роботи удосконаленого та стандартного Alpha-miner методу

	Удосконалений Alpha-miner метод	Стандартний Alpha-miner метод
Усього подій у журналі	10	10
Кількість відображених подій на результуючій схемі	10	10
Кількість вірно розташованих подій	10	7
Визначений початковий стан	Так	Так
Визначений кінцевий стан	Так	Так
Кожна окрема подія перебуває між початковим та кінцевим станом	Так	Ні

ВИСНОВКИ

У результаті виконання магістерської роботи був виконаний огляд методів Process Mining в задачах побудови і удосконалення бізнес процесів волонтерської організації. Були проаналізовані існуючі бізнес процеси волонтерської організації. У результаті аналізу запропоновано виконати етап моделювання з життєвого циклу управління процесами. Проведено порівняльний аналіз існуючих методів інтелектуального аналізу процесів - Process Mining. Обрано Alpha-miner метод для побудови моделі бізнес процесів волонтерської організації. Удосконалено Alpha-miner метод шляхом реалізації додаткового первинного етапу синхронізації даних зі сховищем ІС, попередньої обробки вхідних даних, видалення помилок, сортування та конвертації у зрозумілий алгоритму формат.

У результаті створення програмного модулю удосконаленого Alpha-miner алгоритму волонтерська організація отримала можливість побудови поточної моделі фактичних процесів без створення додаткової системи для синхронізації та первинної обробки даних. Аналіз та удосконалення поточної моделі процесів організації дасть змогу підвищити ефективність існуючих бізнес-процесів, виявити слабкі місця, а також може бути використана для побудови нових компонентів системи.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлювання. – Чинний від 22.06.2015. – Київ: ДП «УкрНДНЦ», 2016. – 31 с.
2. ДСТУ 8302:2015. Інформація та документація. Бібліографічні посилання. Загальні положення та правила складання. – Чинний від 04.03.2016. – Київ: ДП «УкрНДНЦ», 2016. – 20 с.
3. Chalyi S., Levykin I., Biziuk A., Vovk A., Bogatov I. Development of the technology for changing the sequence of access to shared resources of business processes for process management support. *Eastern-European Journal of Enterprise Technologies*, 2020. Vol 2, NO 3 (104). С. 22-29.
4. Chalyi S., Bogatov Ie. Method of constructing an attribute description of the business process "as is" in the process approach to enterprise management EUREKA: *Physics and Engineering*. 2018. Vol. 6. P. 35-40.
5. Чалий С.Ф., Богатов Є.О. Упорядкування трас логу на основі порівняння атрибутів подій в задачі побудови моделей бізнес-процесів засобами. *Process mining Materials of the VII International Scientific Conference «Information-Control System and Technologies» 17th-18th September, 2018, Odessa*. С.152-154.
6. Чалий С.Ф., Кузьма Є.А., *Process Mining – Інструмент менеджмента бізнес процесів* Матеріали X-ої Ювілейної Міжнародної науково-практичної конференції «Free and Open Source Software», Харків, 20-22 листопада 2018 р. – Харків: Харківський національний університет будівництва та архітектури, 2018. С.75.
7. Chalyi S., Levykin I., Petrychenko A., Bogatov I. (2018). Causality-based model checking in business process management tasks. *IEEE 9th International Conference on Dependable Systems, Services and Technologies DESSERT'2018*
8. Чалий С.Ф., Левикин І.В. Концепція двоконтурного управління множини наскрізних бізнес-процесів на основі прецедентного підходу Поліграфічні,

мультимедійні та web-технології. Матеріали III Міжнародної науково-технічної конференції (17–19 жовтня 2018 р.). — Львів : Українська академія друкарства, 2018. —С.

9. Левикін В. М., Чала О.В. Оцінювання часових характеристик подій дискретних процесів у відповідності до концепції GUM. Метрологія та прилади. 2015. № 6 (56). С.19-23.

10. Чала О.В. Формалізація неявних процедурних залежностей в знання-ємних бізнес-процесах. Наукові праці Вінницького національного технічного університету. 2016. № 4. С. 43-47.

