

Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____
(повна назва)

Кафедра _____ Інформаційних управляючих систем _____
(повна назва)

АТЕСТАЦІЙНА РОБОТА **Пояснювальна записка**

рівень вищої освіти _____ другий (магістерський) _____
(рівень вищої освіти)
(позначення документа)

_____ Дослідження методів _____
_____ кластеризації трас журналів подій _____
_____ в проектах інтелектуального аналізу процесів _____
(тема)

Виконав: студент 2 курсу, групи УПГІТм-18-1 _____
Волошин М.С. _____
(прізвище, ініціали)

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

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

Освітня програма Управління проектами в галузі _____
інформаційних технологій _____
(повна назва освітньої програми)

Керівник д.т.н., проф. каф. ІУС Чалий С.Ф. _____
(посада, прізвище, ініціали)

Допускається до захисту
Зав. каф. ІУС

(підпис)

Петров К.Е.
(прізвище, ініціали)

2019 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наукКафедра Інформаційних управляючих системРівень вищої освіти другий (магістерський)Спеціальність 122 – Комп'ютерні науки
(код і повна назва)Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)Освітня програма Управління проектами в галузі інформаційних технологій
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедрою _____
(підпис)

«___» _____ 2019 р.

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ

студентові Волошину Микиті Сергійовичу
(прізвище, ім'я, по батькові)1. Тема роботи Дослідження методів кластеризації трас журналів подій в проектах інтелектуального аналізу процесівзатверджена наказом по університету від 31.10 2019р. № 1588Ст2. Термін подання студентом роботи до екзаменаційної комісії 13.12. 2019 р.3. Вихідні дані до роботи Журнали подій, методи кластеризації журналів подій, методи дослідження багатоваріантних моделей процесів, моделі та методи представлення процесів.4. Перелік питань, що потрібно опрацювати в роботі Аналіз сучасних методів аналізу журналів подій та моделей процесів; Дослідження методів кластеризації журналів подій; Сучасні методи дослідження моделей процесів та моделей конфігурованих процесів; Апробація запропонованих рішень у програмі для алгоритмів дослідження процесів

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на атестаційну роботу магістра	29.03.2019	
2	Попереднє дослідження галузі завдання	29.03.2019 – 06.04.2019	
3	Аналіз вхідних даних роботи	07.04.2019 – 16.04.2019	
4	Аналіз існуючих алгоритмів кластеризації	17.04.2019 – 25.04.2019	
5	Дослідження та опис обраного алгоритму кластеризації	25.04.2019 – 08.05.2019	
6	Аналіз методів дослідження конфігурованих процесів	09.05.2019 – 22.05.2019	
7	Вдосконалення методі дослідження конфігурованих процесів	23.05.2019 – 20.06.2019	
8	Опис та реалізація вдосконаленого методу	21.06.2019 – 12.07.2019	
9	Оформлення пояснювальної записки та графічного матеріалу	10.12.2019 – 12.12.2019	
10	Подання студентом роботи для перевірки на плагіат	13.12.2019	

Дата видачі завдання __ _____ 2019 р.

Студент _____
(підпис)

Керівник роботи _____ д.т.н, проф. каф. ІУС Чалий С.Ф.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Реферат атестаційної роботи магістра: 88 с., 29 рис., 4 табл., 10 формул, 22 джерела.

ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ПРОЦЕСІВ, ЖУРНАЛ ПОДІЙ, ТРАСИ, КЛАСТЕРИЗАЦІЯ, КРАСТЕРИЗАЦІЯ ТРАС, ДОСЛІДЖЕННЯ ПРОЦЕСІВ, ДЕРЕВО ПРОЦЕСІВ, КОНФІГУРОВАНІЙ ПРОЦЕС

Об'єктом дослідження є процес формування моделей бізнес процесів «як є» за технологією process mining .

Предметом дослідження є методи кластеризації трас журналів подій у проектах інтелектуального аналізу процесів.

Мета роботи полягає у Удосконалення методу кластеризації трас на основі побудови ієрархії дій процесу.

Наукова новизна полягає у удосконалення методів побудови моделей багатоваріантних процесів на основі аналізу логів шляхом ієрархічної кластеризації трас з подальшим виділення окремих конфігурацій таких процесів.

ABSTRACT

Abstract on master thesis: 88 p., 29 pic., 4 tables, 10 formulas, 22 sources

INTELLECTUAL ANALYSIS OF PROCESS, EVENT LOG, TRACE, CLUSTERING, TRACE CLUSTERING, PROCESS, PROCESS TREE, CONFIGURABLE PROCESS DISCOVERY.

The object of the study is the process of forming business process models "as is" by process mining technology.

The subject of the research is methods of clustering of event log trails in projects of intellectual analysis of processes.

The purpose of the work is to improve the method of clustering routes based on the construction of a hierarchy of process actions.

The scientific novelty is to improve the methods of constructing models of multivariate processes based on log analysis by hierarchical clustering of trails, followed by the isolation of individual configurations of such processes.

ЗМІСТ

Перелік скорочень і термінів.....	7
Вступ.....	8
1. Аналіз предметної галузі та постановка задачі.....	10
1.1 Дослідження характеристик проектів інтелектуального аналізу процесів.....	10
1.2 Дослідження структури журналу подій.....	11
1.3 Дослідження методів аналізу процесів.....	13
1.3.1 Аналіз процесів.....	14
1.3.2 Розпізнавання процесу.....	15
1.3.3 Перевірка відповідності.....	16
1.3.4 Покращення моделі.....	17
1.4 Постановка задачі.....	19
2. Кластеризація трас та дослідження конфігурованих процесів.....	20
2.1 Дослідження конфігурованих процесів.....	28
2.1.1 Вирівнювання розмітки.....	36
2.1.2 Приховання та блокування.....	36
2.1.3 Пониження операторів.....	38
2.1.4 Оцінка дерев конфігуративних процесів.....	41
2.2 Удосконалення методу побудови моделі багатоваріантного процесу	45
3. Побудова плану розробки проекту.....	47
3.1 Розробка плану проекту.....	53
3.2 Дослідження ПП на симуляцію запізнення виконання задач.....	59
4. Реалізація удосконаленого методу кластеризації трас логу.....	62
Висновки.....	69
Перелік джерел посилання.....	70
Додаток А. Графічний матеріал.....	74

ПЕРЕЛІК СКОРОЧЕНЬ І ТЕРМІНІВ

BPMN – Business Process Model and Notation

ETM – Evolutionary Tree Miner

ETMd – ETM для дерева процесів із одного журналу подій

ETMc – ETM для дерева процесів із декількох журналів подій

PERT – Project Evaluation and Review Technique

CPM – Critical-Path Method

БП – Бізнес -процес

ІАП – Інтелектуальний Аналіз подій

ПП – План проекту

ВСТУП

Існуючі техніки дослідження конфігурованих процесів виконують дослідження моделі на організаційному рівні, тобто напряду з журналів подій (логів) тих самих організацій. У цій роботі запропоновано підхід, який дозволяє проводити дослідження моделі на поведінковому рівні, тобто до уваги приймається тільки спостережувана поведінка на трасах, а не її оригінал. Увесь підхід складається з трьох кроків.

Перший полягає у тому, щоб знайти групи трас, схожі за поведінкою. Для того, щоб це зробити, використано ієрархічну кластеризацію трас. Алгоритми кластеризації частіше за все потребують вимір несхожості, тоді як існуючі підходи для вимірювання несхожості не включають в себе відповідність моделі, що приводить до складностей із порядком виконання та циклічними процедурами. У даній роботі запропоновано підхід семантичного міркування, новий вимір несхожості, який долає вищевказані проблеми за допомогою відносного дерева процесів.

Другим кроком є надання кінцевому користувачу можливості розуміння роботи внутрішніх процесів нашого підходу та пояснення, чому траси об'єднані в групи (згруповані). Ми робимо це з даними трас подіями у таких групах. У цій роботі розглянуто, як описувати ієрархічну кластеризацію за допомогою анотацій даних використовуючи алгоритми наївного Баєса або алгоритм C4.5 для класифікатора дерев рішень. У сучасному повсякденному житті ієрархічна кластеризація може зазнати широкого застосування. Існує багато методів, які спрямовані на усунення ієрархії, однак вони не беруть до уваги анотації даних. У цій роботі запропоновано алгоритм, який проводить зведення за допомогою опису даних. Іноді немає можливості описати данні у ієрархічній кластеризації із використанням даних з журналу подій, тому що журнал подій не містить в собі дані, які описують поведінку. Для вирішення цієї проблеми журнал подій можна поповнити атрибутами даних, які абстраговані від трас.

Третім кроком є створення конфігурованої моделі процесів із обраних груп трас у ієрархічному кластеруванні. У цій роботі увагу сфокусовано на деревах процесів. До написання даної роботи єдиним шляхом дослідження дерев процесів був підхід «грубої сили». У даній роботі приведений більш структурний підхід, який дозволяє досліджувати конфігурації будь яких процесів та добірок журналів подій. Підхід, який наведений у даній роботі, продемонстровано за допомогою журналу подій автошколи, який спрямований на удосконалення роботи автошкіл та може бути легко імплементований у будь які умови іншої предметної області.

Отриманим результатом роботи є удосконалений метод кластеризації трас журналів подій для отримання моделей конфігуративних процесів..

Отримані результати виконання магістерської атестаційної роботи оформлені у вигляді пояснювальної записки.[1]

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Дослідження характеристик проектів інтелектуального аналізу процесів

Через збільшення використання інформаційних систем збільшується запис даних. Більшість цих даних неструктурована та складна для пояснення. Одним з основних завдань є вилучення інформації із цих сирих даних та визначення їх цінності [2]. Ціллю аналізу даних є отримання інформації із сирих даних. Часто інформаційні системи виконують процеси та записані дані описують події цього процесу. Аналіз процесів представляє собою добування (вилучення) інформації про процес організації, використовуючи в основному записані процеси. Аналіз процесів поділяється на три основні види діяльності: дослідження, перевірка відповідності та вдосконалення. Це і є завдання дослідження нової моделі процесу на основі записаних подій – перевірити, як записані події відповідають моделі процесу чи наданій низці правил, та вдосконалення існуючих моделей процесів на основі записаних даних [2]. У цій роботі ми загострюємо увагу на дослідженні процесів.

В області дослідження процесів виконання процесу називається позначається як траса. Добірка трас називається журналом подій. Журнали подій можуть бути вилучені з існуючих інформаційних систем самих організацій. Для підтримки швидкої міграції з існуючих систем на хмарні рішення дослідження процесів може бути використане для пошуку конфігурованої моделі процесу, яка підтримує кожен з муніципалітетів. Існуючі техніки досліджують конфігуровану модель процесу із налаштуваннями для кожного вхідного журналу подій, через що конфігурації досліджуються для кожного муніципалітету. Такий підхід досліджує конфігурації на організаційному рівні. У цій роботі запропоновано інший підхід, який дозволяє проводити дослідження на поведінковому рівні, тобто ми спостерігаємо тільки видиму поведінку на трасах, а не їх початок. Дослідження на поведінковому рівні вважає добірку журналів подій, які ми

отримали з муніципалітетів, як один журнал подій. Для дослідження конфігурованої моделі процесу нам потрібно розділення (сегрегація) трас. Для досягнення розділення ми визначаємо траси, які однакові за поведінкою, тобто однакові дії були зроблені під час виконання процесу, та згрупувати їх. Далі ми надаємо пояснення кінцевому користувачу, чому ці групи були сформовані за даними трас та подіями в групах. Виконуючи це, бачимо, як дані пов'язані зі поведінкою за якою спостерігаємо. І зараз можливо автоматично або за допомогою кінцевого користувача зробити добірку цікавих груп, для яких досліджується конфігурована модель процесу.

1.2 Дослідження структури журналу подій

Журнал подій – це ключове поняття в області дослідження процесів.[2] Журнал подій містить у собі добірку трас, а траси містять у собі послідовність подій (діяльності). Кожен журнал подій, траса або подія можуть містити у собі атрибути даних. Атрибути даних події має містити хоча би опис типу виконуваної діяльності та також має містити іншу інформацію, таку як інформація про ресурси та часові рамки. Кожний тип діяльності також називається класом подій.

У таблиці 1.1 показано дві траси робочого прикладу. Для кожної траси вказано: метод оплати, студент, інструктор, екзаменатор, кількість відвіданих занять, кількість незаліків студента. Таблиці 1.2 відображає події цих трас. Скомбіновані дані цих таблиць утворюють журнал події.

Таблиця 1.1 Дві траси робочого прикладу з атрибутами даних

№ траси	Атрибут				Кількість занять	Кількість незаліків	...
	Оплата	Студент	Інструктор	Екзаменатор			
221	Абонемент	Артур	Сергій	Вікторія	3	0	...
285	Після класу	Марія	Сергій	Микола	2	1	...
...

Таблиця 1.2 Дві траси робочого прикладу з атрибутами даних

№ траси	№ події	Атрибути			Ресурс	...
		Часова мітка	Дія			
221	41251	02.05.2014 13:05	Початок(A)		Микола	...
	41252	02.05.2014 13:12	Придбати зняття + Екзамен(B)		Микола	...
	41257	02.05.2014 13:31	Заняття в класі(C)		Сергій	...
	41258	02.05.2014 14:32	Потрібно ще(E)		Сергій	...
	41261	05.05.2014 15:14	Заняття в класі (C)		Сергій	...
	41262	05.05.2014 16:17	Потрібно ще (E)		Сергій	...
	41269	07.05.2014 13:01	Теоретичний Екзамен(F)		Сергій	...
	41271	10.05.2014 14:59	Заняття в класі (C)		Сергій	...
	41280	15.05.2014 15:57	Здати Практичний Екзамен(G)		Вікторія	...
	41295	21.05.2014 14:07	Отримати водійське посвідчення(I)		Микола	...
258	53151	21.05.2014 14:07	Початок(A)		Микола	...
	53151	21.05.2014 14:07	Заняття в класі (C)		Микола	...
	53151	21.05.2014 14:07	Сплатити за заняття (D)		Сергій	...
	53151	21.05.2014 14:07	Потрібно ще(E)		Сергій	...
	53151	21.05.2014 14:07	Заняття в класі (C)		Сергій	...
	53151	21.05.2014 14:07	Потрібно ще (D)		Сергій	...
	53151	21.05.2014 14:07	Теоретичний Екзамен(F)		Сергій	...
	53151	21.05.2014 14:07	Заняття в класі (G)		Сергій	...
	53151	21.05.2014 14:07	Здати Практичний Екзамен(J)		Вікторія	...
...	

1.3 Дослідження методів аналізу процесів

Методи аналізу процесів дозволяють автоматично моделювати процеси за журналами подій, перевіряти коректність моделей та вдосконалювати їх.

Інструменти бізнес – аналізу (BI, business intelligence) допомагають приймати рішення на основі даних о подіях. Він складається з ВАРМ (моніторинг бізнес – активності), СРМ (моніторинг ефективності підприємства), СРІ (неперервне вдосконалення процесів) та ВРІ (аналіз бізнес – процесів). Лише деякі інструменти бізнес – аналізу надають можливості вилучення даних, але вони не є процесно – орієнтованими, а призначені тільки для вилучення інформації, яка допомагає у прийнятті тактичних рішень і не охоплюють сам процес.[3]

Інструменти управління бізнес – процесами (BPM, business process management) дозволяють за допомогою відповідних моделей аналізувати поточні та плановані процеси, але ці моделі частіше за все не співпадають з реальними даними.[5] Виходячи з цього, результати аналізу ненадійні, оскільки засновані на ідеалізованій моделі, а не на фактах які реально спостерігаються

Задача аналізу процесів (process mining) – заповнити простір між ВІ та ВРМ шляхом одночасного використання даних про події, які зберігаються у журналах подій та моделях процесів.[9] На відміну від традиційних підходів, ціллю аналізу процесів є не одиничне конструювання статичної моделі. Аналітиками будується динамічну мінливу карту процесу на основі актуальних даних для того, щоб робити прогнози і відповідати на конкретні питання.

1.3.1 Аналіз процесів

Початком аналізу процесів є отримання журналу подій [4]. Кожна подія порівнюється дією (чітко визначеним етапом деякого процесу) та відноситься до конкретного прецеденту (трасі). Події кожного прецеденту упорядковані та описують одну ітерацію процесу. У журналах можуть зберігатися інші дані, пов'язані з подіями. Це можуть бути додаткові відомості (наприклад, про людину, пристрій або запускаючу дію), часові відмітки та інші атрибути і показники процесу.

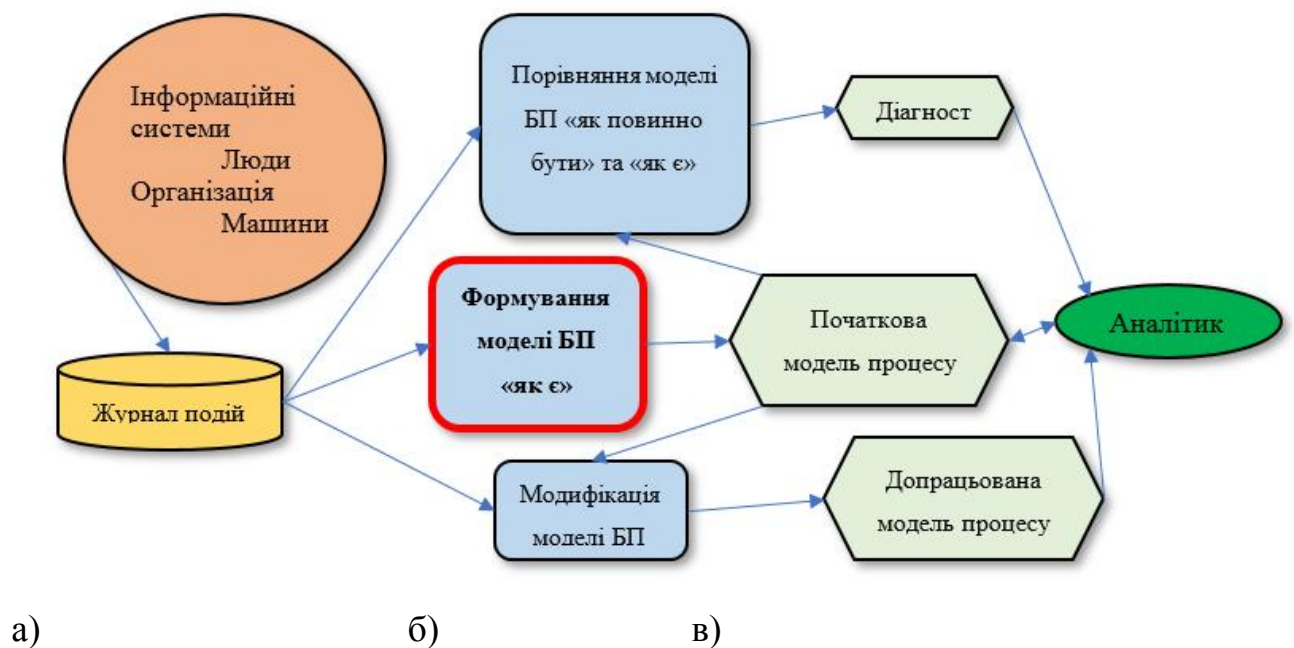


Рисунок 1.1 – Аналіз процесу: а) За допомогою алгоритму розпізнавання процесу за журналом подій будується початкова модель; б) На етапі перевірки відповідності виявляються розбіжності між спостережуваною та модельованою поведінкою; в) Початкова модель вдосконалюється із використанням даних про події.

