

Український державний університет науки і технологій

Факультет Комп'ютерні технології та системи
Кафедра Комп'ютерні інформаційні технології

Пояснювальна записка
до кваліфікаційної роботи
магістра

на тему: «Дослідження технологій та засобів прогнозування користувацьких дій при конструюванні бізнес процесів виробництва»
за освітньою програмою **12 Інженерія програмного забезпечення**
зі спеціальності: **121 Інженерія програмного забезпечення**

Виконав: студент групи «ПЗ2221»

/Володимир ПОГРЕБНЯК/

Керівник:

/доц. Вадим ГОРЯЧКІН/

Нормоконтролер:

/ доц. Світлана ВОЛКОВА/

Засвідчую, що у цій роботі немає запозичень з праць інших авторів без відповідних посилань
Студент

(підпис)

Міністерство освіти і науки України
Український державний університет науки і технологій

Факультет: Комп'ютерних технологій і систем
Кафедра: Комп'ютерні інформаційні технології
Рівень вищої освіти: магістр
Освітня програма: Інженерія програмного забезпечення
Спеціальність: Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ
Завідувач кафедри КТС
_____ Вадим ГОРЯЧКІН
_____ січень 2022 р.

ЗАВДАННЯ

На кваліфікаційну роботу _____ Магістр
студенту Погребняку Володимир Миколайовичу

1. Тема дипломної роботи: «Дослідження технологій та засобів прогнозування користувацьких дій при конструюванні бізнес процесів виробництва»
Керівник роботи: Горячкін Вадим Миколайович
затверджені наказом 1334 ст від 05.12. 2022 року
2. Строк подання студентом роботи 17.01. 2024 року
3. Вихідні дані до дипломної роботи:
Побудована бізнес модель.
4. Зміст пояснювальної записки (перелік питань до розробки):
 - 4.1. Аналітична частина.
5. Перелік демонстраційного матеріалу:
 - 5.1. презентація;
 - 5.2. демонстраційне відео.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Вступ	05.03.22 – 21.04.22	
2	Аналіз сучасного стану дослідження проблеми за науковими літературними джерелами	03.06.23 – 07.07.23	
3	Аналіз сучасного стану програмно-апаратного забезпечення, яке потребує вдосконалення для вирішення проблем дослідження	10.07.23 – 15.07.23	
4	Постановка задачі, технічне завдання	16.07.23 – 29.07.23	30%
5	Виконання досліджень	07.08.23 – 30.09.23	60%
6	Оформлення тез доповідей	30.10.23 – 05.12.23	
7	Оформлення пояснювальної записки	01.05.23 – 31.12.23	
8	Розробка демонстраційних матеріалів	05.01.24 – 14.01.24	100%
9	Подання кваліфікаційної роботи до кафедри	15.01.24 – 17.01.24	
10	Захист кваліфікаційної роботи на засіданні Екзаменаційної комісії	24.01.24	

Студент _____ /Володимир ПОГРЕБНЯК/
 Керівник роботи _____ / Вадим ГОРЯЧКІН /

РЕФЕРАТ

Об'єкт дослідження - процес автоматизації прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва.

Мета роботи - розробка програмного забезпечення для прогнозування користувацьких дій під час конструювання схем бізнес-процесів у сучасному виробничому середовищі, яка має важливе значення в зв'язку з поєднанням динамічного розвитку технологій та необхідності вдосконалення продуктивності й ефективності виробництва.

Методи дослідження - в даній роботі використовуються методи теоретичного дослідження: аналіз, синтез та узагальнення. Також застосовувалися емпіричні методи: експеримент, спостереження, порівняння, вимірювання та опис.

Актуальність теми - створення програмного забезпечення для управління різними видами господарської діяльності підприємства під час якого дуже велику частину витрат робочого часу займає моделювання бізнес-процесів. Потрібно зазначити, що в основі розробки головне місце посідає стандартизований процес збору вимог до системи та під час створення моделі, де етапність та правильність застосування дає прогрес у проєктуванні та реалізації проєкту автоматизації.

В зв'язку з цим актуальним є розв'язання проблеми пришвидшення конструювання релевантних бізнес моделей програмного забезпечення за рахунок автоматизації процесу прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва.

Предмет дослідження - є адаптація програмного забезпечення до змінних умов ринку та вимог споживачів, прогнозування їхньої взаємодії з бізнес-системою, яка стане вирішальним фактором у вирішенні викликів, пов'язаних з оптимізацією процесів, підвищенням рівня персоналізації та покращенням якості продукції чи послуг.

Наукова новизна - програмне забезпечення для прогнозування користувацьких дій стає ключовим інструментом для підвищення продуктивності та конкурентоздатності виробничих підприємств у сучасному

економічному середовищі, сприяючи їх успішному розвитку за рахунок автоматизації процесу. Цей спосіб, на відміну від існуючих, дозволить прискорити конструювання схем бізнес- процес не менше ніж на 20%.

Практична цінність - програмне забезпечення для прогнозування користувацьких дій в період воєнного стану та в подальшому у мирний час дозволить підняти економіку країни на високий рівень. Виробничі підприємства за рахунок програмного забезпечення прогнозування матимуть можливість на ринку економіки та праці стати конкурентоспроможними та продуктивними.

Структура та обсяг роботи. Магістерська дисертація складається зі вступу, п'яти розділів, висновків та додатків.