11. Чала О.В. Еволюційний підхід до управління життєвим циклом знання-ємних бізнес-процесів. Наукоємні технології. 2017. № 1(33). С. 53-59.

12. Левикін В. М., Чала О.В. Підхід до виявлення аномальної поведінки процесів в системах процесного управління на основі аналізу логів. Вісник Національного технічного університету «ХПІ». Системний аналіз, управління та інформаційні технології. 2017. № 55(1276). С. 77-81.

13. Chala O.V., Chaliy S.F. Situational script management of business processes with changeable structure. International conference: “ITA 2008” Varna, Bulgaria. N 3. Vol. 2. 2008. P. 62-66.

14. Левыкин В. М., Чалая О.В. Застосування методів інтелектуального аналізу процесів для виявлення патернів поведінки динамічних об'єктів. Проблеми розвитку глобальної системи зв'язку, навігації, спостереження та організації повітряного руху CNS/ATM: Тези доп. наук.-техн. конф. (Київ, 21-23 листоп. 2016). Київ: НАУ, 2016. С. 67.

15. Introduction to Process Mining. URL: <https://towardsdatascience.com/introduction-to-process-mining-5f4ce985b7e5>

16. Alpha miner. URL: <https://www.futurelearn.com/info/courses/process-mining/0/steps/15636>

17. ProM Tips — Which Mining Algorithm Should You Use?. URL: <https://www.fluxicon.com/blog/2010/10/prom-tips-mining-algorithm/>

18. Public repository for the PM4Py (Process Mining for Python) project. URL: <https://github.com/pm4py/pm4py-core>
19. Process Management. URL: <https://www.integrify.com/landing-pages/process-management/>
20. State-of-the-art-process mining in Python. URL: <https://pm4py.fit.fraunhofer.de/>
21. What is Process Management? URL: <https://kissflow.com/workflow/bpm/what-is-process-management/>
22. Gitlab documentation. URL: <https://docs.gitlab.com/ee/> ()
23. Business Process Management Definitions. URL: <https://www.comidor.com/knowledge-base/business-process-management-kb/business-process-management-bpm/>
24. PROCESS MANAGEMENT – strategy, organization. URL: <https://www.referenceforbusiness.com/management/Or-Pr/Process-Management.html>
25. Alpha algorithm – Wikipedia. URL: https://en.wikipedia.org/wiki/Alpha_algorithm
26. What is Data Flow Diagram? URL: <https://www.visual-paradigm.com/guide/data-flow-diagram/what-is-data-flow-diagram/>
27. What is DFD (Data Flow Diagram)? URL: <https://www.geeksforgeeks.org/what-is-dfddata-flow-diagram/>
28. IDEF - Definition, Methods, and Benefits. URL: <https://www.edrawsoft.com/what-is-idef.html>
29. The Complete Guide To Understand IDEF Diagram. URL: <https://www.edrawmax.com/article/the-complete-guide-to-understand-idef-diagram.html>
30. IDEF0 standard with ConceptDraw DIAGRAM. URL: <https://www.conceptdraw.com/How-To-Guide/idef0>
31. Hoang Huy Nguyen. (2019). Stage-Aware Business Process Mining, pp. 220.

32. Kunal Gupta, Astha Sachdev, Ashish Sureka. (2017) Empirical Analysis on Comparing the Performance of Alpha Miner Algorithm in SQL Query Language and NoSQL Column-Oriented Databases Using Apache Phoenix.
33. Thatchanok Wiriyaratnakul (2016). Combination Usage of Process Mining and Adaptive Case Management.
34. Hoogendoorn, G.E.. (2017). A comparative study for process mining approaches in a real-life environment.
35. Методичні вказівки щодо розробки та оформлення кваліфікаційної роботи (для студентів усіх форм навчання другого (магістерського) рівня програми "Інформаційні управляючі системи та технології") / Упоряд.:Петров К.Е., Левикін В.М., Чалий С.Ф., Євланов М.В., Саєнко В.І., Міхнов Д.К., Міхнова А.В., Чала О.В. – Харків: ХНУРЕ, 2021. – 30с.