Аналіз процесів складається з трьох етапів. На першому етапі аналітик за допомогою методів розпізнавання процесу (process discovery) відновлює модель із

використанням журналу подій.[4] Також цю початкову модель можна побудувати вручну. Далі за допомогою механізмів перевірки відповідності (conformance checking) визначаються розбіжності між журналом подій та початковою моделлю процесу. На етапі покращення моделі (model enhancement) за допомогою збережених у журналі показників процесу [8] виконується покращення й доповнення моделі. Наприклад за відмітками часу до моделі можна додати інформацію про часові рамки (час очікування та обслуговування) Отриману допрацьовану модель процесу можна застосувати для прийняття рішень.[3]

1.3.2 Розпізнавання процесу

Події відносяться до екземплярів процесу, як, наприклад транзакція до заказу покупця. Усі події, які відносяться до певного екземпляру процесу (прецеденту), можна упорядкувати. Так як ці події також відносяться до дій, то кожен прецедент представити у вигляді послідовності дій – трас.

Наприклад процес містить у собі дії А, Б, В, Г та Д, які можуть бути розміщені у наступних послідовностях: АБВД, АВБД та АГД. Припустимо, що журнал подій записаний для цього процесу, містить у собі інформацію про 238 прецедентів з послідовністю АБВД, 56 прецедентах з послідовністю АВБД та 88 прецедентах з послідовністю АГД. Алгоритм розпізнавання процесів можуть перетворити цей журнал подій у модель процесу, яка описує спостережувану поведінку. У даному випадку досить легко створити процес, який завжди починається з дії А, закінчується дією Д, а між ними відбуваються дії Б та В (у будь – якій послідовності), або Г.

Розпізнавання моделі процесу, яка складається із десятків та навіть сотен дій, набагато складніше. На Рисунку 1.2 наведено фрагмент великого журналу подій процесу обробки страхового випадку та автоматичного побудована за ним

модель у вигляді блок схеми у нотації BPMN (Business Process Modeling Notation). На її прикладі є можливість спостереження паралельних дій, розгалуження, цикли та інші конструкції потоку управління, ідентифіковані із послідовності у наведеному журналі подій [8].

Процес, наведений на Рисунку 1.2 завжди починається з дії «реєстрація запиту» та закінчується виплатою відшкодування або відмовою. Дія «вивчення» та «перевірка квитанції» відбуваються одночасно. Вивчення буває двох типів. Рішення про відшкодування або відмову приймається в залежності від результату перевірок. Також буває, що прийняти рішення не є можливим. Дія «повторний запит» знову запускає дію перевірки, тим самим створюючи цикл.

В останні роки була розроблена велика кількість алгоритмів розпізнавання процесів для вивчення моделей із широко кола журналів подій у самих різних областях діяльності.

1.3.3 Перевірка відповідності

Моделі процесів можуть бути описуючими (які відображають реальні події) та нормативними (які задають те, що повинно відбуватись). Їх можна розпізнати шляхом аналізу процесів або будувати вручну. У будь-якому випадку бажано порівнювати модель з реальністю, яка відображена у журналі подій.

Виявляти розбіжності між змодельованою та спостережуваною поведінкою допомагають алгоритми перевірки відповідності. Вони видають показники ступеня відповідності та діагностичні відомості, пояснюючі спостережувані розбіжності. З їхньою допомогою можна детально аналізувати прецеденти, які не є у складі побудованої моделі [24].

Механізми перевірки відповідності допомагають аналітикам оцінювати якість розпізнавання моделей процесів, але важніше за все є те, що ці механізми застосовуються як допоміжна технологія при аудиторських перевірках,

оптимізації процесів за методом Six Sigma та контролю виконання нормативних актів.

1.3.4 Покращення моделі

На етапі покращення моделі аналітик доповнює початкову модель процесу, використовуючи даними журналу подій. Як показано на Рисунку 1.2, відмітки часу причинно зв'язаних подій можна використовувати для оцінки тривалості дій. Наприклад, аналіз відміток часу може показати, що в середньому на прийняття рішення після перевірки квитанції потрібно 21 день. Отримані дані допомагають виявити вузькі місця та прогнозувати строки виконання тих чи інших етапів процесу.

Якщо протокол подій містить відомості про ресурси, то є можливість з'ясувати ролі, розподілення робочих обов'язків та інші характеристики. Аналіз додаткових атрибутів подій та траси (прецеденту) дозволяє зрозуміти правила прийняття рішень, пояснюючі варіанти вибору: наприклад, траси, в яких перевірку виконує Сью, зазвичай закінчуються відмовою.

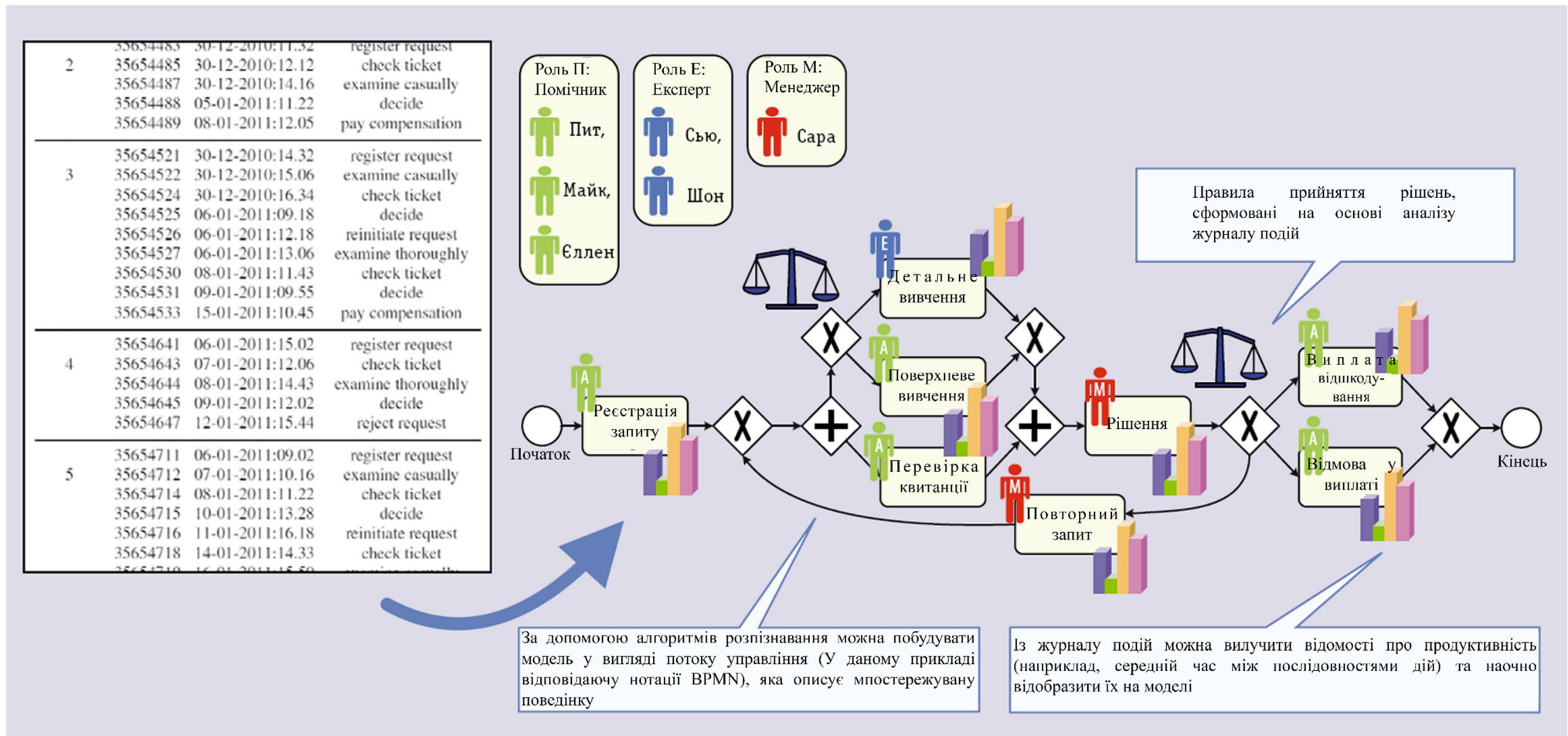


Рисунок 1.2 – Приклад аналізу процесу із застосуванням журналу подій

1.4 Постановка задачі

Для досягнення поставленої мети в роботі були вирішені такі задачі:

- Аналіз предметної галузі;
- Дослідження характеристик проектів ІАП;
- Дослідження структури журналу подій;
- Дослідження методів аналізу процесів;
- Аналіз та розпізнавання БП, перевірка відповідності до моделі;
- Удосконалення алгоритму кластеризації трас подій;
- Планування розробки модулю ієрархічної кластеризації трас;
- Практична реалізація алгоритму.

2. КЛАСТЕРИЗАЦІЯ ТРАС ТА ДОСЛІДЖЕННЯ КОНФІГУРОВАНИХ ПРОЦЕСІВ

Кластер – це колекція об'єктів даних містить схожі об'єкти в одному кластері. Це означає, що об'єкти є аналогічними один до одного в межах однієї групи, і в той же час вони досить різні, або пов'язані з об'єктами в іншій групі або в інших кластерах. Кластерний аналіз також називають кластеризацією або сегментацією даних. Кластерний аналіз розподіляє даний набір точок даних в набір кластерів або груп. Ці точки даних якомога більше схожі в межах однієї групи та віддалені наскільки це можливо від інших груп. Кластерний аналіз відноситься до навчання без вчителя (unsupervised learning) з огляду на те, що на початку немає визначених класів. Це суттєво відрізняє його від класифікації, де потребується навчання з учителем (supervised learning) або завдання міток класу для побудови моделі класифікації

Існує дві основні класифікації алгоритмів кластеризації:

- Ієрархічні і неієрархічні (плоскі). Ієрархічні алгоритми будують систему вкладених розбиттів, тобто на виході алгоритму представляється дерево кластерів, з коренем у якості всієї вибірки і листками – у якості найменших кластерів. Неієрархічні алгоритми будують лише одне розбиття об'єктів на кластери.

- Чіткі і нечіткі. Чіткі алгоритми надають всім об'єктам вибірки відповідний номер кластера, що означає, що кожен об'єкт повинен відноситися лише до одного кластеру.

Нечіткі алгоритми надають кожному об'єкту у відповідність набір значень, які демонструють ступінь належності об'єкта до кластерів. Отже, кожен об'єкт відноситься до кожного кластеру з певною ймовірністю.

Ієрархічний метод кластеризації – це така кластеризація за якої, починаючи з кластера, що складається з одного елемента, кластери ітеративно зливаються у в кластери вищого рівня. Також можливо починати з єдиного великого

макрокластера, який ітеративно розділяється на маленькі кластери. Таким чином формуються ієрархія кластерів. Для їх формування не потрібно задавати кількість кластерів K , такий тип кластеризації є більш детермінованим та не потребує ітеративних уточнень. Ієрархічні методи кластеризації включають в себе дві категорії алгоритмів. Перша категорія має назву агломераційної. Вона починається з одноелементного кластера, що зливаються два кластери, щоб побудувати ієрархію кластерів «знизу вгору». Друга група – Divisive methods – методом розділення великий макрокластер, що містить всі елементи, розділяється на дві групи, кожна з них також на дві групи і так далі. Таким чином генерується ієрархія кластерів «зверху вниз». На рисунку зображено обидва підходи.

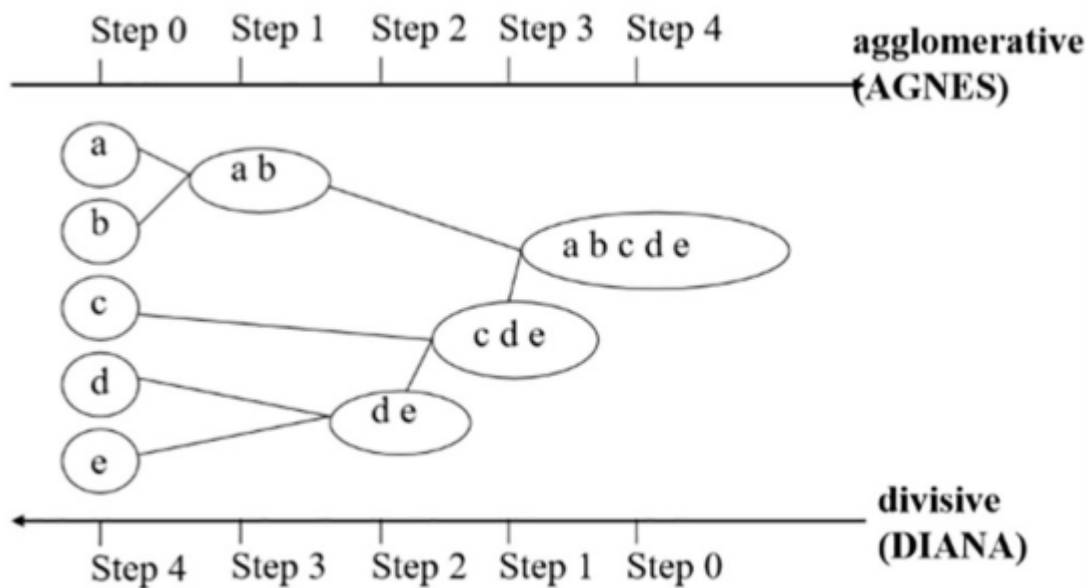


Рисунок 2.1 – Хід алгоритмів AGNES, DIANA

Агломеративні алгоритми – це такі, що кластеризують «знизу вгору». На початку алгоритму кожна точка розглядається як кластер, потім алгоритм намагається об'єднати найближчі сусідні точки в один більший кластер і так далі, щоб зрештою об'єднати всі кластери в один великий кластер. Агломеративні

алгоритми також називають AGNES (AGglomerative NESting). Хід алгоритму виглядає наступним чином:

- Використовується метод одноканального зв'язку «найближчий сусід» та матриця відмінностей;
- Вузли, що мають найменші відмінності, зливаються;
- Всі вузли об'єднуються в один кластер.

Агломеративна кластеризація залежить від використання мір подібності кластерів:

- Одноканальний зв'язок (найближчий сусід);
- Повний зв'язок (діаметр);
- Середній зв'язок (середнє по групі);

Одноканальний зв'язок (найближчий сусід):

- Подібність двох кластерів - це подібність між їх найбільш подібними членами (найближчий сусід);
- Приділяється увага найближчим точкам, ігнорується структура кластера;
- Можливість будувати кластери неправильної форми;
- Такий вид зв'язку чутливий до даних з шумами та значень, що вибиваються з множини. Схематичне зображення наведено на Рисунку 2.2.

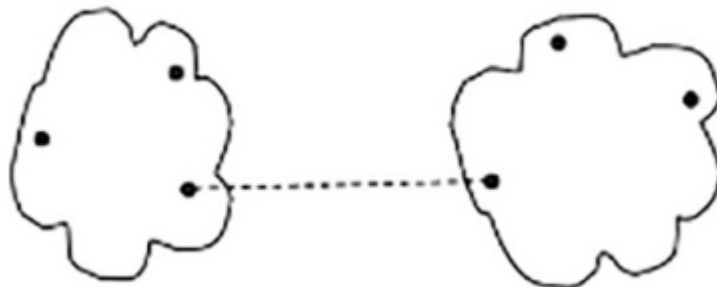


Рисунок 2.2 – Одноканальний зв'язок

Повний зв'язок:

- Подібність двох кластерів рахується як подібність їх найменш подібних членів;
- Два кластери об'єднуючись формують кластер з щонайменшим діаметром; На виході – кластери компактної форми;
- Чутливий до значень, що суттєво відрізняються. Схематичне зображення наведено на рисунку 2.3.

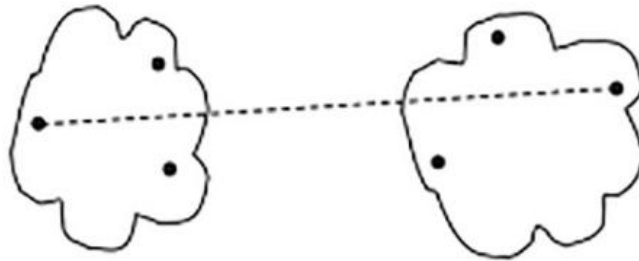


Рисунок 2.3 – Повний зв'язок

Середній зв'язок – середня відстань між елементами в парі кластерів, що зображено на Рисунку 2.4. Особливістю є затратне обчислення.

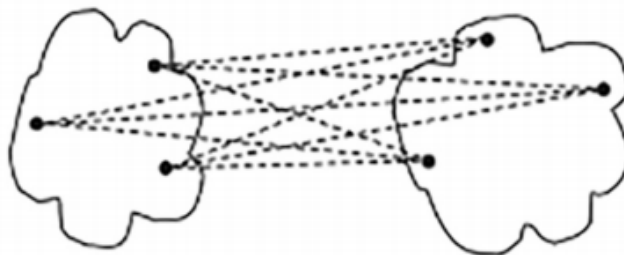


Рисунок 2.4 – Середній зв'язок

Кластеризація трас є задачею знаходження однорідних груп трас. Кластеризація трас відрізняється від звичайної кластеризації тільки вхідними даними. Ними є журнал подій, який і є добіркою трас. Кожна траса у журналі подій є об'єктом у задачі кластеризації. Одним з перших обговорювань значення кластеризації трас у дослідженні процесів було у Medeiros et al. [7]. Там було

зазначено, що алгоритми дослідження процесів краще за все працювали на структурованих моделях з невеликою кількістю шумів, але зазвичай це є ідеальним, тобто недосяжним випадком. Зазвичай дуже складно визначити обсяг процесу та дуже часто у ньому дуже багато шумів.[6] Через це моделі процесів мають дуже заплутаний вигляд. Кластеризація трас призначена для вирішення цієї проблеми. Завдяки кластеризації журналу подій ми можемо отримати більш однорідні групи трас. Це веде до більш зрозумілої моделі процесів під час дослідження. У цій роботі спостерігається подібна проблема. Потрібно знайти групи однорідних трас, завдяки чому ми можемо створити адекватні конфігурації моделі процесів для цих груп.

Виходячи з проблеми, яка розглядається у роботі, припустимо, що можна розташувати траси журналу подій у двомірному просторі. Результат буде схожий на те, що представлено на Рисунку 2.5. Після застосування кластеризації трас ми отримуємо два кластера, що показано границями різного кольору навколо об'єктів.

Більшість алгоритмів кластеризації використовують вимірювання розбіжності. Вимірювання розбіжності є важливим фактором у ідентифікації об'єктів, що повинні бути згруповані в кластери. Існує багато методів для визначення виміру розбіжності для кластеризації трас, але жоден з них не додає змісту моделі процесів. Однак це може значно покращити продуктивність кластеризації. Саме тому у цій роботі запропоновано новий підхід, Підхід Синтаксичного Обґрунтування, який містить в собі відповідність до моделі.