Вступ - розкриває загальну характеристику роботи, описує сучасний стан проблеми, обґрунтовує актуальність напрямку досліджень, поставлену мету та шляхи вирішення цілей дослідження.

Перший розділ - характеризує:

- напрямки розповсюдження машинного навчання;
- загальну ідею опису прецедентів інформаційної одиниці;
- зміст використання та структури для зберігання прецедентного опису.

Другий розділ - описує вимоги до дослідження технологій та засобів програмного забезпечення прогнозування користувацьких дій під час конструювання схем бізнес-процесів.

Третій розділ - презентує особливості програмного забезпечення прогнозування користувацьких дій при конструюванні схем бізнес-процесів, де детально описано принципи роботи та реалізація його алгоритму.

Четвертий розділ - представляє результати тестування розробленого програмного забезпечення прогнозування користувацьких дій під час конструювання схем бізнес-процесів, представлено аналіз отриманих результатів та наведено шляхи до вдосконалення програмного забезпечення.

П'ятий розділ - знайомить з бізнес моделлю, яка дозволяє представити на ринку повноцінний програмний продукт із використанням представлених у роботі напрацювань.

Висновок розкриває - отримані результати даної магістерської роботи.

Додатки - є носіями інформації, а саме: наведено дерево проблем, порівняння потенційних користувачів та канву згідно запропонованої бізнес-моделі.

Ключові слова: прогнозування, програмне забезпечення, дії користувача, схеми бізнес-процесів, виробництво.

ЗМІСТ

Вступ.....	10
1 Характеристика роботи існуючих технологій прогнозування користувацьких дій при роботі з веб-додатками.....	11
1.1 Види машинного навчання та їх характеристика	12
1.2 Опис прецедентів інформаційної одиниці.....	12
1.1.1 Виділення ознак.....	13
1.1.3 Зниження розмірності ознак, як робота процесів виділення та вибирання.	15
1.3 Типові задачі та напрямки використання прецедентного опису.....	15
1.4 Структури зберігання даних при реалізації опису ознак	16
1.4.1 Масив.....	16
1.4.2 Зв'язний список	17
1.4.3 Дерево.....	17
1.4.4 Хеш-таблиця	18
1.5 Висновки за розділом 1	18
2 Постановка задачі дослідження, опис моделі, критерії оцінки.....	19
2.1 Вимоги до способу при конструюванні схем бізнес процесів	19
2.2 Особливості прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва	21
2.3 Типи прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва	23
2.3.1 Аналізуючий модуль.....	23
2.3.2. Ідентифікуючий модуль	25
2.3.3.Прогнозуючий модуль	26
2.4 Висновки до розділу 2	30
3 Дослідження алгоритмів програмного забезпечення прогнозування користувацьких дій	32
3.1 Розкриття змісту «Архітектура».....	32
3.1.1 Алгоритм дії даної структури:	33

	9
3.2 Архітектура взаємодії складових частин	33
3.3 Складові архітектури програмного забезпечення	34
3.4 Характеристика інтерфейсу	37
3.5 Опис програмного застосунку	38
3.6 Висновки до розділу 3.	43
4 Оцінки отриманих результатів	45
4.1 Динамічна довжина ланцюгу користувачьких дій та динамічна максимальна вкладеність дерева прецедентів	46
4.2 Події "повернення до попереднього стану"	47
4.3 Можливість демонстрації результатів прогнозованої дії	47
4.4 Висновок до розділу 4	48
5 Побудова бізнес-моделі	49
5.1 Аналіз проблеми.....	49
Висновки	56
Список використаних джерел	58
ДОДАТОК А.....	1
ДОДАТОК Б	Помилка! Закладку не визначено.
ДОДАТОК В.....	13
ДОДАТОК Г	22

ВСТУП

Розробка програмного забезпечення для прогнозування користувацьких дій під час конструювання схем бізнес-процесів у сучасному виробничому середовищі має важливе значення в зв'язку з поєднанням динамічного розвитку технологій та необхідності вдосконалення продуктивності й ефективності виробництва.

Застосування такого програмного забезпечення дає можливість адаптуватися до змінних умов ринку та вимог споживачів, прогнозуючи їхні взаємодії з бізнес-системою. Розуміння користувацької поведінки стає вирішальним фактором у вирішенні викликів, пов'язаних з оптимізацією процесів, підвищенням рівня персоналізації та покращенням якості продукції чи послуг.

Розробка моделей прогнозування користувацьких дій є складним процесом, який включає кілька ключових етапів для ефективного прогнозування поведінки користувачів (збір інформації з різних джерел, вибір моделей, валідація та оцінка точності, реалізація та впровадження моделі для прогнозування, тощо) Цей цикл може бути ітеративним, з постійним удосконаленням моделі через оновлення даних, додавання нового функціоналу та вдосконалення алгоритмів.

Відповідне програмне забезпечення надає можливість не лише прогнозувати користувацькі дії, але й попередньо аналізувати та планувати свої дії відповідно до цих прогнозів, оперативно мінімізуючи ризики, збільшуючи ефективність та забезпечуючи конкурентні переваги. Зокрема у виробничому середовищі, де кожний крок у бізнес-процесі має вагому значимість, вірогідність передбачити дії користувачів може позитивно вплинути на планування ресурсів, оптимізацію виробництва та зниження витрат.

Таким чином, програмне забезпечення для прогнозування користувацьких дій стає ключовим інструментом для підвищення

продуктивності та конкурентоздатності виробничих підприємств у сучасному економічному середовищі, сприяючи їх успішному розвитку.

1 ХАРАКТЕРИСТИКА РОБОТИ ІСНУЮЧИХ ТЕХНОЛОГІВ ПРОГНОЗУВАННЯ КОРИСТУВАЦЬКИХ ДІЙ ПРИ РОБОТІ З ВЕБ- ДОДАТКАМИ

Сучасний світ прогресує в бік штучного інтелекту, який витісняє людську діяльність, проте, ми є користувачами веб додатків, які допомагають отримати швидкі результати через інформаційні сервери.

Автоматизація людської діяльності при взаємодії з програмними застосунками є доволі складною галуззю, що включає в себе велику кількість часткових випадків, що мають унікальні вимоги до поведінки програмного застосунку, який не діє за звичайним структурним алгоритмом.

Для вирішення поставленої задачі необхідно використовувати більш гнучкі методи, що дозволять прогнозуючому програмному застосунку розширювати власний набір «знань» під час взаємодії із користувачем.

Так як основною метою автоматизації є саме побудова схем бізнес- процесів виробництва, то зазначена вище умова, виставлена до способу прогнозування користувацьких дій, також повинна бути доповнена наступними складовими [1]:

прогнозування користувацьких дій пов'язане з діями виконаних користувачем;

дія повинна мати чіткий безпомилковий кінцевий результат в рамках моделювання схем бізнес-процесів виробництва;

на підставі набору попередніх користувацьких дій здійснюється розрахунок набору прогнозованих дій;

результат виконання прогнозованої дії не може виходити за рамки результатів дій, виконаних користувачем до часу прогнозування;

чинна схема бізнес процесів є результатом виконання прогнозованої дії;

Як відомо, кожен проєкт з моделювання схем бізнес-процесів виробництва є унікальним, та має власні тонкості та обмеження. Саме тому зазначені умови є достатньо обмежуючими для прогнозуючого програмного застосунку в плані надання вільного формату кінцевого результату прогнозованих дій.

Зазначені умови є необхідною мірою для гарантування чіткості та відсутності помилок в результатах прогнозованих дій, які може застосувати користувач в рамках свого унікального проекту, тому найбільш вдалим вибором для реалізації рішення є використання штучного інтелекту, а саме його часткового випадку – алгоритмів його машинного навчання.

1.1 Види машинного навчання та їх характеристика

Машинне навчання – клас методів штучного інтелекту, характерною рисою яких є не пряме вирішення задачі, а навчання в процесі застосування рішень багатьох подібних задач.

Розрізняють два основних типи навчання [2]:

- індуктивне навчання (прецедентний опис);
- дедуктивне навчання.

Проаналізувавши зазначені типи навчання, можна зазначити, що прецедентний опис використовується у випадках, коли необхідно знайти закономірності в наборі даних та побудувати подальший алгоритм обробки початкових даних, а дедуктивне навчання використовується у випадках формалізації експертних даних та переносі їх у електронний вигляд для подальшого використання або вдосконалення.

Тож, виходячи із наведених описів, можна зробити висновок, що для побудови алгоритму прогнозування користувацьких дій слід використовувати обробку користувацьких дій за допомогою машинного навчання з використанням прецедентного опису об'єктів, який базується на закономірності в наборі даних та дозволяє створити в майбутньому алгоритм обробки початкових даних.

1.2 Опис прецедентів інформаційної одиниці

Прецедентний опис – частіше за все, представлений числовими даними та використовується в розділі машинного навчання, n -вимірний вектор ознак. Для

опису ознак, як відомо саме числові показники використовуються для обробки та статистичного аналізу[3].

Наприклад, використання числових ознак можуть виступати як опис прецедентів аудіофайлів.

При подальшому дослідженні інформації кожні з ознак розширюються коефіцієнтами, що додаються до загального опису ознак та використовуються при аналізі та структуруванні ознак під час формування результатів машинного навчання. Прецедентний опис ознак вважається потужною технологією, чим дозволяє значно скоротити аналіз інформації в задачах високих розмірностей.

Загалом процес опису прецедентів інформаційної одиниці можна розділити на три підпроцеси:

- виділення ознак;
- вибір ознак;
- зниженні розмірності ознак.

Розглянемо більш детально кожний з підпроцесів.

1.1.1 Виділення ознак

Виділення ознак – процес перетворення вхідних даних алгоритму на скорочений набір ознак у випадку, коли ці дані признаються надлишковими та не можуть використовуватися для ефективного аналізу. В свою чергу, вхідними даними для процесу виділення ознак є набір даних вимірювань, з якого в подальшому будуються похідні значення, які мають бути ненадлишковими та інформативними в контексті роботи алгоритму [4].

Успішно виділеними ознаками вважаються ті, які мають достатню кількість необхідної інформації для виконання над ними, замість початкових надлишкових даних, заданої роботи із позитивним кінцевим результатом. Основною проблемою, яка виникає при виконанні процесу виділення ознак, є надмірне споживання алгоритмом машинної пам'яті та обчислювальних

потужностей. Для вирішення зазначеної проблеми існують спеціальні алгоритми, що пришвидшують виконання роботи та економлять використання ресурсів.

До найбільш популярних алгоритмів можна віднести наступні [4]:

- метод головних компонент;
- приховано-семантичний аналіз.

Наведемо скорочений опис кожного з алгоритмів.