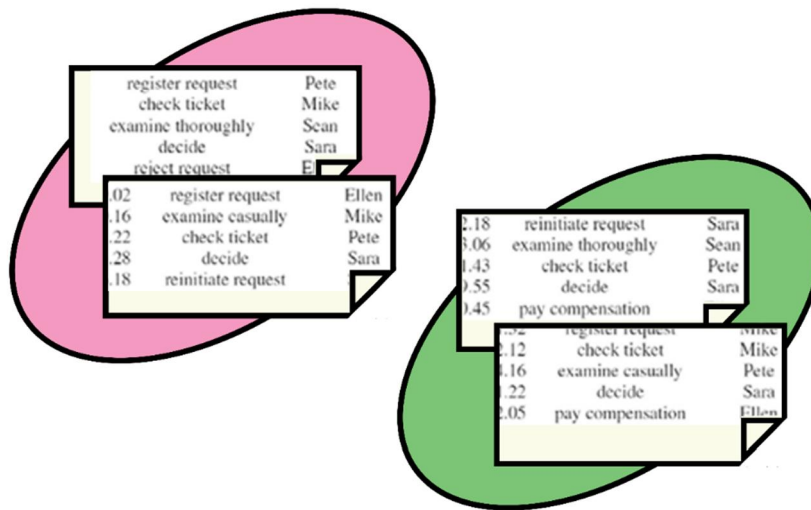


Рисунок 2.5 – Приклад кластеризації трас, що утворює кластери

Кластеризація дуже важлива у дослідженні процесів, завдяки знаходженню однорідних груп трас складність досліджуваних моделей значно знижується. Кластеризація трас застосовується для знаходження груп трас, які описують конфігурації моделі процесів. Алгоритми кластеризації використовують одне з вимірювань несхожості та багато алгоритмів кластеризації мають багато недоліків у визначенні несхожості через проблеми з відповідністю до моделі. У даній роботі розглядається нова міра несхожості, підхід синтаксичної змістовності. За допомогою додавання відповідності моделі, що дає змогу знайти у журналі подій паралельне виконання та цикли. Результатом буде ієрархічна кластеризація, яка показує сегригацію вхідних журналів подій до колекції дерев процесів.

Дерево процесів [10] є одним багатьох видів опису моделювання процесів, який може бути використаний для представлення моделей процесів. Іншими прикладами нотацій процесного моделювання є: Мережі Петрі [2], Модель та Нотація Бізнес-процесів (BPNM) [11], Подійний Ланцюг Процесів (EPC) [12], та багато інших. Однак для цих мов лише незначна частина всіх можливих моделей

звучить, тобто вони не містять безвихідних ситуацій та інших проблем. На дерева процесів ці умови не впливають.

Дерева процесів – це дерева, які містять вузли – оператори та вузли – дії (процеси). Кожен листок дерева є вузлом – процесом, всі інші вузли є вузлами – операторами. Вузли – оператори позначають відношення між своїми нащадками. Існує шість типів операторів: послідовність (\rightarrow), паралельне виконання (\wedge), виключний вибір (\times), вибір (\vee), повторне виконання (\cup), та зворотна послідовність (\leftarrow). Усі оператори, окрім \cup - оператора, можуть мати одного або декількох нащадків. \cup - оператор завжди має мати три нащадки, перший відповідає за виконувану дію, другий відповідає за повтор цієї дії, а третій відповідає за вихід із циклу. Вузли – процеси дерева процесів можуть також позначатися літерою τ , що відображає ту дію, яка не спостерігається.

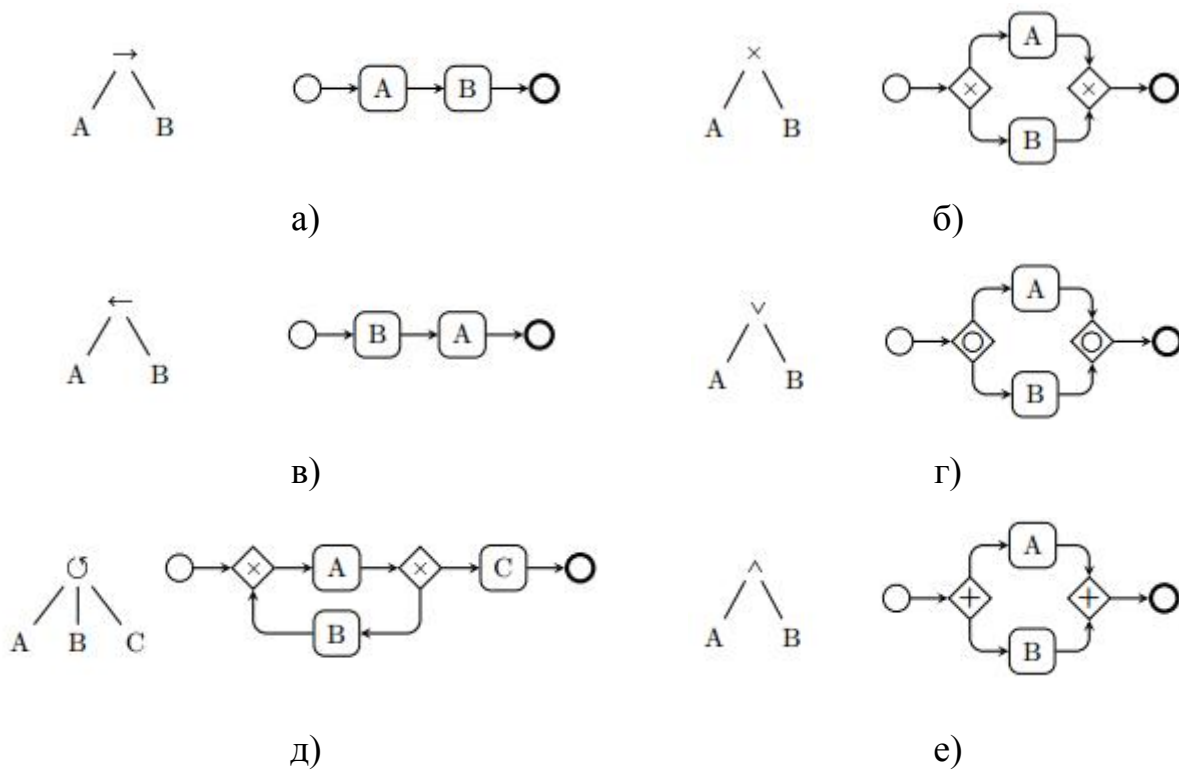


Рисунок 2.6 – Оператори дерева процесів та їх представлення у вигляді моделі BPMN.

На даний момент існує два алгоритми дослідження дерева процесів за допомогою журналу подій, а саме ETMd та Індуктивний Майнер [13]. ETMd являє собою еволюційний алгоритм та пояснюється далі у підрозділі 2.6. Індуктивний Майнер – це структурний підхід до дослідження дерева процесів.

Конфігуровані дерева процесів [14] описують добірку варіантів моделі. Конфігуроване дерево процесу є одним з багатьох налаштованих нотацій моделювання процесів [15, 16, 17]. У більшості з них основними опціями конфігурації є: «дозволити» (allow), «приховати» (hide) та «заблокувати» (block). Опція «дозволити» означає, що вузол має дозвіл на виконання. «приховати» означає, що вузол можна пропустити при виконанні, «заблокувати» означає, що у вузла немає дозволу на виконання та не може бути пропущеним. Вузли у моделі процесів можуть бути позначені за допомогою цих опцій, таким чином отримується конфігурована модель процесів. Добірка опцій формує конфігурацію. Для отримання виконуваної моделі, конфігурацію слід застосувати до моделі.

Конфігуровані дерева процесів підтримують три опції для кожного вузла дерева. Однак, якщо усі нащадки вузла заблоковані або є єдиним нащадком вузла \wedge , \leftarrow або \rightarrow , або якщо нащадок нащадка дії або виходу з вузла циклу заблокований, це означає, що увесь вузол заблокований. Окрім цих опцій конфігуроване дерево процесів підтримує концепт Пониження Операторів. Оператор може бути понижений тільки до більш обмежуючого оператора. На Рисунку 2.7 показано ієрархію пониження, оператори зверху є менш обмежуючими, навідрізу, від тих, що нижче. Деревя процесів можуть обмеженими тільки за допомогою конфігуративних опцій. Існують нотації конфігуративного процесного моделювання, які підтримують розширення моделі процесів [18]. Однак вони не визначені для дерев процесів та не використовуються у даній роботі.

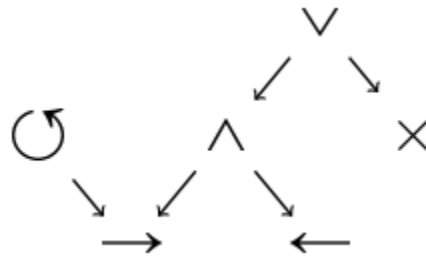


Рисунок 2.7 – Ієрархія пониження операторів дерева процесів

Надалі у цій роботі конфігуративні опції зображуються як сірі виклики. Наприклад якщо виклик містить: $[-, H, V, \rightarrow]$. Це показує, що у першій конфігурації вузол має бути дозволеним, у другій вузол має бути прихованим, у третій вузол має бути блокованим а для четвертої конфігурації вузол має бути понижений до \rightarrow - оператора.

2.1 Дослідження конфігурованих процесів.

Конфігуративні моделі процесів надають можливість працювати з варіативними процесами. Така модель може поєднувати різні варіанти одного і того ж процесу, що дозволяє кінцевому користувачу конфігурувати процес у найзручніший для них варіант. Виходячи з того, як анотувати ієрархічний кластер даними про траси та подіями, бажані кластери можуть бути обрані вручну або автоматично визначені. Також кожен обраний кластер є групою трас, для якої ми намагаємось знайти оптимальну конфігурацію. Для демонстрації прикладу ми обираємо три групи трас: C_1 , C_2 та C_3 , які приведено у Таблиці 2.1. На Рисунку 2.8а зображене дерево процесів, яке використовується як вхідні данні для алгоритму дослідження процесів. Далі, на Рисунку 2.8б зображену дерево процесу, подібно до якого ми бажаємо отримати після роботи алгоритму.

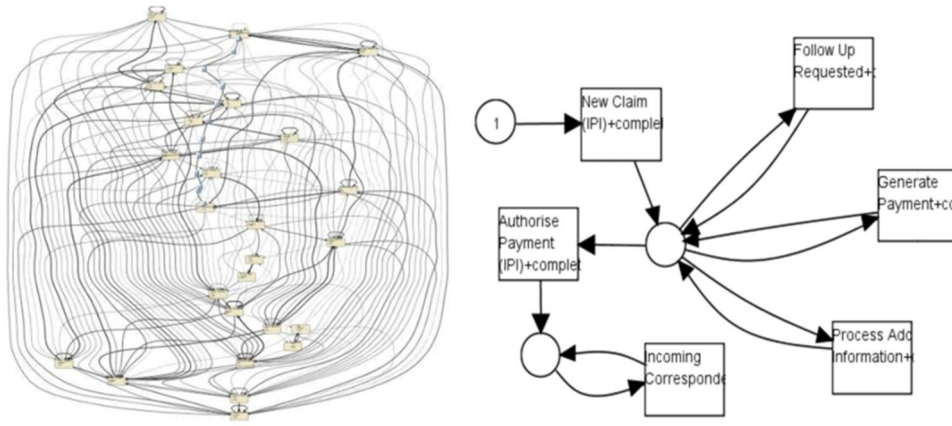


Рисунок 2.8 – Порівняння «спагеті-графа» та квіткової моделі процесу

Таблиця 2.1 – Групи та траси, які містяться у даних групах

Індекс	Траса	Кластер
t_1	$\langle A, B, C, E, C, E, C, F, E, C, G, I \rangle$	C_1
t_2	$\langle A, C, D, E, C, D, F, E, C, D, G, I \rangle$	C_2
t_3	$\langle A, D, F, C, E, D, C, G, H, D, C, E, D, C, E, D, C, G \rangle$	C_3
t_4	$\langle A, D, C, E, D, C, F, E, D, C, E, D, C, G \rangle$	C_3

Для отримання такої моделі аналітику потрібно проводити глибокий аналіз логу для знаходження та визначення подібних процесів. Такий підхід займає дуже багато часу і не є продуктивним. Якщо спробувати обробити такий лог програмними методами, ми отримаємо дуже громіздку та заплутану модель процесів, так званий «спагеті-граф», зображений на Рисунку 2.8. Такий граф потребує подальшого аналізу, у ході якого буде отримано дерево процесів, представлене на Рисунку 2.9а. Це буде одинична конфігурація процесу, для кожного варіанту потрібно будувати нову модель. Запропонований підхід дозволяє отримати модель конфігурований процесів з анотаціями вузлів, як показано на Рисунку 2.9б.

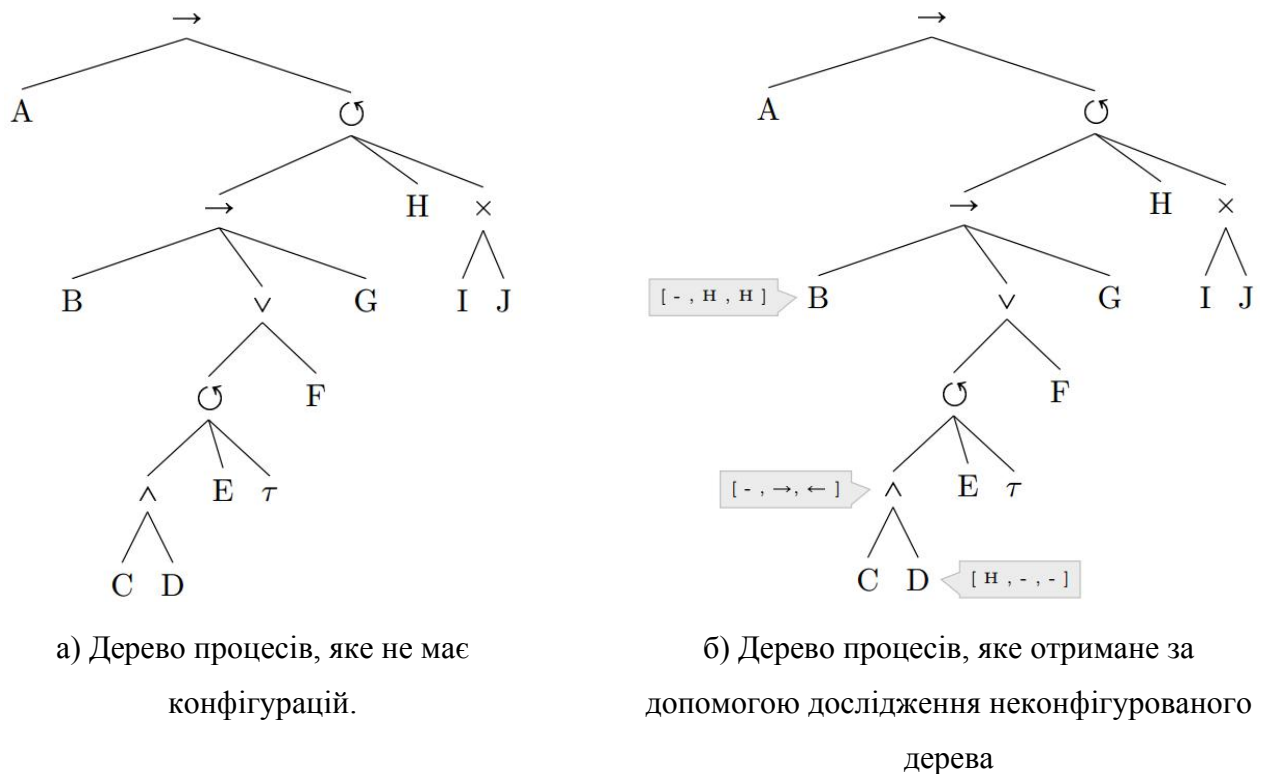


Рисунок 2.9 – Дерева процесів без конфігурації (а) та після (б) застосування алгоритму

На Рисунку 2.9б біля деяких вузлів дерева процесів містяться анотації, які і є конфігураціями, які можна застосувати до цього вузла. Кожна анотація містить у собі три варіанти, кожен з яких належить до свого кластеру відповідно C_1 , C_2 або C_3 . Анотація біля дії B має вигляд $[-, H, H]$, це означає, що для кластеру C_1 вузол є відкритим, а для C_2 та C_3 вузол є прихованим. Анотація біля дії \wedge , що означає паралельне виконання подальшої дії з вузлом E , має вигляд $[-, \rightarrow, \leftarrow]$, це означає, що для кластеру C_1 вузол є відкритим, для кластеру C_2 виконання наступних дій, та для кластеру C_3 виконання наступних дій має відбуватися зворотно послідовно. Анотація біля дії D має вигляд $[H, -, -]$, це означає, що для кластеру C_1 вузол (дія) є прихованою, а для C_2 та C_3 вузол є відкритим.

У деяких випадках замість одної вхідної моделі ми можемо мати велику кількість реферальних моделей. Дослідження дерева за допомогою еволюційного алгоритму дослідження процесів подібного ETMd [19] може

виробляти безліч моделей, які спеціалізуються на другорядних аспектах. Іншою причиною може бути те, що завдяки придбанням, поглинанням чи злиттям ми отримуємо декілька моделей процесів для подібних процесів [20]. Об'єднання цих технологічних моделей в єдину еталонну модель не є тривіальним і може призвести до декількох об'єднаних моделей. За допомогою нашого підхідного методу виявлення процесів ми можемо створити конгуруючий варіант для будь-якої з цих моделей. Однак не банально, яка з цих моделей найкраща. Відповідаючи на четверте дослідницьке запитання, ми вирішуємо це питання, надаючи засіб для оцінювання дерева, що складається з технологічних процесів, за показниками якості.

Модель конфігурованих процесів може бути досліджена різними шляхами. Кожен починається з колекції журналів подій. Ці чотири підходи є наступними, їх представлено на Рисунках, 2.10, 2.11, 2.12 та 2.13.