Метод головних компонент – алгоритм, що використовує ортогональне перетворення множини спостережень з можливо пов'язаними змінними, та базується на логічному припущенні, що значення множини взаємозалежних ознак породжують деякий загальний результат [4].

Приховано-семантичний аналіз – метод обробки інформації, що дозволяє аналізувати взаємозв'язок між набором властивостей, що зустрічаються в інформаційних одиницях, шляхом створення семантичного

«словника» похідних властивостей, який в результаті алгоритму трансформується у набір ознак [5].

1.1.2 Вибір ознак

Вибір ознак – процес фільтрування вхідних ознак від недоречних, тобто тих, що є надлишковими та неінформативними, та в розрізі конкретного дослідження мають низьку корисність [6].

Такі дані можуть бути усунені без впливу на кінцевий результат, що є головною умовою під час процесу вибору ознак.

Наведемо основні методики, що використовуються в процесі вибирання ознак:

надання набору спрощеного вигляду для поліпшення інтерпретування алгоритмом або науковцем;

поліпшення узагальнення ознак через зменшення кількості кроків обробки.

Слід зазначити, що процеси виділення та вибирання можуть здатися подібними між собою, але це не так. Результатом процесу виділення ознак є модифіковані за певним правилом початкові набори даних, а результатом,

відповідно, вибирання ознак, є підмножина початкових ознак, що не піддається модифікуванню.

Власне, основним недоліком цього підходу є відсутність узагальнених алгоритмів, що використовуються для процесу вибору ознак, адже цей процес є дуже індивідуальним та формується під чітко поставлену задачу, тому єдиного рішення для виконання процесу вибирання ознак не існує.

1.1.3 Зниження розмірності ознак, як робота процесів виділення та вибирання.

Зниження розмірності опису ознак – процес, що комбінує в собі роботу процесів виділення та вибирання, результатом якого є об'єднані результати зазначених процесів в скороченому вигляді [7], а саме застосування до результату попередніх процесів алгоритмів стискання великого набору даних, результатом яких є модифікований вхідний ознаковий опис інформаційної структури та словник, який використовується для дешифрування модифікованого набору даних для більш чіткого розуміння дослідником.

1.3 Типові задачі та напрямки використання прецедентного опису

За допомогою прецедентного опису, що є складовою частиною алгоритмів машинного навчання, на теперішній час вирішується велика кількість задач на технічних напрямках діяльності. Основними задачами, притаманними технічній області, є :

- розпізнавання об'єктів за допомогою комп'ютерного зору;
- розпізнавання людської мови;
- розпізнавання печатного та непечатного тексту;
- фільтрування електронної пошти;
- категоризація документів.

Зазначені задачі належать до різних технічних категорій, та мають принципово різні за смисловим навантаженням початкові дані та результати, але

всі вони вирішуються за допомогою використання прецедентного опису до початкових даних задачі за методом вирішення задач «навчання з вчителем».

Проаналізувавши зазначений тип задач, можна відзначити, що суть її полягає в тому, що кожний прецедент представляє собою пару «об'єкт – відповідь», а завдання, яке ставиться до задачі – знайти функціональну залежність відповідей від описів об'єктів і побудувати алгоритм. Таким чином, побудований алгоритм повинен приймати на вході об'єкт з даними, а в результаті повертати побудовану модель, залежно від початкових даних [8].

1.4 Структури зберігання даних при реалізації опису ознак

Слід зауважити, що кожна з існуючих структур даних має власне, найбільш корисне з точки зору економії пам'яті, простоти та швидкодії, використання в задачі конкретного типу, але в іншій задачі може лише заважати в реалізації алгоритму, через що розв'язання задачі вийде надто складне та повільне.

До найбільш популярних структур даних можна віднести наступні:

- масив;
- зв'язний список;
- дерево;
- хеш-таблиця.

1.4.1 Масив

Масив – упорядкований набір даних, що використовується для зберігання елементів різних типів даних, що ідентифікується за допомогою одного чи більше індексів. Також масиви можуть відрізнятися порядком входження та взяття елементів: стек та черга. Стек реалізує правило

«перший зайшов – останній пішов», черга реалізує правило «перший зайшов, першим пішов» [10].

Так як масив – це найпростіша структура даних, що при великій кількості елементів стає складною та повільною в обробці, використовується вона в простих задачах будь-якого типу з невеликою кількістю початкових та кінцевих даних.

1.4.2 Зв'язний список

Зв'язний список – динамічна структура даних, що складається з окремих вузлів, кожен з яких зберігає набір даних та два посилання на попередній та/або наступний елементи списку [11].

До головних переваг зв'язних списків можна віднести структурну гнучкість, невеликі затрати часу на виконання операцій додавання та видалення елементів та здатність зберігати необмежений об'єм інформації. Серед недоліків зв'язних списків слід зазначити складність прямого доступу до елементів, важкість організації паралельного використання списку та кешування елементів.

З виявлених переваг та недоліків можна зробити висновок, що зв'язні списки прекрасно підходять під задачі з великими об'ємами початкових та кінцевих даних та невеликою кількістю елементів, що редагуються під час виконання.

1.4.3 Дерево

Дерево – це ієрархічна структура даних, емулююча деревовидну структуру, що складається з вузлів та ребер, що їх поєднують. По своїй суті, дерева являються зв'язним графами, що розповсюджуються в одному напрямку та мають неорієнтовані ребра однакової ваги [12].

Проаналізувавши основні властивості дерев, можна заявити, що ця структура є найбільш зручною для організації кінцевих даних при роботі з прецедентним описом через її гнучку структуру та швидкість редагування даних.

1.4.4 Хеш-таблиця

Хеш-таблиця – це структура даних, що дозволяє зберігати пари «ключ, значення» і виконувати над даними, що зберігаються три основні операції: додавання, видалення та пошук за ключем [13].

Загалом, хеш-таблиці мають високі показники швидкодії при виконанні з ними операцій додавання, видалення та пошуку елементів. Тому, зважаючи на принцип зберігання даних, хеш-таблиці можуть використовуватися в задачах з великою кількістю структурованих початкових даних.

Проаналізувавши основні переваги та недоліки зазначених структур даних, можна чітко виділити ті, що найбільш підходять для поставленої задачі прогнозування користувацьких дій, а саме: масив та дерево. Зазначені структури обрані через їх простоту в реалізації, ефективність та швидкість в обробці.

1.5 Висновки за розділом 1

Виходячи з вище викладеного матеріалу, вибір при конструюванні бізнес схем процесів зводиться до динамічного формування моделі поведінки конкретного користувача та подальшого аналізу побудованої моделі, використовуючи метод притаманний задачі «навчання з вчителем» через його динамічність та швидкодію.

Прецедентний опис ознак дозволяє значно скоротити аналіз інформації в задачах високих розмірностей та вважається на сьогодні потужною технологією. Через простоту реалізації, ефективність та швидкість в обробці слід обирати структури зберігання даних для алгоритму таких як масив даних та дерево.

2 ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ, ОПИС МОДЕЛІ, КРИТЕРІЇ ОЦІНКИ

2.1 Вимоги до способу при конструюванні схем бізнес процесів

Вимоги до способу та програмного забезпечення прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва були визначені через аналіз існуючих методів і програмних рішень для прогнозування користувацьких дій. Додатково було проведено онлайн анкетування серед потенційних користувачів програмного забезпечення. Анкетування включало питання про корисні функціональності програм, з урахуванням пріоритетів для кожної функціональної вимоги.

Програмне забезпечення має включати:

- графічний інтерфейс для взаємодії зі списком створених дій;
- автоматичний режим для виконання найбільш ймовірних дій, що базуються на попередніх виконаних операціях;
- можливість аналізу побудованої схеми бізнес-процесів для виявлення помилок;
- здатність відновлювати схему бізнес-процесів до попереднього стану після виконання прогнозованої дії;
- автоматичне створення списку прогнозованих користувацьких дій протягом однієї секунди після завершення попередньої дії;
- створення списку прогнозованих користувацьких дій, використовуючи послідовність виконаних користувацьких дій;
- використання оптимізованих, щодо швидкодії та використання пам'яті, структур даних для прогнозування користувацьких дій.

Після виконання будь-якої дії у моделюванні схеми бізнес-процесів, прогнозуючий модуль повинен надавати користувачеві прогнозований список наступних дій. Для прогнозування ймовірних дій використовується дерево прецедентів, яке будується на основі описів виконаних користувацьких дій та

послідовності останніх дій. Створення цього списку повинно вимірювати час не більше однієї секунди процесорного часу та оптимізуватися з точки зору використання пам'яті робочої машини користувача.

Наведений список містить в собі вимоги до способу прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва, а саме:

- побудова списку прогнозованих дій повинна відбуватися на основі послідовності виконаних користувацьких дій;
- побудова списку прогнозованих дій повинна займати не довше однієї секунди після завершення виконання користувачем попередньої дії;
- для побудови списку прогнозованих дій необхідно використовувати оптимізовані за використанням машиннихресурсів структури даних.

Прогнозуючий модуль після виконання будь-якої дії моделювання схеми бізнес-процесів повинен повертати користувачеві прогнозований список подальших дій. Прогнозування вірогідних дій повинно враховуватися на основі дерева прецедентів, що будується під час всього часу роботи із програмним забезпеченням на основі описів виконаних користувацьких дій, та послідовності декількох останніх дій. Побудова списку вірогідних користувацьких дій повинна займати не більше однієї секунди процесорного часу та бути оптимізованою в плані використання пам'яті робочої машини користувача. Також, перед додаванням інформації стосовно виконаної користувачем дії, актуальна схема бізнес-процесів виробництва повинна проходити перевірку на наявність помилок в структурі схеми.

Стосовно вимог до графічної частини слід зазначити, що програмне забезпечення повинно мати графічний інтерфейс для взаємодії із списком прогнозованих користувацьких дій та мати можливість повернення схеми бізнес-процесів виробництва до попереднього стану в будь-який момент роботи із додатком.

2.2 Особливості прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва

Основною задачею способу прогнозування є виконання обчислень у вигляді побудови дерева прецедентів та виконання пошуку входження ланцюга користувацьких дій в дереві. Зазначені обчислення є доволі складними, тому можуть займати відносно великий, до 1000 мілісекунд, проміжок часу роботи процесора. Оскільки запуск програмного забезпечення прогнозування користувацьких дій виконується на однопоточній платформі веб, під час виконання обчислень графічна частина програмного забезпечення прогнозування користувацьких дій буде блокувати свою роботу та не зможе взаємодіяти із користувачем.