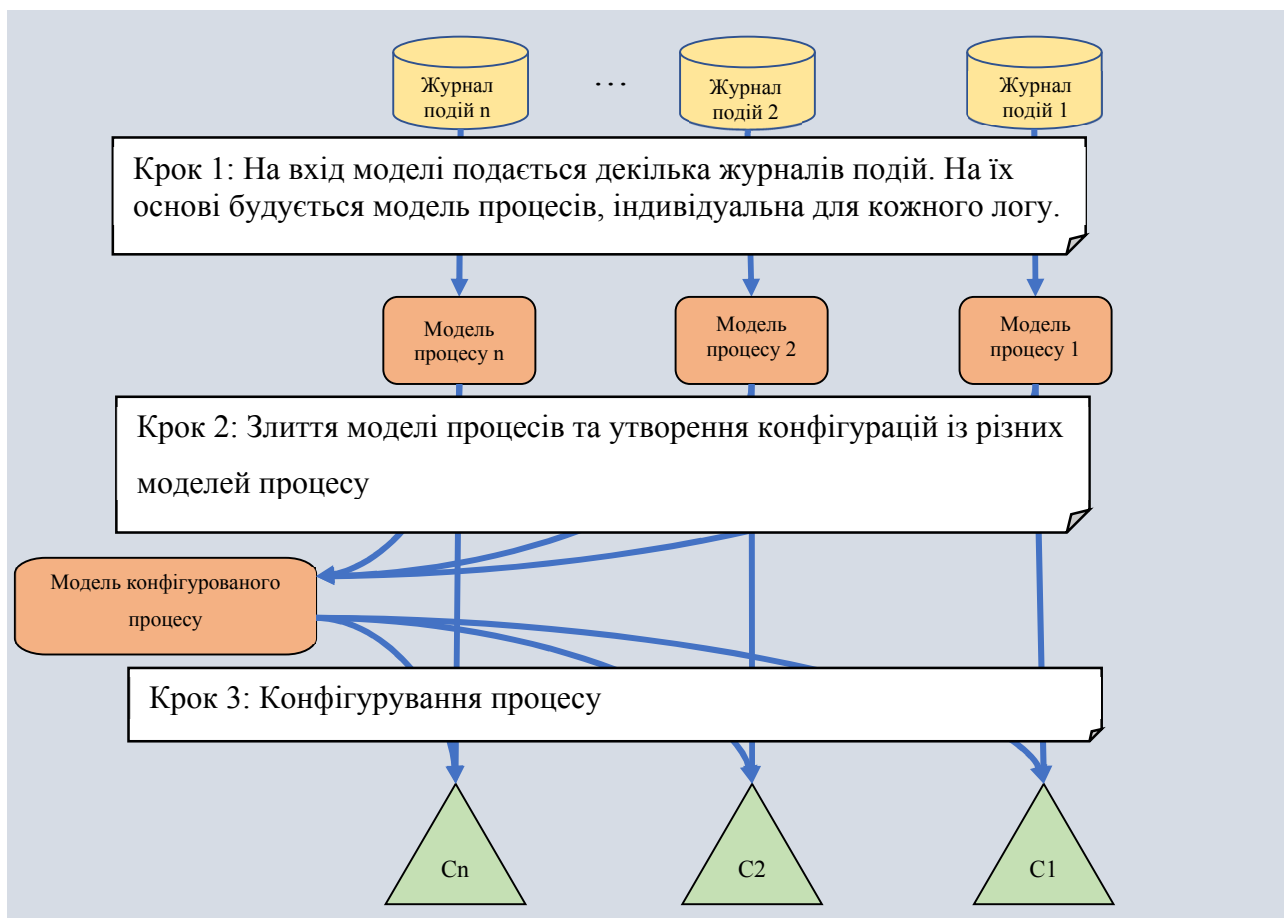


Рисунок 2.10 – Перший підхід

Даний підхід досліджує модель процесу за допомогою усіх вхідних журналів подій. На даному кроці моделі процесів зливаються для отримання єдиної моделі конфігурованого процесу. Однак злиття моделей процесів не є простою задачею. Існує багато методів злиття. Головною проблемою цього підходу є те, що моделі процесів не можуть бути порівняні правильно, алгоритм злиття пропонує простий вибір між вхідними моделями.

Другий підхід намагається покращити перший підхід тим, що спершу відбувається злиття журналів подій. Далі із утвореного журналу подій досліджується модель процесу. Далі спільна модель процесу індивідуалізується згідно з журналами подій. Після цього ці індивідуальні моделі процесів зливаються у єдину модель конфігуративних процесів. Головна ідея полягає у тому, щоб спершу створити єдину модель процесів і потім вивести з них індивідуальні моделі, це дозволяє легше звести їх у модель конфігуративних процесів

Третій підхід працює наступним чином: єдина модель процесів описує поведінку усіх журналів подій. Далі для кожного журналу подій досліджується така конфігурація моделі процесів, яка найкраще підходить до журналу подій. У цьому підході єдина модель процесів є менш точною, так як вона обмежена конфігурацією, але не може бути обмеженою.

Четвертий підхід полягає у тому, що дослідження процесу та конфігурування об'єднаною. Завдяки цьому можна подолати недоліки трьох попередніх підходів. Завдяки цьому відбувається кращий обмін між структурою моделей процесів та опціями конфігурацій.

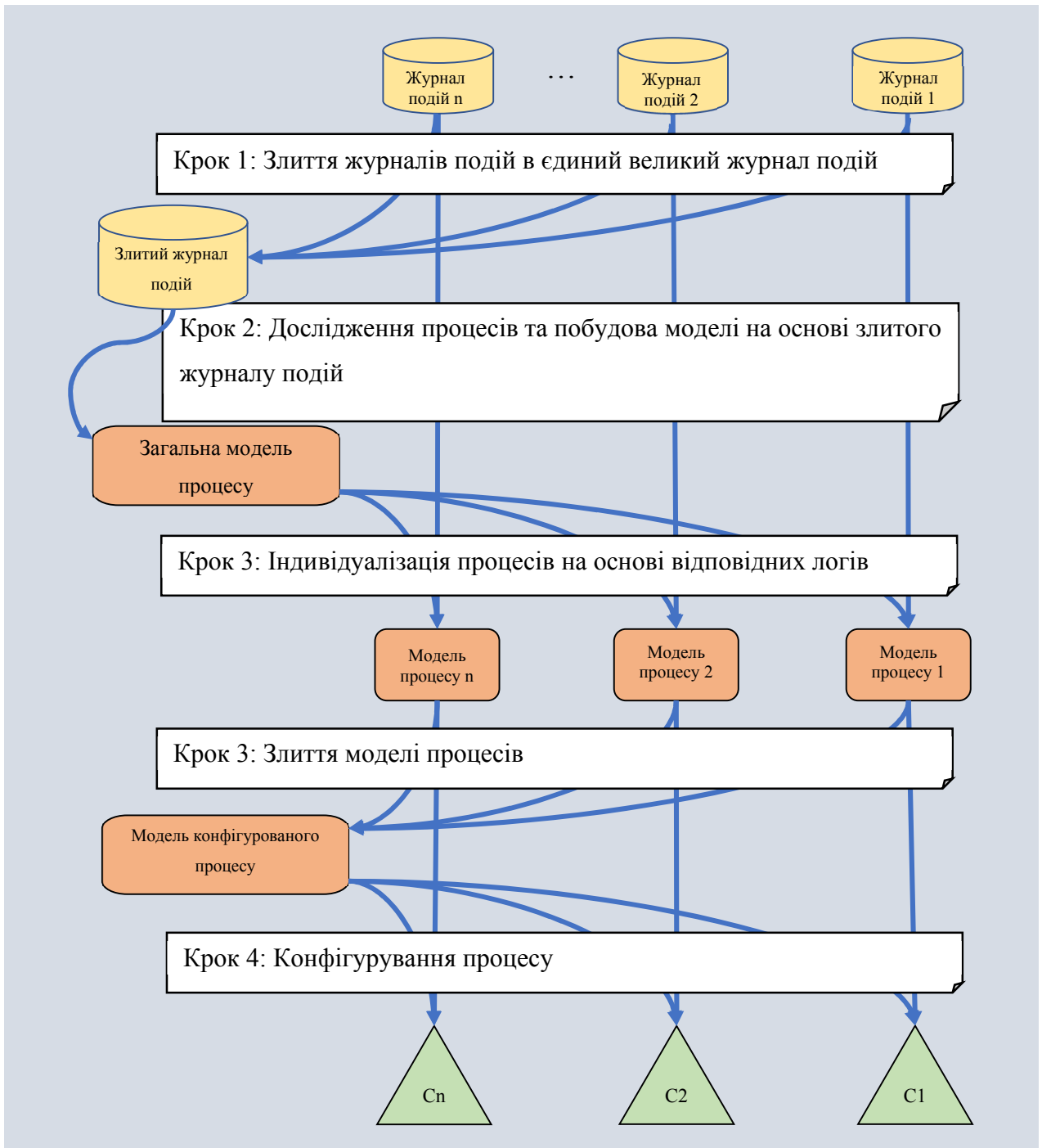


Рисунок 2.11 – Другий підхід. Зведення подібних моделей процесів

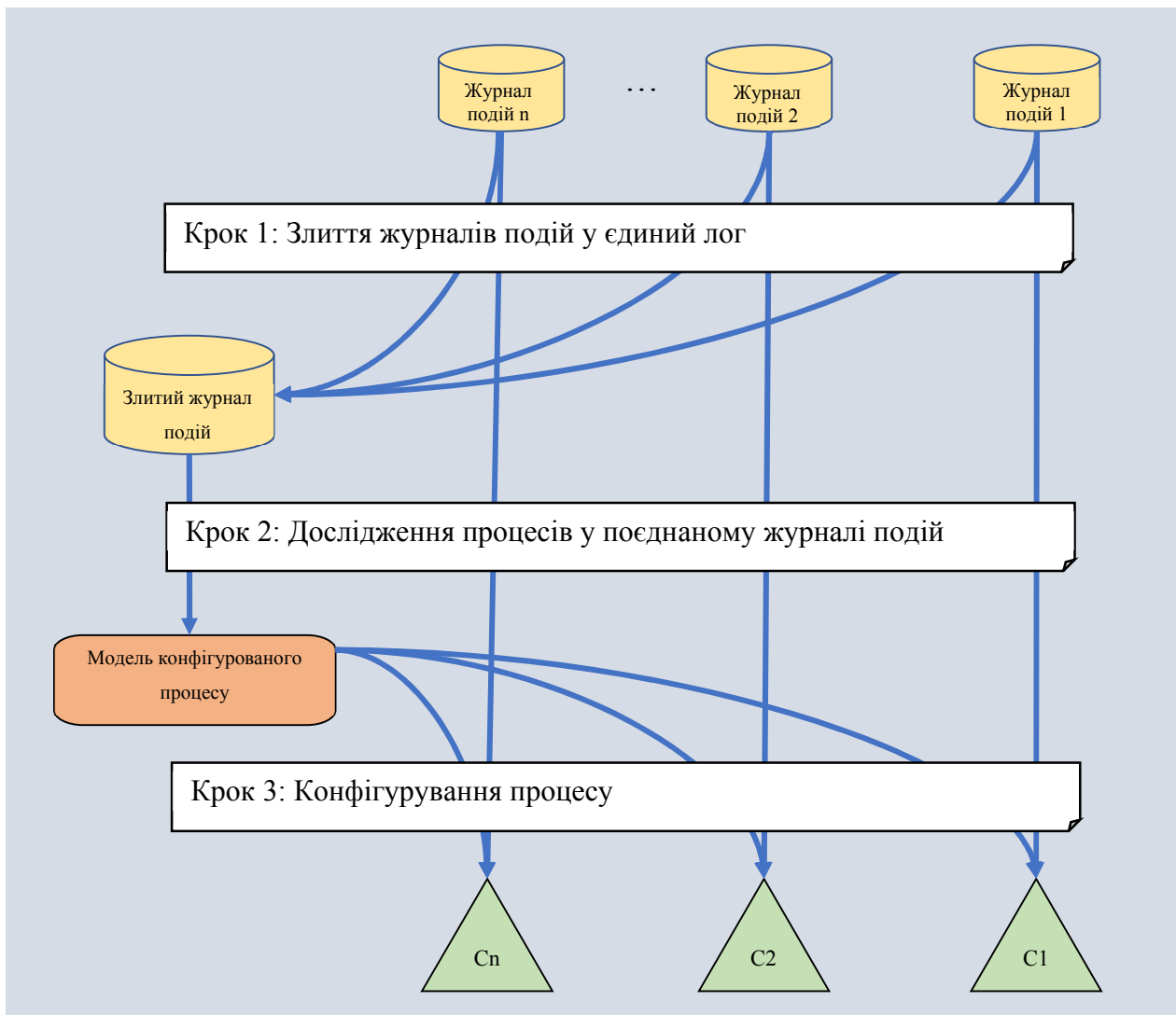


Рисунок 2.12 – Третій підхід. Спершу розроблено єдину модель процесу і після цього розроблено конфігуративну модель

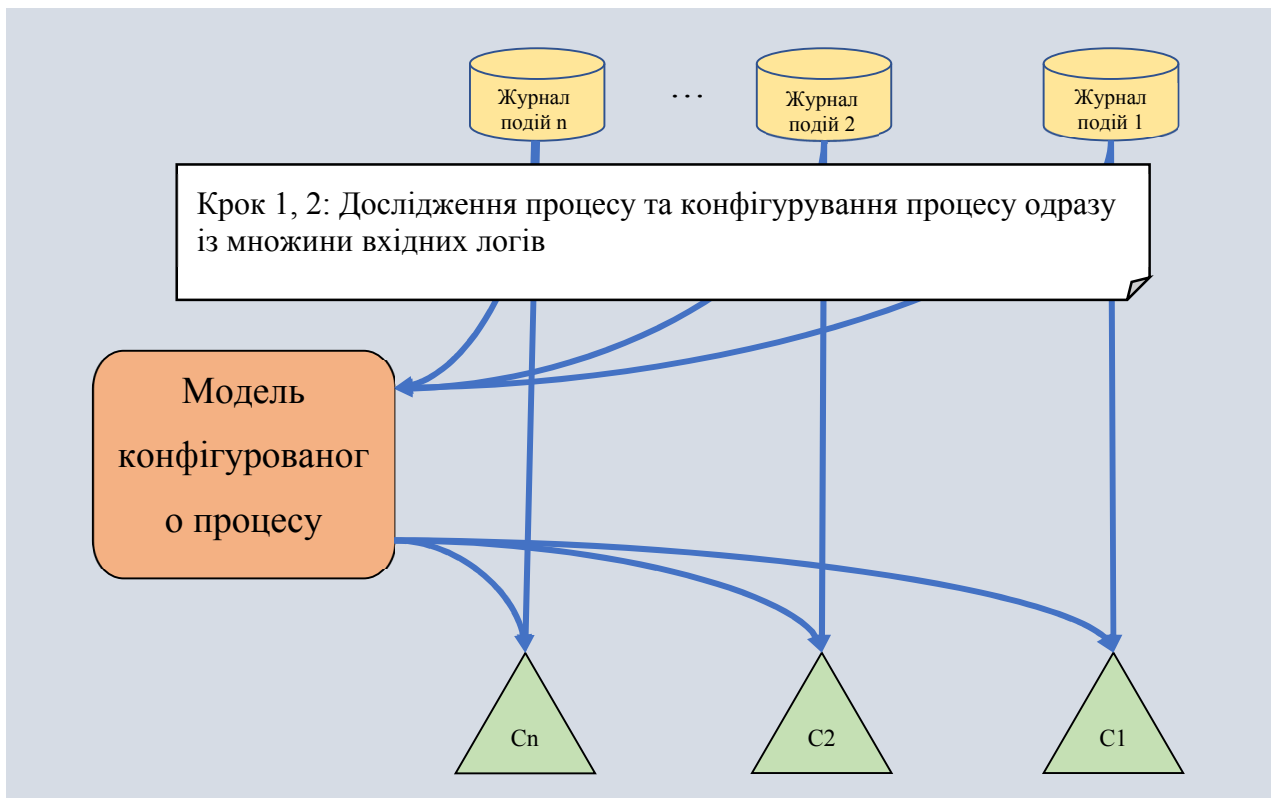


Рисунок 2.13 – Четвертий підхід. Дослідження моделі процесу та конфігурації одночасно.

ETM – це еволюційний фреймворк для дослідження даних, який використовує мутатори. Мутації кожного покоління виконуються мутаторами у моделі процесів. У кожному поколінні є краща (згідно з обраними характеристиками якості) модель. У вищепроведених підходах Бьойса [12] ETM розширено мутатором, який змінює опції конфігурацій. Також у цьому методі представлена якість конфігураційної перспективи. Завдяки цьому ETM може автоматично обирати між якістю моделі процесів та якістю конфігураційної перспективи.

Також цей метод пропонує інтерактивний метод дослідження. Він допомагає кінцевому користувачу у створенні фінальної моделі конфігураційних процесів. Спочатку добуваються фрагменти конфігураційного процесу із добірки журналів подій. Для того, щоб полегшити вибір конфігурації, із журналів подій

досліджуються керівні фрагменти. На кожному кроці конфігурування ці установи динамічно змінюються та доповнюються.

Серед указаних вище підходів перший та другий підходи мають складності зі зведенням моделей процесів згідно з їх поведінкою. Індивідуальні моделі, які мають бути поєднані, можуть бути зовсім різними, що робить складним їх поєднання належним чином. Третій та четвертий підходи дають змогу більш детально виразити поведінку у моделі конфігурованих процесів. Четвертий підхід, серед усіх інших має найбільшу пластичність.

2.1.1 Вирівнювання розмітки

Вирівнювання дозволяє відобразити спостережувану поведінку (наведену у журналі подій) відносно модельованої поведінки (дерево процесів). Також відображення (mapping) використовується для знаходження точок на дереві процесі, що відповідає конфігураціям у журналі подій. Усе починається із руху виключно по логу, такий рух відповідає спостережуваній поведінці, але така поведінка ще не визначена на дереві процесів. Так як ми тільки вводимо обмеження для дерева процесів та не додаємо в нього поведінкові ознаки, такий рух не може бути використаний у дослідженні опцій конфігурації. Далі кожен рух зв'язується з відповідним вузлом дерева та усіма батьківськими вузлами. Однак ми залишаємо сегрегацію (розділення) трас у відношенні до вузлів.

2.1.2 Приховання та блокування

Після розмітки вирівнювання метод починає пошук найсильніших конфігуративних опцій: приховати (hide) та заблокувати (block). Головна ідея

полягає у тому, що якщо у журналі подій частина дерева процесів виконується нечасто, цю частину треба приховати або заблокувати. Опція приховання вказує на те, що вузол можна пропустити, у той час якщо вузол заблокований, то через цей вузол немає можливості пройти та пропустити його. Якщо нащадок вузла заблокований, єдиний нащадок \wedge , \rightarrow , \leftarrow - вузлів блокований, або вихідний нащадок нащадку \cup - вузла заблокованими, сам вузол теж є заблокованим.

Для знаходження цих опцій використовується підхід «зверху вниз», якій починається з кореня дерева процесів. Для кожного вузла у дереві процесів ми порівнюємо частини трас до вузла згідно із журналом подій. Якщо частка цієї траси у порівнянні з журналом подій нижче заданого значення, виконується приховання або блокування. Коли вузол має одну з вказаних вище опцій, ми не відвідуємо нащадків цього вузла та одразу переходимо до наступного вузла, який не є частиною під-дерева прихованого або блокованого вузла. За замовчуванням припускається, що вузол має бути блокований. Задля уникнення поширення блокування на батьківські вузли їх треба приховати. Приховання потрібно для вузлів типу \wedge , \rightarrow , \leftarrow , або вихідний нащадок нащадку \cup - вузла а також якщо нащадки батьків вже заблоковані. Також треба приховувати кореневий вузол.

Нехай L є повним журналом подій та $L_n \subseteq L$ є колекцією трас, які відносяться до вузла n дерева процесів. Нехай t_{hb} є заданим пороговим значенням, яке має значення $0 \leq t_{hb} \leq 1$. Нехай $p(n)$ зображує батька вузла n або невизначено, якщо n є корневим вузлом. Нехай $c(n)$ відображає нащадка вузла n . Нехай $t(n)$ відображає тип вузла n . Ми повинні приховати або заблокувати тільки якщо

$$\frac{|L_n|}{|L|} < 1 - t_{hb} \quad (2.1)$$

Ми вибираємо відповідний варіант пониження опцій наступним чином:

- Якщо $p(n)$ невизначено, приховати
- Якщо $t(p(n)) \in \{\wedge, \rightarrow, \leftarrow\}$, приховати

- Якщо $m(p(n)) = \cup$ та n є дією або вихідним нащадком $p(n)$, приховати
- Нехай $c_{other} = \{c_i \in c(p(n)): c_i \neq n\}$, якщо $\forall c_i = c_{other}: c_i$ заблоковано, приховати
- Інакше, заблокувати

Згідно із визначенням граничним значенням $t_{hb} = 0,95$. Це показує, що у менше 5% трас виконується на вузлі, а тому повинні бути приховані або заблоковані. Після проходження всього дерева отримано початкову конфігурацію із опціями «приховати» та «блокувати». Наступним кроком буде визначаються можливі пониження опцій операцій.