Було побудовано дерево проблем зазначеним на рис. 2.1:

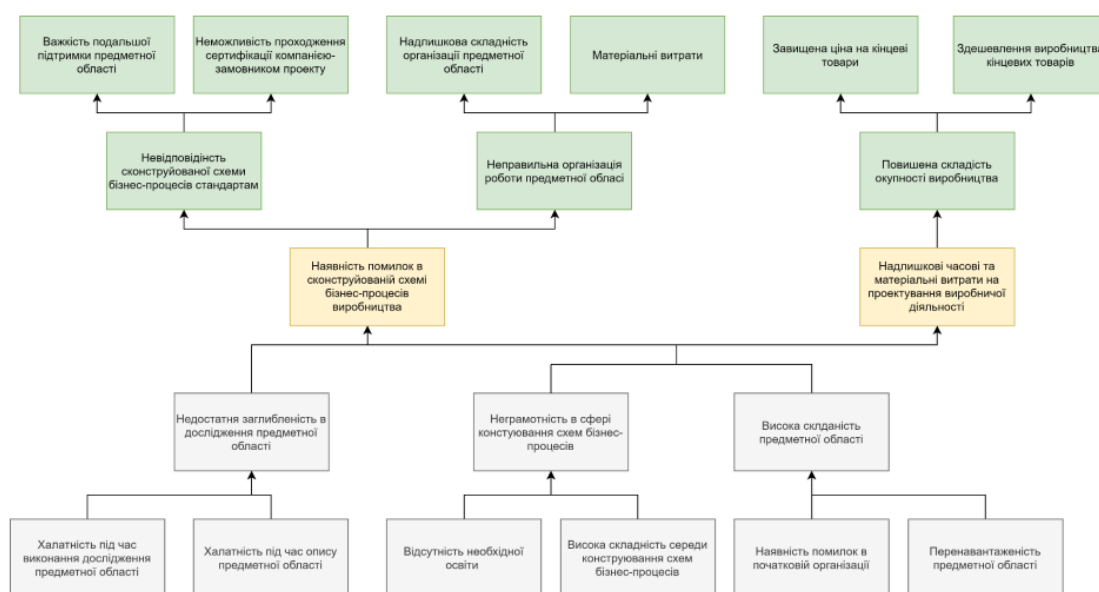


Рисунок 2.1 – Дерево проблем

Виходячи з цього, спосіб прогнозування повинен забезпечувати необхідність виконання обчислень без блокування основного потоку виконання програмного забезпечення.

Також для покращення ефективності роботи бізнес-процесів проведено порівняльний аналіз зацікавлених сторін, зазначеним на рис. 2.2:

Зацікавлена сторона	Зацікавленість в проєкті	Умови зацікавленості в довгостроковості проєкту	Можлива участь
Проектний менеджер	Проект допомагає виправляти допущені помилки та прогнозує майбутні дії за для економії часу, необхідного на проєктування виробничої діяльності	Проект допомагає знайти до 95% помилок в схемі бізнес-процесів та прогнозує користувацькі дії з точністю не менше 90%	Цільова група
Інвестор	Проект допомагає економити час і гроші, необхідні на проєктування виробничої діяльності	Проект допомагає економити час і гроші, необхідні на проєктування виробничої діяльності	Надання інвестицій на розробку проєкту
Виконавець спроектованого проєкту	Упевнений, що помилок в спроектованій системі не допущено, а значить сильніше зацікавлений в його реалізації	Проект допомагає економити час і гроші, необхідні на реалізації спроектованої системи	-
Конкуренти	Зацікавлений в тому, щоб проєкт закритися, тому що проєкт має більшу кількість корисної функціональності, ніж його власний	-	Створення перешкод на шляху реалізації проєкту

Рисунок 2.2 – Порівняльний аналіз зацікавлених сторін

Досягнення вищезазначеною вимоги можливе двома способами:

- використання переривань обчислень при прогнозування користувацьких дій;
- використання додаткового браузерного потоку обчислень.

Проаналізувавши зазначені способи, можна зробити висновок, що використання переривань обчислень в програмному застосунку є більше технічним «трюком», ніж технологією паралельного виконання програмного забезпечення, адже його реалізація зводиться к асинхронному запуску окремих складових частин алгоритму та виконання під час утвореної асинхронної паузи дій, що накопилися з графічної частини програмного застосунку. Таке рішення повністю не вирішує проблеми блокування користувацького інтерфейсу, саме тому було обрано використання додаткового браузерного потоку виконання за допомогою технології вебворкерів.

Суть даного методу полягає в передачі управління над виконанням обчислень додатковому браузерному потоку та асинхронне очікування результатів

обчислень в основному потоці. Таким чином, основний потік веб-додатку не блокується та може обробляти дії користувача і в час, коли вебворкер завершить роботу над обчисленням, отримати результати та відобразити їх користувачеві.

Схема взаємодії основного потоку із додатковим зазначена на рис. 2.3:



Рисунок 2.3 – Схема взаємодії основного та додаткового потоків виконання

2.3 Типи прогнозування користувачьких дій при конструюванні схем бізнес-процесів виробництва

Процес прогнозування користувачьких дій при конструюванні схем бізнес-процесів виробництва можливо розбити на 3 три модулі: аналізуючий, ідентифікуючий та прогнозуючий.

2.3.1 Аналізуючий модуль

Використання аналізуючого модуля дозволяє поетапно аналізувати схеми бізнес процесів виробництва, що конструюється під час роботи з додатком та виявлятиме помилки.

Виявлення помилок за допомогою аналізуючого модуля, їх типи:

- відсутність початкових даних/результатів роботи;
- відсутність виконавців та ресурсів роботи;
- наявність незатунельованих стрілок на схемі;
- наявність незастосованих до взаємодії елементів на схемі;
- відсутність підписів елементів на схемі.