2.1.3 Пониження операторів

Після визначення прихованих та заблокованих опцій потрібно знайти понижувані опції для вузлів – операторів (вузлів, які не є листами). Оператори можуть стати тільки більш обмежувачими. На Рисунку 2.3 зображена ієрархія пониження. У наведеній ієрархії вищі оператори є менш обмежувачими, ніж ті, що знаходяться нижче. Знову використовуємо підхід «зверху вниз» та відвідуємо усі вузли \vee, \wedge або \cup , які не є конфігурованими. Для пониження цих операторів використовуються наступні правила.

Паралельне виконання (\wedge) може бути понижене до операторів \rightarrow або \leftarrow . Кожна траса, яка відноситься до вузла \wedge , оцінюється по тому, як він виконується: зліва направо, з права на ліво або в іншому порядку. Якщо частина трас, яка виконується зліва направо по відношенню до загальної кількості трас, які відносяться до вузла \wedge , є меншою за вказане порогове значення, оператор понижується до оператора \rightarrow . Аналогічно відбувається пониження до оператора \leftarrow . Пониження оператора визначається наступним чином.

Нехай n є неконфігурованим вузлом \wedge , або вузлом \vee , який конфігурований як оператор \wedge дерева процесів. Нехай L є повним журналом подій та $L_n \subseteq L$ є колекція трас, які відносяться до вузла n . Нехай $L_{ltr} \subseteq L_n$ є колекцією трас які виконуються нащадками вузла n зліва направо. Нехай $L_{rtl} \subseteq L_n$ є колекцією трас які виконуються нащадками вузла n з права наліво. Не існує жодної траси, яка виконується в обох напрямках, тобто $L_{ltr} \cap L_{rtl} = \emptyset$. Нехай t_\wedge є заданим пороговим значенням $0,5 \leq t_\wedge \leq 1$.

Понижувати оператор \rightarrow можна тільки в тому випадку, якщо

$$\frac{|L_{ltr}|}{|L_n|} \geq t_\wedge \quad (2.2)$$

Понижувати оператор \leftarrow можна тільки в тому випадку, якщо

$$\frac{|L_{rtl}|}{|L_n|} \geq t_\wedge \quad (2.3)$$

Згідно із визначенням граничним значенням $t_\wedge = 0,95$. Це показує, що у менше 95% трас, які відносяться до вузла, виконується зліва направо, а тому повинні понизити оператор до \rightarrow , якщо 95% трас виконуються з права наліво, ми повинні понизити оператор до \leftarrow .

Оператор вибору (\vee) може бути понижений до оператора виключаючого вибору (\times) або паралельного виконання (\wedge). Спочатку припускається що ми можемо понизити до обох цих операторів. Тепер потрібно порівняти усіх нащадків оператора вибору до інших нащадків даного вузла. Якщо частина трас, які виконуються обома нащадків по відношенню до трас, які виконуються хоча би одним з двох нащадків, більша за вказане порогове значення, оператор вибору (\vee) понижується до оператора \wedge . Якщо це значення є меншим за порогове, оператор понижується до оператора \times . Якщо оператор понижується до \wedge , то проводиться подальше дослідження на пониження до \rightarrow або \leftarrow , як описано вище. Якщо оператор \vee понижений до \wedge , \rightarrow або \leftarrow , усі їх нащадки замість опції

блокування отримують опцію приховання. Пониження оператора визначається наступним чином.

Нехай n є неконфігурованим вузлом \vee . Нехай $c(n)$ відображає нащадка вузла n та визначає пари різних нащадків як $cp(n) = \{(c_1, c_2) \in c(n) \times c(n): c_1 \neq c_2\}$. Нехай L є повним журналом подій та $L_n \subseteq L$ є колекцією трас, які відносяться до вузла n . Нехай t_v є заданим пороговим значенням $0 \leq t_v \leq 1$.

Понижувати оператор \wedge можна тільки в тому випадку, якщо

$$\forall (c_1, c_2) \in cp(n): \frac{|L_{c_1} \cap L_{c_2}|}{|L_{c_1} \cup L_{c_2}|} \geq t_v \quad (2.4)$$

Понижувати оператор \leftarrow можна тільки в тому випадку, якщо

$$\forall (c_1, c_2) \in cp(n): \frac{|L_{c_1} \cap L_{c_2}|}{|L_{c_1} \cup L_{c_2}|} < 1 - t_v \quad (2.5)$$

Згідно із визначенням граничним значенням $t_v = 0,95$. Це показує, що у менше 95% трас, які відносяться до вузла, поинжується до оператора \wedge , та якщо менше ніж 5%, для виконання траси більш ніж одного нащадка оператор понижується до \times .

Оператор ітеративного виконання (\cup) понижується до оператора послідовного виконання (\rightarrow), якщо частина трас, які виконують дію циклу «зробити» або «повторити» у порівнянні з трасами, які мають тільки дію «зробити» нижче заданого порогового значення. Пониження оператора визначається наступним чином.

Нехай n є неконфігурованим вузлом \cup . Нехай дія – нащадок «зробити» вузла n позначається як c_{do} а нащадок «повторити» позначається я c_{redo} . Нехай L є повним журналом подій та $L_n \subseteq L$ є колекцією трас, які відносяться до вузла n . Нехай t_\cup є заданим пороговим значенням $0 \leq t_\cup \leq 1$.

Понижувати оператор \cup можна тільки в тому випадку, якщо

$$\frac{|L_{cdo} \cap L_{credo}|}{L_n} < 1 - t_{\sigma} \quad (2.6)$$

Згідно із визначенням граничним значенням $t_v = 0,95$. Це показує, що якщо у менше ніж 5% трас, які відносяться до вузла, виконуються нащадки «зробити» та «повторити», його треба понизити до оператора \rightarrow .

2.1.4 Оцінка дерев багатоваріантних процесів

Аналіз якості дерева конфігуративних процесів робиться із використанням вимірів якості. Ці виміри якості визначаються як:

- Сумісність із журналом подій (replay fitness): Описує, наскільки близька поведінка моделі процесів до спостерігаємої у журналі подій. Моделі з ідеальною сумісністю повинна мати абсолютно ідентичну поведінку порівняно з журналом подій. Квіткова модель (a flower - model) – це модель, яка дозволяє виконання процесу у будь якій час. Приклад квіткової моделі з шістьма діями приведений на Рисунку 2.11. Квіткова модель містить у собі усі дії (activities), може повторити усі траси журналу подій, що робить її моделлю з ідеальною сумісністю.

- Точність: Ідеальна сумісність може бути досягнена завдяки створенню квіткової моделі, яка містить у собі усі дії. Однак ця модель underfit журнал подій так як містить у собі дуже багато поведінкової інформації. Точність описує наскільки модель процесів подібна до журналу поведінки. Для квіткової моделі (Рисунок 2.14) точність буде дуже низькою. Ідеальної сумісності та точності одночасно можна досягти за допомогою створення трасової моделі. Трасова модель починається з вибору між послідовностями дій, кожна з яких описує трасу журналу подій. Приклад трасової моделі для трьох трас представлений на Рисунку 2.15. Для рецензування даної моделі нам потрібні поняття generalization та простоти (simplicity).

– Узагальнення: Не вся поведінка може бути охоплена і занесена у журнал подій. Поняття узагальнення описує переповненням моделі моделі по відношенню до журналу подій. Модель, яка не є узагальненою, вважається переповненою. Трасова модель точно описує журнал подій, що робить її (модель) переповненою

– Простота: Охоплює принцип Бритви Оккама. Розмір моделі є основним індикатором її простоти, це означає, що найменша модель процесів є кращою моделлю у цьому вимірі якості. Трасова модель (Рисунок 2.15) містить у собі послідовності для усіх трас, через це модель стає об'ємною, що призводить до зниження простоти.

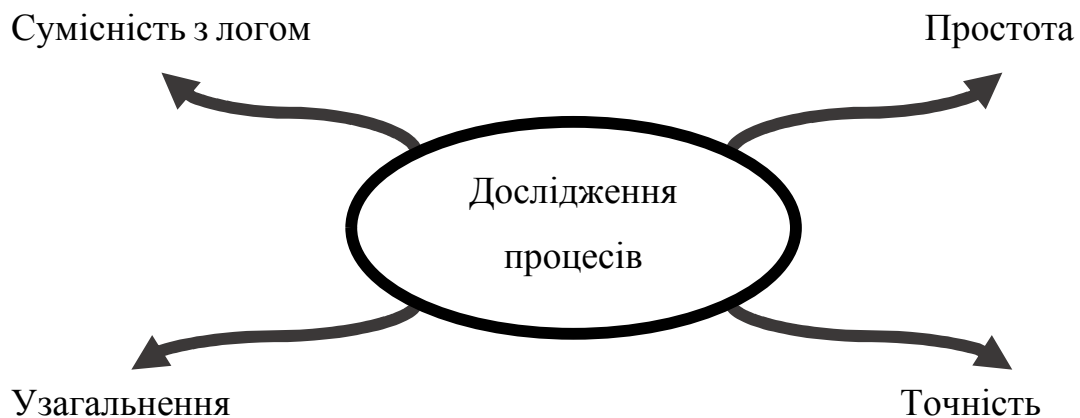


Рисунок 2.14 – Чотири виміри конкуруючих якостей.

Для визначення оцінки якісного виміру для дерева конфігуративних процесів як середнього зваженого оцінок варіантів конфігурації для дерева конфігуративних процесів. Цей усереднений показник якості визначається наступним чином.

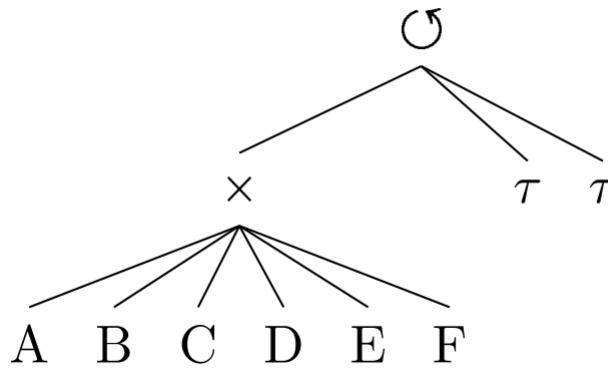


Рисунок 2.15 – Квіткова модель для дій A, B, C, D, E та F. Ця модель має ідеальну сумісність повтору для журналу подій, який містить тільки ці дії.

Нехай LC позначає колекцію вхідних журналів подій, та PT^c позначає дерево конфігурованих процесів. Нехай варіанти конфігурацій PT^c для вхідного журналу подій $L \in LC$ позначається як PT_L^c .

Нове визначення оцінки якісного виміру PT^c виглядає наступним чином

$$Q(PT^c, LC) = \frac{\sum_{L \in LC} |L| \times Q(PT_L^c, LC)}{\sum_{L \in LC} |L|} \quad (2.7)$$

Використовуючи такий опис стає можливою оцінка дерева конфігуративних процесів на чотирьох вимірах якості. Однак таке визначення бере до уваги лише розмір дерева процесів (простота), як дерево відображає спостережувану поведінку (Сумісність з логом) та до якої міри його можна розшири для неспостережуваної поведінки (Узагальнення). Якість опцій конфігурації не враховується. Ідея полягає у тому, що модель конфігуративних процесів із найменшою кількістю опцій конфігурації є найкращою. Задля цього запропоновано введення п'ятого виміру якості – доречність конфігурації, яка визначається наступним чином.

Нехай PT^c позначає дерево конфігурованих процесів. Доречність конфігурації визначається як відношення кількості вузлів конфігурації у PT^c до кількості усіх вузлів у PT^c .

Використовуючи вказані вище визначення можна отримати загальну оцінку якості для дерева конфігуративних процесів.

Нехай $Q_{rf}(PT^c, LC)$, $Q_s(PT^c, LC)$, $Q_p(PT^c, LC)$ та $Q_g(PT^c, LC)$ позначають моменти $Q(PT^c, LC)$ та відповідають вимірам якості. Нехай w_{rf} , w_s , w_p та w_g відповідно відображають вагу цих вимірів якості.

Дві опції для визначення загальної якості дерева конфігуративних процесів є наступними:

Сумма ваги (Загальна), у якій w_c позначає вагу виміру доречності конфігурації. Тоді сума ваг п'яти вимірів якості визначається як

$$Q_{wt}(PT^c, LC) = w_{rf}Q_{rf}(PT^c, LC) + w_sQ_s(PT^c, LC) + w_pQ_p(PT^c, LC) + w_gQ_g(PT^c, LC) + w_cQ_c(PT^c) \quad (2.8)$$

Сумма ваги (Роздільна) згідно із цим α , яке приймає значення $0 \leq \alpha \leq 1$ та відображає важливість виміру доречності конфігурації. Тоді сума ваг п'яти вимірів якості визначається як

$$Q_{ws}(PT^c, LC) = (1 - \alpha)(w_{rf}Q_{rf}(PT^c, LC) + w_sQ_s(PT^c, LC) + w_pQ_p(PT^c, LC) + w_gQ_g(PT^c, LC)) + \alpha Q_c(PT^c) \quad (2.9)$$

Для обох цих опцій чим більшою є оцінка, тим кращим є дерево процесів. Виходячи з цього, дерево конфігурованих процесів з найвищою оцінкою є найкращім деревом конфігурованих процесів. Перша опція надає повністю інтегроване вимірювання моделі та вимірів якості. Друга опція відмежує ці два аспекта. Найчастіше першим бажано розглядати якісні виміри моделі і лише у випадку невизначеності надавати перевагу моделі із кращою доречністю конфігурації.

2.2 Удосконалення методу побудови моделі багатоваріантного процесу

Запропоновано удосконалений метод кластеризації трас, який полягає у побудові ієрархії пар подій за частотою їх використання в трасах логу.

Запропонований метод досліджує сусідні пари подій у трасах. Якщо пара сусідніх подій (а, б) зустрічається у 100% досліджуваних трас, такій парі присвоюється значення ваги 1. Значення ваги відповідає процентному значенню появи пари у трасах. Нехай значення ваги позначається як w_t . Якщо T позначає кількість трас, а L позначає кількість появ пари подій у трасі, то значення ваги траси має значення:

$$w_t = \frac{L}{T} \quad (2.10)$$

Ті пари подій, які мають однакове значення ваг, відносяться до одного кластера. Далі ми отримуємо значення ваг усіх пар для подальшого кластерування.

Розглянемо наступний приклад. Нехай нам дано траси, які наведено в Таблиці 2.1. Кожна траса може містити у собі дії, які для прикладу позначені числами від 1 до 10. За допомогою алгоритму ієрархічної кластеризації ми утворюємо групи пар подій, які мають однакову вагу. У даному прикладі ми маємо три індивідуальні траси. Пари подій (1, 2), (2, 3) та (6, 7) містяться у кожній з трьох наведених трас, що наведено на Рисунку 2.1.

Таблиця 2.2 – Добірка трас робочого прикладу.

t_1	<1, 2, 3, 4, 6, 7, 8>
t_2	<1, 2, 3, 5, 6, 7, 9>
t_3	<1, 2, 3, 4, 5, 6, 7, 8, 10>

1	2	3	4	-	6	7	8	-	-
1	2	3	-	5	6	7	-	9	-
1	2	3	4	5	6	7	8	-	10

Рисунок 2.1 – Схематичне зображення трас та обраних пар подій.

Після обчислень ваги кожної з пар згідно з трасами, ми отримуємо три виражені кластери із парами подій, відокремленими у різні кластери за значенням ваги. Для роботи методу використовується агломеративний алгоритм алгоритм ієрархічної кластеризації. Спочатку кожна пара утворює окремий кластер.

Метод містить у собі наступні етапи:

1. Розрахунок кількості повторів пар сусідніх подій на одній трасі та по трасам;

2. Відбір пар подій таких, що задовільняють умові $k_{ei} \geq k_t$, де k_{ei} – кількість пар (e_i, e_{i+1}) , k_t – кількість трас. Отримана множина подій складає базову конфігурації. Бізнес процесу;

3. Розбиття множини подій, що залишилися на задану кількість кластерів. Результатом буде множина додаткових конфігурацій багатоваріантного БП.

Таким чином завдяки ієрархічній кластеризації ми отримуємо так званий базис моделі, спрощену модель процесу який складається з пар трас журналу подій, які зустрічаються у кожній трасі хоча б один раз. Завдяки цьому ми можемо детальніше визначити основні процеси та знаходити конфігурації у кластерах трас із меншою вагою завдяки, наприклад методу Ліктя.

3. ПОБУДОВА ПЛАНУ РОЗРОБКИ ПРОЕКТУ

Проект – це процес з чітко визначеними часовими рамками, мета якого – створити унікальний продукт або отримати певні інноваційні результати. Відповідно, управління проектами – це конкретна діяльність, мета якої – реалізувати всі поставлені проектом завдання. Для цього прикладається максимальна кількість зусиль, знань, досвіду, методик та інструментарію.

Управління проектами допомагає як найефективніше та за найбільш короткий час досягти виконання поставлених задач. Окрім цього, під час проведення проектного менеджменту формується ціла система комплексів та методів, які можуть бути використані для найбільш ефективного у плані витрат коштів та ресурсів рішення для досягнення поставлених цілей

Також управління проектами можна визначити як процес управління командою, ресурсами проекту за допомогою спеціальних методів та прийомів з метою успішного здійснення поставленої цілі.

Проектний менеджмент складається із ряду послідовних дій, а саме:

- Визначення та формування вимог проекту;
- Формування максимально чітких та зрозумілих цілей;
- Встановлення та реалізація комунікації між задіяними у проекті сторонами;
- Спілкування з командою, врахування її потреб/побажань/очікувань і корекція існуючих планів відповідно до отриманих матеріалів.

Всі ці дії сегментуються на окремі етапи:

- Ініціація (тобто старт проекту) є певним знайомством з проектом. Визначається його суть і цілі, формується відповідна до цього команда.
- Планування – найважливіша частина в управлінні проектом. Як стверджується в класичній методології по РМВОК, це повинно зайняти приблизно 50% всього часу в процесі реалізації проекту. Складність в тому, що

під час цього етапу ретельно прописуються всі дії, які повинна здійснити команда для досягнення заданої мети. Для цього проект спочатку розділяється на частини і набір дрібних завдань. Створюється певний «графік робіт», в якому прописуються дедлайни для кожної з задач. Також опрацьовується список необхідних ресурсів. При цьому планування включає в себе періодичне коригування, адже в процесі роботи постійно з'являються нові нюанси і підзадачі, стають явними певні «підводні камені» проекту;

– Виконання і контроль. Цей етап слід чергувати з попереднім. В ідеальній системі управління проектом все виглядає так: поставили завдання, зробили його, проконтролювали, внесли в план необхідні корективи, поставили нову задачу і так далі. На етапі виконання зазвичай в хід йдуть певні інструменти для полегшення перебігу процесів: делегування, тайм-менеджмент, матриця Ейзенхауера і викреслювання справ. Що стосується кожного з методів, то є маса навчальних посібників, а про тайм-менеджменті ми писали раніше в нашому блозі.

– Завершення проекту. На цьому етапі робиться контрольна перевірка виконаної роботи і обов'язково зберігаються вихідні дані, задіяні інструкції і регламенти. Це потрібно для того, щоб навіть нова людина в команді змогла розібратися, що і як робили до неї.

Саме ретельне планування, організація завдань і проектних складових, забезпечення необхідними ресурсами і контроль дієвості обраної стратегії – це і є те головне, що входить в управління проектами з перспективою досягнення поставленої мети.

Існує досить велика кількість методів, як здійснюється управління проектами. Крім класичного, який ми якраз і описали вище, використовується Agile – коли один великий проект ділиться на масу міні-проектів з поетапною реалізацією; Scrum – поділ проекту на складові частини; Lean – розподіл проекту на дрібні пакети робіт; Kanban – варіант для проектів, не обмежених по дедлайнах, адже їх можна ставити на паузу. У кожної з методик є багато власних

нюансів, переваг і недоліків. Вибір підходящої системи залежить від специфіки організації і команди, яка буде працювати над конкретним проектом.

Ціль проекту – це бажаний результат діяльності, який намагаються досягти за певний проміжок часу при заданих умовах реалізації проекту. Ціллю даної роботи є удосконалення існуючого методу кластеризації трас журналів подій для більш точно побудови моделей процесів.

Досягнення цілей проекту можливе при застосуванні наступних загальновідомих підходів до управління:

- Класичний підхід, який об'єднує такі основні функції управління, як планування, організацію, впровадження, контроль та керівництво. Цей підхід був запропонований Генрі Файолем у 1949 році;

- Управління проектом як циклом розв'язання проблеми. Для вирішення проблеми, що виникла, потрібно здійснити відповідні кроки реалізації проекту. На першому етапі необхідно провести аналіз проблеми, зібрати відповідні дані, розробити можливі пропозиції вирішення проблеми, провести їх оцінку. На наступному етапі важливо здійснити альтернативний вибір (прийняти найкраще рішення). Наступним кроком є організація команди виконавців, розробка плану, його впровадження, та проведення моніторингу.

- За життєвим циклом проекту.

При постановці цілей проекту потрібно знайти відповіді на наступні питання:

- Як конкретно повинен виглядати результат проекту (характеристика результатів проекту)?

- Які умови повинні враховуватись в процесі реалізації проекту (вимоги та обмеження)?

- Хто це буде робити? Коли це буде зроблено?

- Скільки це буде коштувати?

У даному випадку результатом проекту буде удосконалений метод проекту представлений у вигляді допрацьованого модулю для фремворку роботи з алгоритмами дослідження процесів ProM. Обов'язковими умовами є вимоги до

написання атестаційної роботи. Виконувати це повинен студент у відведений для виконання цього проекту період навчального року. Проведення коштовної оцінки не є можливим, так як даний проект не є комерційним.

Основними принципами управління проектами є:

- Цілеспрямованість. Це поняття виражається в цільовій орієнтації проекту на забезпечення кінцевих цілей діяльності підприємства;

- Системність. Дане поняття передбачає розгляд проекту нововведень із системних позицій. Це означає, з однієї сторони, те, що процес управління проектами є одним цілим із своїми закономірностями формування й розвитку, а, з іншої, можливість розділення проекту на підсистеми і дослідження їх взаємозв'язку, оскільки кожна з них впливає як на всі інші підсистеми, так і на весь проект в цілому. Таким чином виникає можливість відкрити і спроектувати раціональний зв'язок підсистем, їх співвідношення і субординацію, дати кількісні й якісні оцінки ходу реалізації проекту та його окремих частин. На практиці потрібна чітка структуризація проекту й розробка комплексу взаємозв'язаних організаційно-економічних, законодавчих, політичних, техніко-технологічних та інших заходів, що забезпечують його реалізацію;

- Комплексність передбачає розгляд явищ в їх зв'язку і залежності. Комплексний підхід в проект-менеджменті передбачає: спільне використання різних форм та методів управління при розробці і реалізації нововведень; розгляд всіх спільних цілей управління по рівнях і ланках організаційної й виробничої структури підприємства; зв'язок окремих елементів проекту між собою і з головною (кінцевою) ціллю проекту; розгляд окремих проблем проекту з точки зору часових інтервалів;

- Забезпеченість, яка полягає в тому, що всі заходи, що передбачені проектом, повинні бути укомплектовані різними видами ресурсів, що необхідні для його реалізації;

- Пріоритетність означає, що при розробці і реалізації проекту перевага надається першочерговим завданням, виходячи з загальної концепції стратегічного розвитку;

– Економічна безпека заходів, що плануються. Вона повинна розраховуватися на основі оцінки ймовірності виникнення збитків або будь-яких втрат в результаті нездійснених запланованих проектом подій.

Життєвий цикл проекту – це час від моменту його задуму до моменту ліквідації. Слід позначити, що для різних учасників проекту події “початок проекту - кінець проекту” будуть різними. Наприклад, для інвесторів початок проекту пов’язаний з вкладенням коштів у підприємство. Закінченням проекту буде вилучення об’єкту з експлуатації та припинення отримання прибутку від вкладених коштів. Для інших учасників та виконавців його закінченням буде припинення виконання окремих етапів робіт.

Життєвий цикл проекту розбивають на фази та стадії. Виділяють такі фази життєвого циклу проекту:

- Зародження;
- Зростання;
- Зрілості;
- Завершення.

Стадія проекту одна з послідовно виконуваних частин створення проекту, встановлена нормативними документами, що закінчується заданим результатом.

Фаза зародження включає такі стадії, як розробка концепції, яка характеризується появою загальної ідеї; стадія аналізу та вивчення можливостей, що визначає приблизні витрати, обсяг робіт, терміни виконання, визначається реальність даного проекту.

Фаза зростання включає стадії планування та конструкторської розробки. На даному етапі розробляється план виконання, готуються необхідні документи, тобто загальний бюджет проекту, план ресурсного забезпечення та календарний план. Крім того, розробляються та погоджуються конструкторські розробки (загальна схема, креслення кожного компоненту).

Фаза зрілості включає стадію забезпечення необхідними матеріалами та обладнанням та стадію виробництва. Проводиться контроль обсягів, витрат, якості та своєчасності виконання робіт.

Фаза завершення характеризується завершенням робіт, проводиться оцінка отриманих результатів, аудит, порівняння з наміченими цілями, підсумкова звітність, нагороджується та розпускається команда. Зрозуміло, що наприкінці проекту робляться відповідні висновки, узагальнюються позитивні та негативні результати, їх причини з метою прийняття відповідних рішень та накопичення досвіду.

План проекту являє собою перелік робіт із зазначенням термінів, виконавців, результатів, які ведуть до отримання комплексу показників, що намічені концепцією проекту. Він є основою для проведення тендерів, укладання контрактів із розробниками й виконавцями всіх передбачених робіт, складання детального сіткового графіка роботи та оцінки робіт, що заплановані.

План складається за участю всіх зацікавлених осіб, що сприяють його реалізації. Частіше всього передінвестиційна стадія закінчується розробкою бізнес-плану, який являє собою конкретний план дій підприємця (суб'єкта господарювання) по реалізації своєї ідеї. Бізнес-план являє собою складну систему техніко-економічних розрахунків, оцінок, обґрунтувань, що на даній стадії детально не завжди вдається зробити. Але всі розділи бізнес-плану повинні бути продумані, а потім уточнені на інвестиційній стадії.

Сітьовий графік – це інформаційно-динамічна модель, яка відображає всі логічні взаємозв'язки та результати робіт, основний документ системи сітьового планування, а саме сітьове планування – одна з форм графічного відображення змісту робіт і тривалості виконання планів, яка забезпечує наступну оптимізацію розробленого графіка на основі економіко-математичних методів та комп'ютерної техніки. Сітьові графіки визначенням логічних зв'язків між ними. Залежно від способу зображення розрізняють стрільчаті графіки і графіки передування. Від графіків передування простіше перейти до діаграм Ганта, які є формою календарного планування. Діаграма Ганта дає можливість наочно визначити, які роботи є критичними, а які – не критичними, який резерв часу мають не критичні роботи, логічний зв'язок між роботами. Діаграма Ганта є прекрасним засобом планування й контролю, дає зрозуміти ідею резерву часу і

його використання. Її достоїнствами є легкість побудови та читання, наочність представлення перебігу виконання робіт за проектом. Основна частина. Перед тим, як розмістити роботу на діаграмі, потрібно розглянути, чи існує логічний зв'язок між роботами, яка тривалість робіт залежно від забезпечення необхідними ресурсами, який розподіл ресурсів між роботами. Ідея графічного зображення взаємозв'язків між роботами не є новою. Новими є методи, інакше техніки оптимізації часових та вартісних параметрів і обробка інформації при використанні ЕОМ. Поєднання нових методів із старими привело до створення методу оцінки та перегляду планів PERT. Завдяки системі PERT менеджери швидко можуть визначити «вузькі місця» у виконанні графіків та розподілити належним чином ресурси з метою ліквідації відставань. Існує близько 100 різновидів методу PERT, але всі вони мають і загальні характеристики. PERT – це засіб аналізу задач, необхідних для виконання проекту, зокрема, аналізу часу, необхідного для виконання кожної окремої задачі, а також визначення мінімально необхідного часу для виконання всього проекту. PERT розроблено для складання масштабних і складних проектів без точного знання деталей з урахуванням невизначеності, оскільки передбачає задання для кожної роботи трьох варіантів її тривалості: оптимістичного, песимістичного і найбільш реального. Найбільш популярною частиною PERT є метод критичного шляху СРМ для розв'язання детермінованих задач. Критичний шлях – це послідовність робіт, що не мають резерву часу, і зміна їх тривалості призведе до зміни тривалості всього проекту, тому на них слід звертати максимум уваги. СРМ спирається на побудову сітьових графіків методом попередніх, стрілочних або умовних діаграм.

3.1 Розробка плану проекту

Для даної роботи був розроблений план проекту представлений у вигляді діаграми Ганта, як показано на Рисунку 3.1. В неї були занесені усі завдання, які потрібно було виконати для захисту атестаційної роботи. Кожна задача має часові рамки, у які повинна бути виконана. У протилежному випадку, якщо не витримати встановлений режим роботи над вказаними задачами, виникне запізнення, що призведе до скорочення відведеного часу роботи над проектом. Далі буде проведено експеримент, під час якого виконання першого етапу задачу було почато із затримкою у 170 днів, та які зміни треба внести у план проекту для компенсації цієї затримки.

На Рисунку 3.1 наведено робочий простір програми MS Project у режимі відображення «Діаграма Ганта». З лівої сторони робочої зони міститься таблиця, у якій наведено усі задачі, які потрібно виконати упродовж життєвого циклу даного проекту. Кожна задача представляє собою кортеж, який містить наступні атрибути:

- Назва задачі. Це є унікальним ідентифікатором задачі;
- Тривалість. Скільки часу потрібно на виконання конкретної задачі;
- Дата початку. Запланований час початку виконання задачі
- Дата кінця. Запланований час завершення роботи над завданням
- Попередня задача. Даний атрибут вказує на зв'язок поточної задачі попередніми. Наприклад, задача 23 «Підготовка презентації атестаційної роботи» почнеться тільки після завершення задачі 10 «Аналіз методів дослідження конфігурованих процесів» та одночасно із початком задачі 15 «Початок роботи над пояснювальною запискою».

З правої сторони робочої зони міститься саме Діаграма Ганта, яка являє собою відрізки (графічні плашки), розміщені на горизонтальній шкалі часу. Кожен відрізок відповідає окремому завданню або підзадачі. Завдання і підзадачі, складові плану, розміщуються по вертикалі. Початок, кінець і довжина відрізка на шкалі часу відповідають початку, кінцю і тривалості завдання. На деяких діаграмах Ганта також показується залежність між завданнями

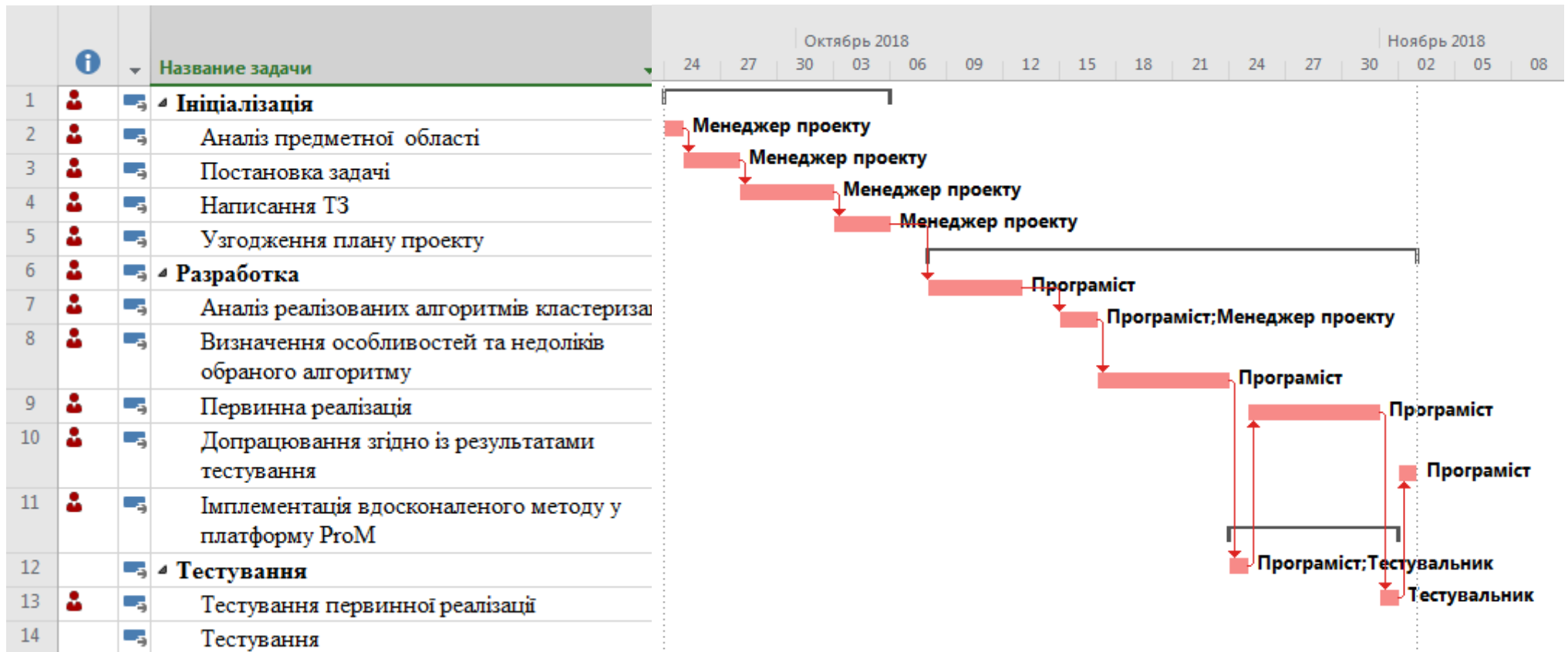


Рисунок 3.1 – План проекту, розроблений у програмі MS Project

Діаграма може використовуватися для представлення поточного стану виконання робіт: частина прямокутника, що відповідає завданню, заштриховується, відзначаючи відсоток виконання завдання; показується вертикальна лінія, що відповідає моменту «сьогодні».

На діаграмі, наведеній на Рисунку 3.1 зображені відрізки, кожен з яких відповідає конкретній унікальній дії. Кожен з цих відрізків має довжину, яка відповідає тривалості роботи над завданням, початок та кінець відрізків співпадає із часом початку і кінця виконання завдання. Кожен відрізок має зв'язок з іншим, що відображається стрілками. Відрізки також мають різний колір. Червоний колір відрізка означає, що задача лежить на критичному шляху.

Критичний шлях – це шлях моделі, тривалість якої дорівнює критичній. Критична тривалість – це мінімальна тривалість, протягом якої може бути виконаний весь комплекс робіт проекту. Тобто виходячи з цього, критичний шлях є послідовністю робіт з нульовим резервом часу. На Рисунку 3.1 він позначений червоним кольором.

Важливим моментом для використання метода критичного шляху (CPM) є побудова моделі проекту, що включає наступне:

- Список усіх дій, що потребуються для завершення проекту;
- Необхідна тривалість для кожної діяльності для завершення;
- Залежності між діями;
- Логічні фінальні точки, такі як віхи чи готові вироби.

Використовуючи ці значення, CPM вираховує найдовший шлях запланованих задач до логічних фінальних точок чи до кінця проекту, а також найбільш ранню та найпізнішу дату, коли кожна діяльність може бути почата та завершена без затримки проекту в цілому. Цей процес визначає, які діяльності є «критичними» (тобто на найдовшому шляху) та які «повністю гнучкі» (тобто можуть бути відкладеними без затримки проекту). У проектному менеджменті, критичний шлях є послідовністю дій мережі проекту, що додаються до найдовшої тривалості проекту, незалежно від того чи гнучка ця тривалість, чи ні. Це визначає найкоротший можливий час для завершення проекту. У рамках

критичного шляху можуть зустрічатися «повна гнучкість» (невикористаний час). Наприклад, якщо проект тестує сонячну панель і задача «В» потребує «сходу сонця», може бути заплановане обмеження для старту тестування діяльності допоки не настане час сходу сонця. Це може встановити мертвий час (повна гнучкість) у графік дій за даним шляхом згідно сходу сонця відповідно до потреби чекання настання цієї події. Цей шлях, зі отриманими обмеженнями повної гнучкості, стане довшим, роблячи повну гнучкість частиною найкоротшої можливої тривалості проекту в цілому. Іншими словами, індивідуальні задачі на критичному шляху згідно обмежень можуть бути затримані без подовження критичного шляху; це є «повною гнучкістю» даної задачі. Крім того, доданий час до проекту цими обмеженнями насправді є критичним шляхом опору, на величину якого тривалість проекту буде поведена згідно кожної діяльності критичного шляху і обмежень.

Зв'язки між задачами допомагають не тільки визначити зв'язок між задачами та їх послідовність, але й показати, що одна дію повинна бути строго виконана після завершення попередньої дії або починатися разом із зв'язаною дією. Види зв'язків є наступними:

- Закінчення – початок;
- Закінчення – закінчення;
- Початок – початок;
- Початок закінчення.

У Microsoft Project можна обрати тип зв'язку, як зображено на Рисунку 3.2

Кожне завдання потребує ресурсу для його виконання. Управління ресурсами - одна з головних підсистем управління проектом, яка включає процеси планування, закупівлі, постачання, розподілу, обліку і контролю ресурсів.

Поняття «ресурс» в методології управління проектами трактується як все, що має в своєму розпорядженні проект, у тому числі трудові, фінансові і матеріально-технічні ресурси, команда проекту, час (тривалість, терміни обмеження), інформація, знання і технології.