Отже, аналізуючий модуль можна використовувати користувачем як алгоритм дій при використанні.

Опис схеми бізнес процесів виробництва виступатимуть у форматі багаторівневого об'єкту де для алгоритму використовують вхідні дані. На основі цього елементи стають відповідно до свого рівня у схем бізнес процесів, тому робота алгоритму полягатиме в рекурсивному перебої отриманого об'єкту, а саме перевірки на відповідність заданим правилам та повернення результатам перевірки.

Блок-схему аналізуючого алгоритму наведено на рис. 2.4:

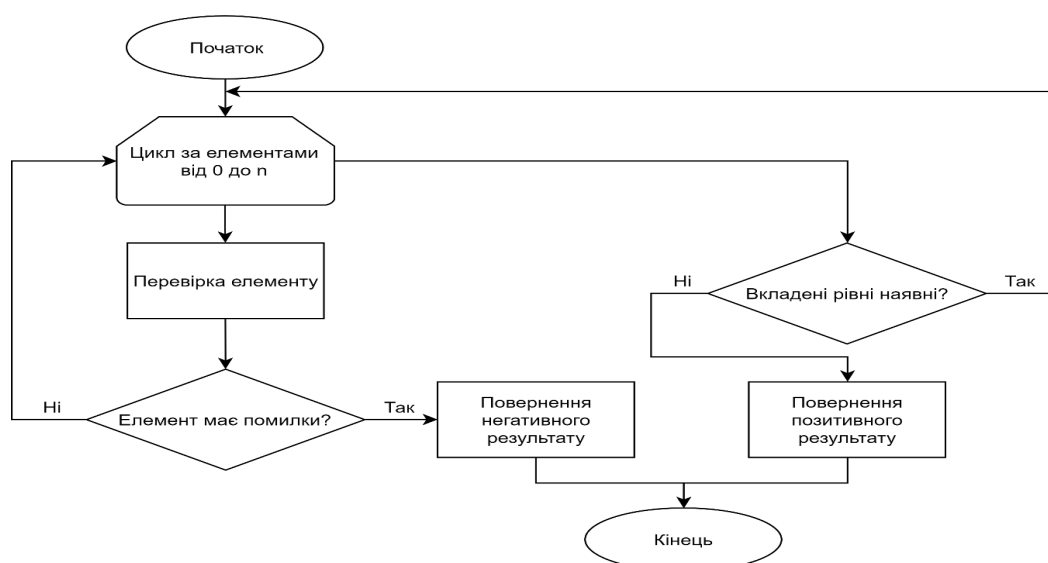


Рисунок 2.4 – Блок схема аналізуючого алгоритму

Даний результат показує чи відбулася зміна стану, і чи можна його надалі застосовувати в прогнозуючому модулі веб додатку.

2.3.2. Ідентифікуючий модуль

Ідентифікуючий модуль однозначно трансформує виконану користувачку дію у структуру даних, що описує всю інформацію про отриману дію у форматі, зрозумілому для подальшої обробки прогнозуючим модулем.

Алгоритм ідентифікації полягає у послідовній побудові об'єкта, що описує користувачку дію, основуючись на даних, отриманих від графічної

частини програмного забезпечення. У випадку, коли дія, що надійшла, відома алгоритму, їй присвоюється ідентифікатор та починається циклічне зчитування додаткової інформації. Отриманий в результаті об'єкт передається до прогнозуючого модуля. Якщо дія, що надійшла до алгоритму, є невідомою, то система повідомляє про це помилкою в лог подій та нічого не повертає, чим перериває подальшу обробку дії.

Додатково слід зазначити, що інформація стосовно всіх можливих користувачьких дій в програмному застосунку передається ідентифікуючому модулю під час завантаження веб-додатку.

Перетворення опису користувачької дії с наявними додатковими атрибутами в корисну функціональність відбувається таки же чином, лише в протилежному напрямку.

Детальну блок-схему алгоритму ідентифікації зазначено на рис. 2.5:

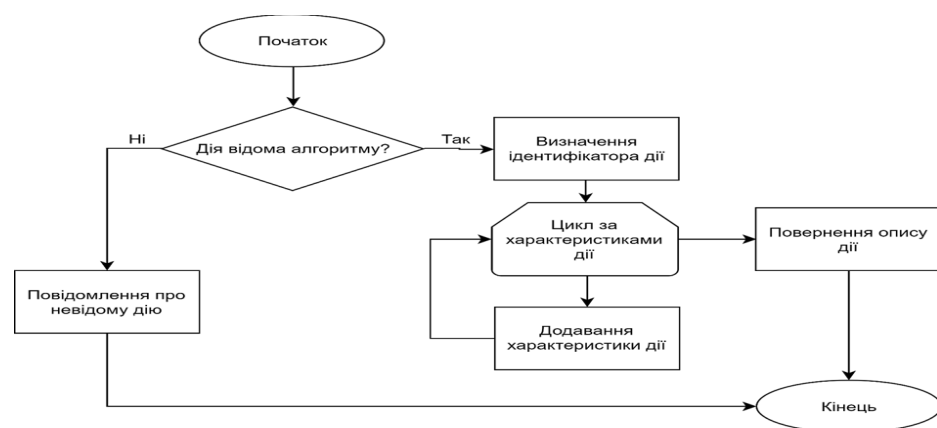


Рисунок 2.5 – Блок-схема ідентифікуючого алгоритму

2.3.3.Прогнозуючий модуль

Загальне призначення прогнозуючого модуля – побудова списку прогнозованих дій, який утворюється на основі декількох останніх виконаних користувацьких дій та дереві, що описує взаємозв'язки між виконаними діями.