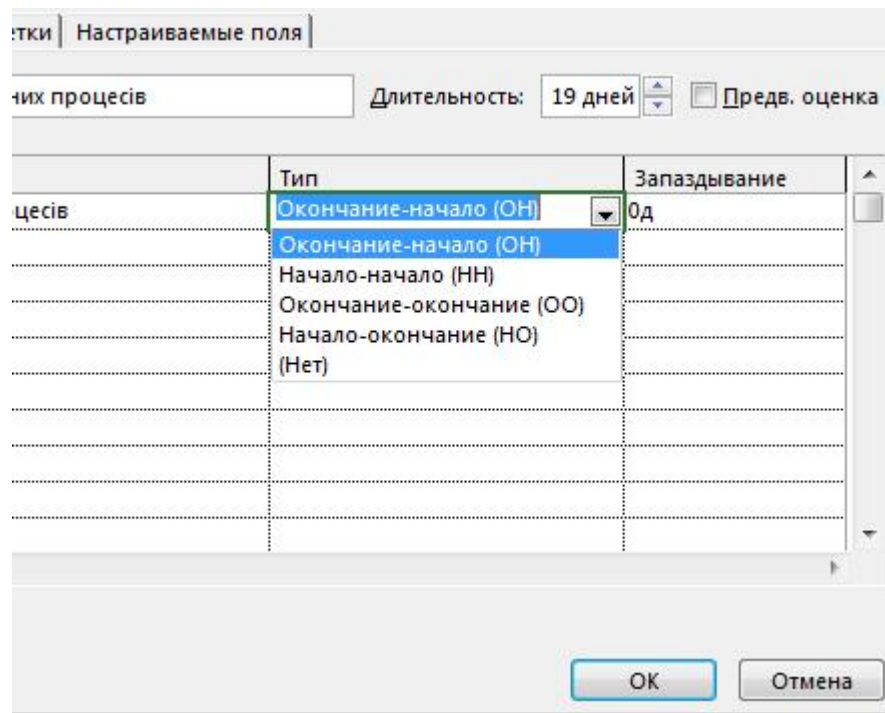


Рисунок 3.2 – Панель обрання типу зв'язку у вікні «Відомості про задачу»

Основна мета управління ресурсами - забезпечити їх оптимальне використання для досягнення кінцевої мети управління проектом. На Рисунку 3.3 зображено процес призначення ресурсу для конкретної задачі. Усі ресурси проекту, які мають бути розплановані та використані у проекті а також процес введення нового ресурсу зображено на Рисунку 3.4.

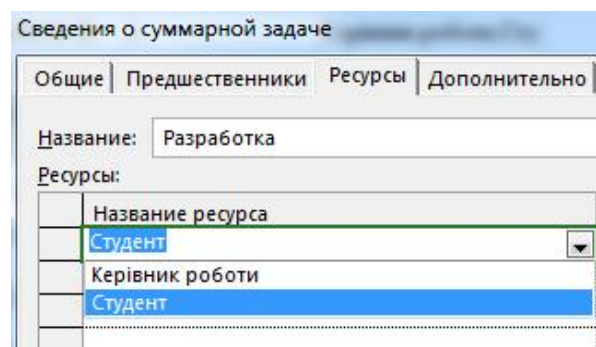


Рисунок 3.3 – Обрання ресурсу для виконання обраної задачі

Название ресурса	Тип	Кр на:	Макс. едини	Стандартна ставка	Ставка сверхуроч	Начисление	Базовый календарь	Код
Студент	Трудовой ПА		100%	\$9.38/ч	\$14.07/ч	По окончании	Fish	
Керівник роботи	Трудовой К		100%	\$4.06/ч	\$6.09/ч	По окончании	Fish	

Рисунок 3.4 – Таблица ресурсов у програмі Microsoft Project

3.2 Дослідження плану проекту на симуляцію запізнення виконання задач

Припустимо, що студент не дотримався плану роботи, який вони узгодили з керівником. Припустимо, що аналіз роботи почав виконуватися на 160 днів пізніше, ніж вказано у плані. Тоді ми маємо діаграму Ганта, представлену на Рисунку 3.5. Через структуру плану проекту усі наступні завдання зазнають зміщення. Оскільки даний проект має особливість у рамках виконання, а саме він повинен бути виконаний чутко до вказаного строку, нам потрібно вносити корективи у план проекту.

Так як ми не можемо завершити проект пізніше наведеного терміну (це приведе до повного провалу проекту), нам потрібно прости корегування усіх наступних задач для того, щоб повернутися у часові рамки проекту.

Задля цього, по-перше, потрібно скоротити час на виконання усіх наступних завдань таким чином, щоб останній етап припадав на узгоджену дату закінчення проекту. Також є ще одна ключова дія, а саме «Проходження процедури антиплагіату», так як вона теж повинна бути виконана у відведений строк.

Однак скорочення часу призведе до того, що деякі завдання будуть виконуватися одночасно або в один день.

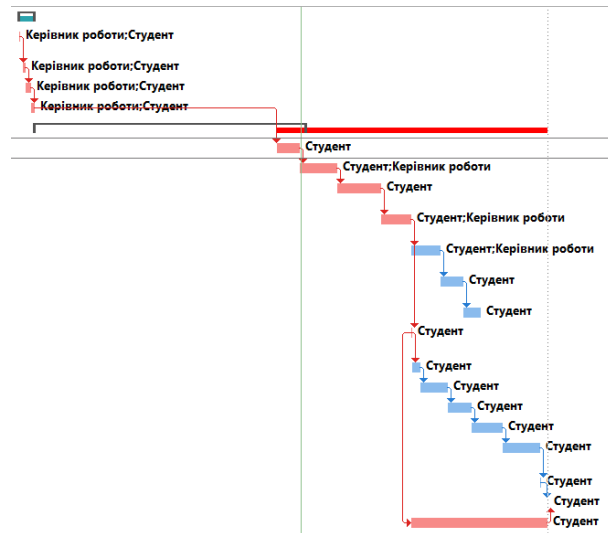


Рисунок 3.5 – Діаграма Ганта при внесенні затримання задачі

Отже у нашому розпорядженні період з 21.11 по 15.12, так як після проходження антиплагіату не можна змінювати пояснювальну записку, та період з 15.12 по 18.12, коли ще можна допрацювати графічні матеріали.

Якщо абстрагуватися від предметної області, то для компенсації затримки виконання певного завдання є кілька рішень. По-перше, при виникненні такої ситуації можна провести зустріч із замовником та обговорити нові часові рамки проекту та узгодити відшкодування за затримку. Другий варіант полягає у залученні додаткових ресурсі на проект задля збільшення продуктивності та можливості скорочення часу роботи над конкретним завданням.

Повертаючись до нашого конкретного проекту, перший варіант не задовольняє особливостям проекту, так як він повинен бути виконаний до чітко відведеного строку. Отже для вирішення поставленої задачі ми обираємо другий варіант.

По-перше, нам треба скоротити терміни виконання усіх задач після тої, яка була виконана із затримкою задля того, щоб вписатися у відведений час. По-друге, потрібно назначити додаткові ресурси у вікні, зображеному на Рисунку 3.4. Після застосування цих дій та перепланування проекту діаграма Ганта має вигляд, представлений на Рисунку 3.5

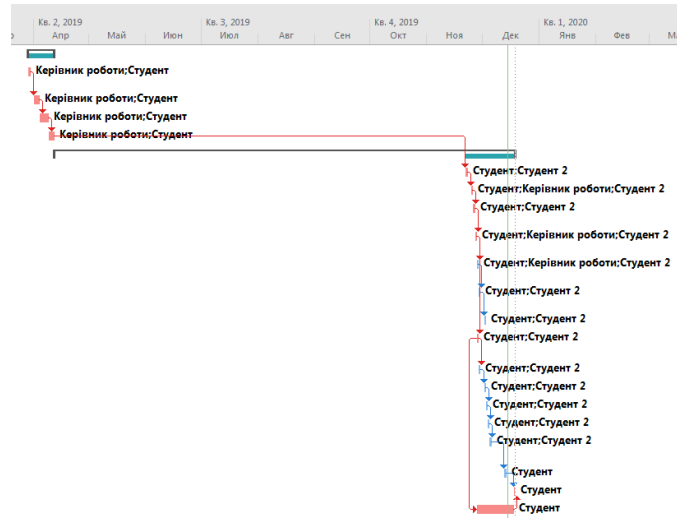


Рисунок 3.5 – Діаграма Ганта після внесення змін у терміни виконання задач

4. РЕАЛІЗАЦІЯ УДОСКОНАЛЕНОГО МЕТОДУ КЛАСТЕРІАЦІЇ ТРАС ЛОГУ

У даній роботі удосконалюється плагін для кластеризації трас та дослідження конфігурацій. Даний плагін розроблений для платформи дослідження процесів ProM.

ProM – це платформа для алгоритмів дослідження процесів з відкритим кодом, що дозволяє користувачам та розробникам користуватись алгоритмами process mining, які прості у використанні та є простими для розширення їх можливостей.

Обраний для вдосконалення плагін дозволяє проводити кластеризацію трас з анотаціями даних та дослідження дерев процесів для добірки цих кластерів. На вхід плагін приймає єдиний журнал подій та одне або декілька дерев процесів. На початку роботи з плагіном нас зустрічає вікно майстра, яке складається з п'яти кроків:

- Привітання: На цьому кроці показується текст із привітанням та пояснюється призначення плагіну;
- Загальне: Вибір максимальної кількості ядер процесора та класифікатор подій. Класифікатор подій визначає які атрибути подій визначають клас (тип дії);
- Кластеризація: Налаштування виміру незбіжності, вибір та налаштування алгоритмів кластеризації;
- Анотування: Цей крок містить алгоритм анотування, добірку атрибутів даних у журналі подій, які можуть бути використані у анотуванні.
- Оцінка: На останньому кроці ваги та головні значення встановленні для оцінки дерева конфігурованих процесів.

Усі налаштування окрім класифікатора подій, можуть бути змінені. Після завершення роботи майстра, ведеться розрахунок результатів та візуалізація

даних. На Рисунку 4.1 наведено обробку журналу подій, частину якого було показано у Розділі 1. Вікно візуалізації має наступні секції:

- Settings: Дане вікно має наступні кнопки: «General settings», «Clustering settings», «Annotation settings» та «Evaluation settings», що відкривають відповідні вікна майстра. Усі налаштування, окрім «Evaluation settings» застосовуються негайно, візуалізація оновлюється із новими результатами. Оцінка є важкою для процесора операцією та не виконується до натискання кнопки «Evaluate preferred clusters»

- Hierarchy overview: Це вікно відображає анотовану ієрархічну кластеризацію. Кожен кластер представляє собою вузол у вигляді кола на графі. Якщо вузол має зелений колір, це вказує на те, що кластер є вподобаним, тобто він приймається за журнал подій у дослідженні конфігурованого процесу. Вподобаний статус кластера може бути увімкнений шляхом подвійного натискання на вузол. Одиночний клік відкриває «Cluster inspector».

- Cluster inspector: Після вибору вузла у ієрархічній кластеризації у цьому вікні відображається вигляд усіх трас та відповідних подій. Кластер може бути переданий як об'єкт XLog до робочого простору ProM за допомогою натискання кнопки «Push as XLog to workspace». XLog є імплементацією журналу подій у ProM

- Tree overview: Після натискання кнопки «Evaluate preferred clusters» відкриваються конфігурації дерев вхідного процесу, а характеристики якості обчислюються відповідно до параметрів оцінки. На цій панелі впорядковано відображаються деревоподібні технологічні дерева (найбільш доцільні в першу чергу). У списку показано загальний бал придатності дерева, що настроюється, та бали за бажаний кластер. Після натискання на одне з налаштованих дерев процесу воно візуалізується на панелі «Tree inspector»

- Tree inspector: На цій панелі показано обране дерево конфігурованих процесів. За допомогою вікна вибору у заголовку дерева процесів можна налаштувати для кожного з бажаних кластерів. Дерево процесу можна перенести на робочу область ProM, натиснувши на кнопку «Push to workspace».

The screenshot displays the ProM 6 software interface for a simulated log analysis. At the top, there's a 'DRI Running instance' header and a 'Settings' button. Below this, a control panel includes 'General settings', 'Clustering settings', 'Annotation settings', and 'Evaluation settings'. The main workspace is divided into five panels:

- Hierarchy overview:** Shows a clustering hierarchy tree. Node C1 is the root, with children C2 and C461. Annotations include 'Payment = PACKAGE' for C2, and 'Payment = AFTER' and 'Payment = BEFORE' for C461.
- Cluster inspector:** Provides details for the selected cluster (259). It includes a list of instances (e.g., 693, 974, 586), a table of 99 events with attributes A, D, C, and E, and a detailed view of case 259 with attributes like 'Payment: BEFORE', 'Result: SUCCESS', and 'description: Simulated process ins'.
- Tree overview:** Shows an overview of process trees with an 'Evaluate preferred clusters' button. It displays overall fitness values for different clusters (e.g., #1 - Overall fitness 0.8275).
- Tree inspector:** Visualizes the selected process tree (C2). It shows a complex tree structure with nodes A, B, X, and E, and various operators like AND, OR, and XOR.

Рисунок 4.1 – Візуалізація вікна плагіну для симульованого логу робочого прикладу. Вікно складається з п'яти панелей: «Settings», «Hierarchy overview», «Cluster inspector», «Tree overview» and «Tree inspector».

Плагін, який використовується у даній роботі, може бути легко розширений за допомогою наступних елементів:

- Виміри несхожості;
- Алгоритми кластеризації;
- Алгоритми анотування.

Розширення додається за допомогою створення класу, який є розширенням одного з базових класів. Для включення розширень у інтерфейсі потрібно додати додаткові анотації. Інтерфейс автоматично шукає кореневий пакет розширень плагіна. Для активації розширення в графічному інтерфейсі, воно повинно бути розміщено у субпакеті `org.processmining.plugins.yvo`. Кожна анотація інтерфейсу

містить список `UISetting`, що дозволяє розробнику обрати функції, які можуть налаштовуватись в інтерфейсі. Далі пояснено про кожен варіант розширення у подробицях:

– **Виміри несхожості:** Для того, щоб додати новий вимір розбіжності, розробнику потрібно створити новий клас, який розширює `AbstractTraceDistanceFunction`. Цей абстрактний клас впроваджує імплементацію методу `distance(t1, t2)`, який повинен обчислювати відстань між двома трасами. Відстань розраховується паралельно, тому кожен вимір розбіжності повинен бути у своєму потоці. Також фреймворком передбачено, щоб усі виміри несхожості були симетричними. Замість використання `AbstractTraceDistanceFunction` рекомендується підходити до набору функцій для розширення класу `AbstractFeatureSetTraceDistanceFunction`. Цей клас реалізує функцію відстані як евклідову відстань двох наборів ознак. Під час розширення цього класу розробнику потрібно лише реалізувати метод `createFeatures (t1)`, який повинен повернути функції для заданого сліду.

– **Алгоритми кластеризації:** Для того, щоб додати новий алгоритм кластеризації, слід створити новий клас, який розширює алгоритм `AbstractClusteringAlgorithm`. Цей клас примушує реалізувати метод `clusterHierarchical ()`, який повинен виконувати фактичну кластеризацію та повертати ієрархію. Метод `clusterHierarchical ()` не має жодних параметрів, але натомість має доступ до об'єкта центрального сховища, який забезпечує журнал подій та відстань відстеження. Однією з найвідоміших рамок видобутку даних з відкритим кодом є ELKI. ELKI містить реалізацію для більшості відомих алгоритмів кластеризації. Щоб дозволити швидке використання ELKI, розробник може замість цього поширитись на `AbstractElkiClusteringAlgorithm`, який забезпечує основну функцію обгортання.

– **Алгоритми анотування:** Новий алгоритм анотації додається шляхом розширення класу `AbstractAnnotatorAlgorithm`. Цей абстрактний клас примушує реалізувати метод анотації (кластеризації), який повинен додавати вирази анотацій до заданої колекції кластерів. Weka - це ще одна відома рамка

вилучення даних з відкритим кодом. Weka має реалізацію для більшості відомих алгоритмів класифікації. Щоб дозволити швидке використання Weka, розробник може замість цього поширитись на `AbstractWekaAnnotatorAlgorithm`, який забезпечує основну функцію обгортання.

Кожне розширення має доступ до об'єкта `CentralRepository`. Цей об'єкт виступає в якості центрального сховища інформації, що використовується в плагіні, тобто він зберігає конфігурацію, журнал подій та основну інформацію цього

журнал подій, проміжні результати різних етапів та контексти дерева процесів. Конфігурація - це сховище ключових значень, в якому рядкові клавіші відображають об'єкти довільних типів, наприклад, кількість корисних процесорів

ядра представлені ключовим `general.cpu.cores` і цілим значенням. Контексти дерева процесів представлені об'єктами `ModelRepository`, які зберігають дерево процесу, (послідовні) вирівнювання та частково впорядковані вирівнювання.

У даній роботі удосконалення відноситься саме до алгоритму кластеризації, а саме впровадження анованої ієрархічної кластеризації, який ми описали у розділі 2.2. завдяки такому удосконаленню ми позбулися підходу грубої сили, коли конфігураційна модель враховувала ітеративні та паралельні процеси, що утворювало велику кількість вузлів та зв'язків, які по своїм змістом одною и тою ж самою послідовністю дій. Саме тому в цій роботі був використаний вдосконалений метод семантичних змістів, завдяки якому траси (послідовності подій) проходять процес кластеризації, тим самим утворюючи різні варіанти вузлів. Ті самі різні варіанти утворюють анотацію вузлів, кожен варіант якої є дослідженим кластером однорідних трас. Завдяки цьому кінцевий користувач може обрати конфігурацію моделі, яка задовільнює умовам застосування даної моделі та дозволить іншим користувачам з тої самої предметної області використовувати ту саму модель, але з різними конфігураціями.

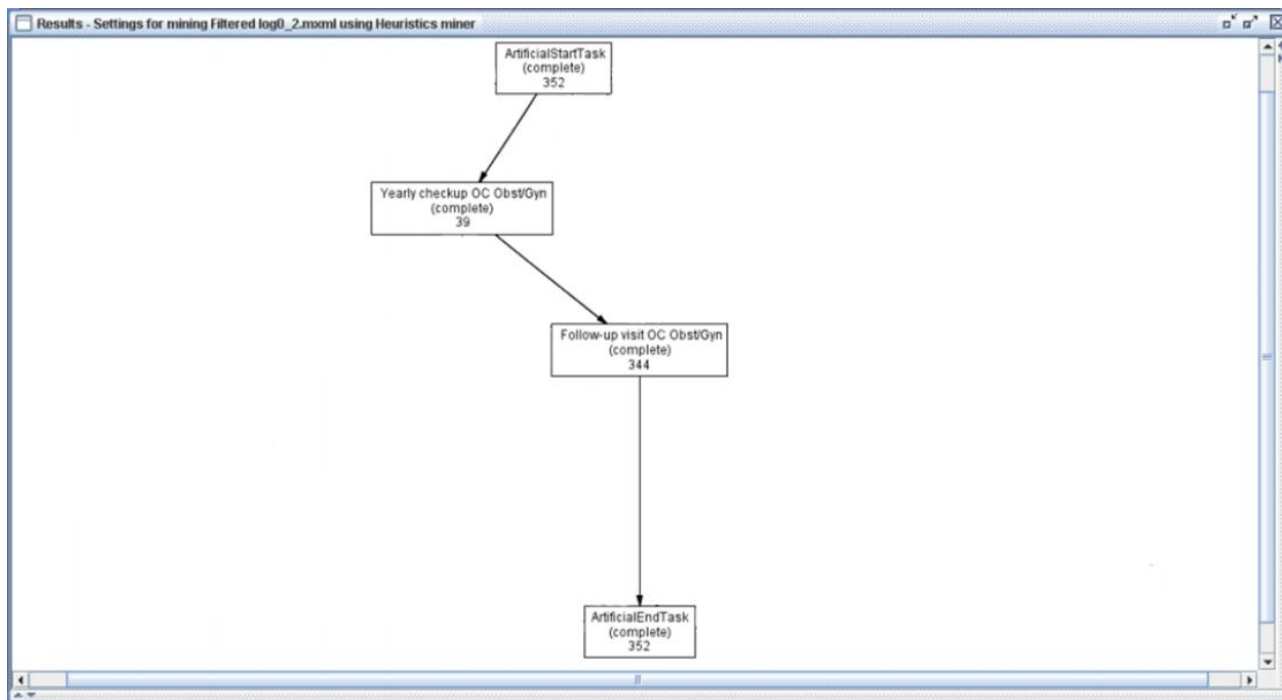


Рисунок 4.2 – Базисна модель, побудована парами трас із значенням

$$w_t \geq 1$$

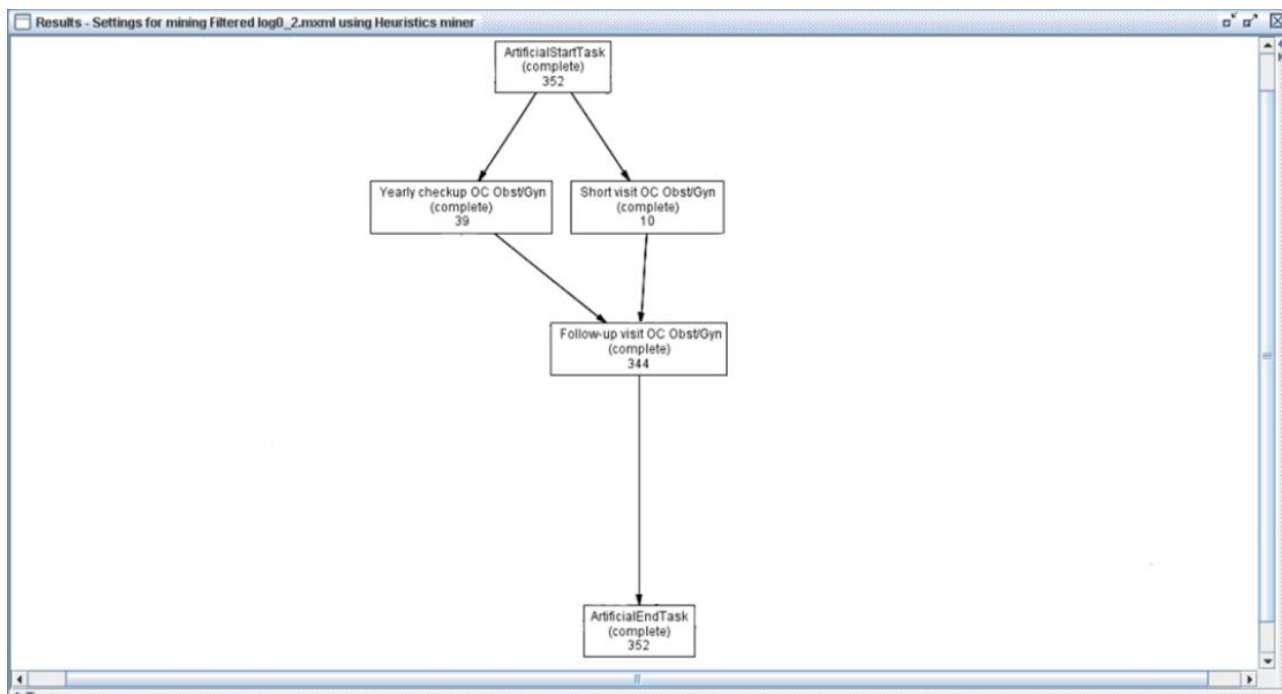


Рисунок 4.3 – Базисна модель, побудована парами трас із значенням

$$w_t \in (0,75; 1]$$

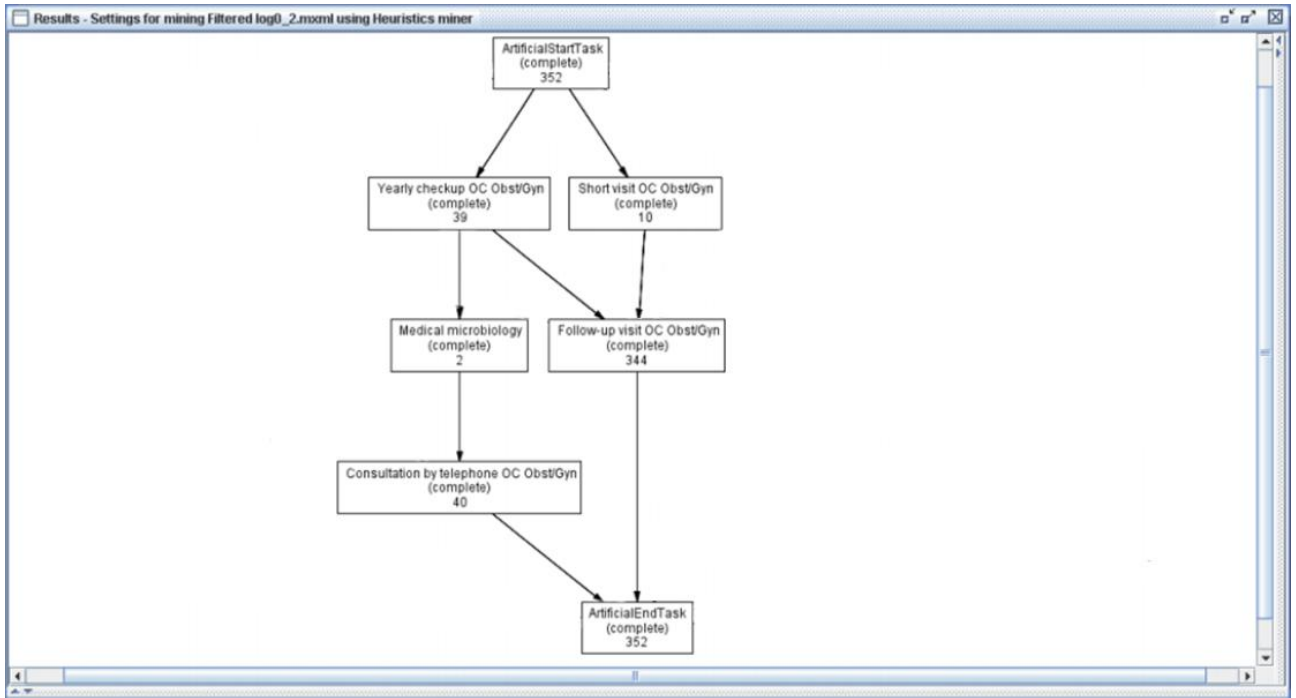


Рисунок 4.4 – Базисна модель, побудована парами трас із значенням $w_t \in (0,50; 0,75]$

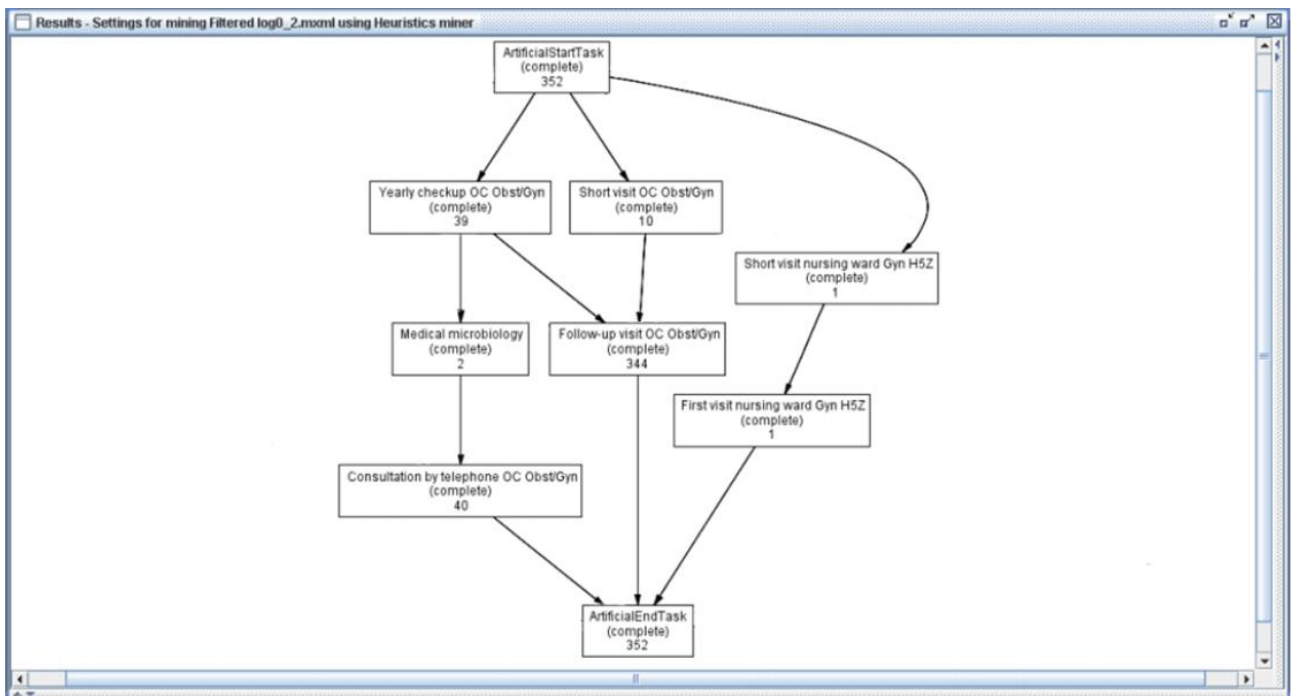


Рисунок 4.5 – Базисна модель, побудована парами трас із значенням $w_t \in (0; 0,50]$

ВИСНОВКИ

В результаті виконання магістерської атестаційної роботи був виконаний огляд методів кластеризації трас журналів подій у проектах інтелектуального аналізу процесів. Проаналізовано існуючі методи кластеризації логів.

Під час виконання атестаційної роботи були виконані наступні завдання: Аналіз задач process mining; Дослідження методів кластеризації для інтелектуального аналізу процесів; Удосконалення інтелектуального аналізу процесів на основі ієрархічної кластеризації трас журналу подій; Розробка плану проекту розробки модулю ієрархічної кластеризації трас; Експериментальна перевірка отриманих результатів.

Був проведений аналіз та удосконалення методів побудови моделей багатоваріантних процесів на основі аналізу логів шляхом ієрархічної кластеризації трас з подальшим виділення окремих конфігурацій таких процесів

Практичним результатом є алгоритм кластеризації трас для побудови моделей багатоваріантних процесів засобами

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки щодо розробки та оформлення магістерської атестаційної роботи за спеціальністю 122 Комп'ютерні науки (освітня програма «Управління проектами в галузі інформаційних технологій» освітньо-кваліфікаційного рівня «магістр» / упоряд.: К. Е. Петров, В. М. Левикин, С. Ф. Чалий та ін. Харків : Харків. нац. ун-т радіоелектроніки, 2019. 24 с.
2. Van der Aalst W. P. M. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. 1st ed. Springer Berlin Heidelberg, 2011. XVI, 352 p. 1, 7, 8, 11, 13 DOI: <https://doi.org/10.1007/978-3-642-19345-3>.
3. Van der Aalst W. Using Process Mining to Bridge the Gap between BI and BPM. *Computer*. December 2011, Vol. 44, Iss. 12. P. 77–80. DOI: <https://doi.org/10.1109/MC.2011.384>.
4. Чалий С. Ф., Прибильнова І. Б. Побудова ситуаційного представлення знань на основі аналізу логів. *Вісник Національного технічного університету «Харківський політехнічний інститут»*. Серія: Системний аналіз, управління та інформаційні технології. 2017. № 28 (1250). С. 70–73. DOI: <https://doi.org/10.20998/2079-0023.2017.28.11>.
5. Левикин В. М., Чалый С. Ф., Селезнев С. Н., Кривчикова А. А. Логическая модель представления бизнес-процесса для решения задач интеллектуального анализа процессов. *Бионика интеллекта*. 2010. № 1 (72). С. 24–28.
6. Чалий С. Ф., Богатов Є. О., Прибильнова І. Б. Методи формування упорядкованих по процесам трас журналу подій в задачах процесного управління. *Вісник Національного технічного університету «Харківський політехнічний інститут»*. Серія: Системний аналіз, управління та інформаційні технології. 2018. № 21 (1297). С. 43–47. DOI: <https://doi.org/10.20998/2079-0023.2018.21.08>.

7. De Medeiros A. K. A., Guzzo A., Greco G., van der Aalst W. M. P., Weijters A. J. M. M., van Dongen B. F., Sacc D. Process Mining Based on Clustering: A Quest for Precision // Business Process Management Workshops / eds: A. ter Hofstede, B. Benatallah, H. Y. Paik. Springer Berlin Heidelberg, 2008. P. 17–29. 15, 16 (Lecture Notes in Computer Science ; Vol. 4928). DOI: https://doi.org/10.1007/978-3-540-78238-4_4.

8. Чалий С. Ф., Богатов Є. О. Упорядкування трас логу на основі порівняння атрибутів подій в задачі побудови моделей бізнес-процесів засобами process mining // Информационные управляющие системы и технологии : материалы VII Междунар. науч.-практ. конф. (Одесса, 17–18 сент. 2018 г.) / Одес. нац. політех. ун-т, Одес. нац. мор. ун-т, Харків. нац. ун-т радіоелектроніки, Нац. ун-т кораблебудування ім. адмірала Макарова, Лодзький тех. ун-т. Одеса, 2018. С. 160–163.

9. Кузьма Є. А., Чалий С. Ф. Process Mining – інструмент менеджменту бізнес-процесів // Free and Open Source Software : матеріали X ювілей. Міжнар. наук.-практ. конф. (Харків, 20–22 листоп. 2018 р.) / М-во освіти і науки України, Харків. нац. ун-т будівництва та архітектури. Харків, 2018. С. 75.

10. Buijs J. C. A. M., van Dongen B. F., van der Aalst W. M. P. On the Role of Fitness, Precision, Generalization and Simplicity in Process Discovery // On the Move to Meaningful Internet Systems: OTM 2012 / eds: R. Meersman, H. Panetto, T. Dillon et al. Springer Berlin Heidelberg, 2012. P. 305–322. 8, 11 (Lecture Notes in Computer Science ; Vol. 7565). DOI: https://doi.org/10.1007/978-3-642-33606-5_19.

11. OMG. Business Process Model and Notation (BPMN). Object Management Group, dtc/2010-06-05edition, 2010. 8 Business Process Model and Notation (BPMN). Ver. 2.0.2 : December 2013 / Object Management Group. URL: <https://www.omg.org/spec/BPMN/2.0.2/PDF> (дата звернення: 02.06.2019).

12. Van der Aalst W. M. P. Formalization and Verification of Event-driven Process Chains. Eindhoven: Technische Universiteit Eindhoven, 1998. (Computing Science Reports ; 98/01). 8 URL: <https://research.tue.nl/en/publications/formalization->

and-verification-of-eventdriven-process-chains(22be58e3-b406-4c77-9d98-1b6cb525c0f4).html (дата звернення: 22.04.2019).

13. Leemans S. J. J., Fahland D., van der Aalst W. M. P. Discovering Block-structured Process Models From Event Logs – A Constructive Approach // Application and Theory of Petri Nets and Concurrency / eds: J. Colom, J. Desel. Springer Berlin Heidelberg, 2013. P. 311–329. 8, 59, 60, 61 (Lecture Notes in Computer Science ; Vol. 7927). DOI: https://doi.org/10.1007/978-3-642-38697-8_17.

14. Buijs J. C. A. M., van Dongen B. F., van der Aalst W. M. P. Mining Configurable Process Models from Collections of Event Logs // Business Process Management / eds: F. Daniel, J. Wang, B. Weber. Springer Berlin Heidelberg, 2013. P. 33–48. 1, 9, 12, 33, 34, 35, 39, 52, 53, 66 (Lecture Notes in Computer Science ; Vol. 8094). DOI: https://doi.org/10.1007/978-3-642-40176-3_5.

15. Gottschalk F., van der Aalst W. M. P., Jansen-Vullers M. H., La Rosa M. Configurable workflow models. *International Journal of Cooperative Information Systems*. 2008. Vol. 17, Iss. 2. P. 177–221. 1, 9, 31 DOI: <https://doi.org/10.1142/S0218843008001798>.

16. Rosemann M., van der Aalst W. M. P. A configurable reference modelling language. *Information Systems*. 2007. Vol. 32, Iss. 1. P. 1–23. 9 DOI: <https://doi.org/10.1016/j.is.2005.05.003>.

17. Schunselaar D. M. M., Verbeek E., van der Aalst W. M. P., Rajjers H. A. Creating Sound and Reversible Configurable Process Models Using CoSeNets // Business Information Systems / eds: W. Abramowicz, D. Kriksciuniene, V. Sakalauskas. Springer Berlin Heidelberg, 2012. P. 24–35. (Lecture Notes in Business Information Processing; Vol. 117). 9, 33 DOI: https://doi.org/10.1007/978-3-642-30359-3_3.

18. Hallerbach A., Bauer T., Reichert M. Capturing Variability in Business Process Models: The Provop Approach. *Journal of Software Maintenance and Evolution: Research and Practice*. 2010. Vol. 22, Iss. 6–7. P. 519–546. DOI: <https://doi.org/10.1002/spip.446>.

19. Buijs J. C. A. M., van Dongen B. F., van der Aalst W. M. P. A genetic algorithm for discovering process trees // 2012 IEEE Congress on Evolutionary Computation. Brisbane, QLD, Australia, 10–15 June 2012. IEEE, 2012. P. 1–8. 12, 34 DOI: <https://doi.org/10.1109/CEC.2012.6256458>.

20. Gottschalk F., van der Aalst W. M. P., Jansen-Vullers M. H. Mining Reference Process Models and Their Configurations // On the Move to Meaningful Internet Systems: OTM 2008 Workshops /eds: R. Meersman, Z. Tari, and P. Herrero, Springer Berlin Heidelberg, 2008. P. 263–272. 33 (Lecture Notes in Computer Science ; Vol. 5333). DOI: https://doi.org/10.1007/978-3-540-88875-8_47.

21. Левыкин В. М., Чалая О. В. Разработка модели многовариантного знание-емкого бизнес-процесса. *Вісник Херсонського національного технічного університету*. 2016. № 4 (59). С. 195–202. URL: [http://kntu.net.ua/ukr/content/download/52042/306877/file/ВІСНИК%20№%204\(59\).pdf](http://kntu.net.ua/ukr/content/download/52042/306877/file/ВІСНИК%20№%204(59).pdf) (дата звернення: 07.05.2019).

22. Левикін В. М., Чала О. В. Виділення реляційних залежностей бізнес-процесу на основі аналізу його логу. *Наукоємні технології*. 2016. Т. 32, № 4. С. 405–409. URL: <http://jrn1.nau.edu.ua/index.php/SBT/article/view/11183/14849> (дата звернення: 23.08.2019).