Для своєї роботи прогнозуючий модуль використовує дві основні структури, що зберігають дані: ланцюг користувацьких дій та дерево прецедентів.

Ланцюг користувацьких дій – це послідовність описів користувацьких дій, що зберігається в одномірному масиві по принципу черги (перший прийшов, перший пішов). Іншими словами можна сказати, що це обмежена за кількістю дій історія користувацької активності.

Дерево прецедентів – N-арне дерево, що зберігає в собі інформацію про кожну виконану дію, включаючи інформацію про додаткові атрибути, та послідовності виконання користувацьких дій протягом всієї роботи із програмним застосунком, доповнені інформацією про повторення окремих.

Приклади ланцюга та дерева прецедентів зображено на рис. 2.6 – 2.7:

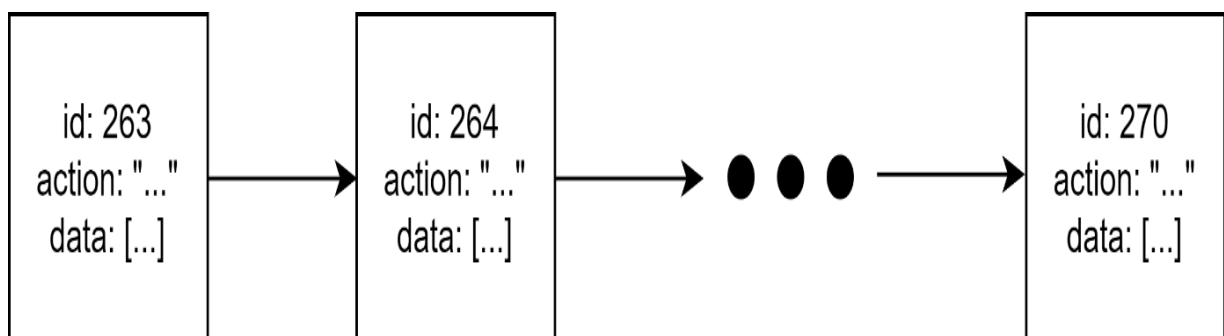


Рисунок. 2.6 – Приклад ланцюга користувацьких дій

Для побудови списку вірогідних для виконання користувачем дій, прогнозуючий модуль виконує комплекс робіт, а саме:

- редагування ланцюга користувацьких дій;
- побудова дерева прецедентів з описів користувацьких дій;
- пошук входжень елементів в дереві прецедентів.

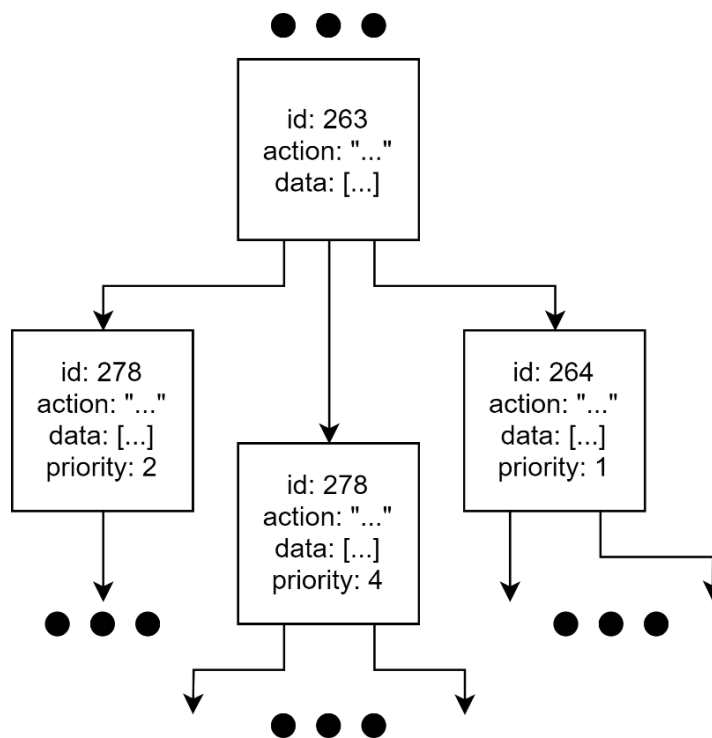


Рисунок 2.7 – Приклад дерева прецедентів

Редагування ланцюга користувацьких дій призначене для оновлення інформації стосовно актуальних дій, що були виконані. Під час редагування в кінець масиву додається елемент, що описує останню користувацьку дію, а також, з початку масиву видаляється перший елемент, як неактуальний для прогнозування.

Побудова дерева прецедентів з описів користувацьких дій виконується після кожної користувацької активності, що призвела до запису інформації в ланцюг користувацьких дій. Дерево прецедентів будується для того, щоб побудувати детальну мапу користувацької активності та використовувати її для прогнозування майбутніх користувацьких дій.

Алгоритм побудови дерева прецедентів базується на пошуку входжень перших $N-1$ елементів з ланцюга користувацьких дій. У випадку, коли входження всього ланцюга знайти не виходить, відсутні елементи ланцюга додаються до дерева прецедентів. Після того, як ланцюг було знайдено, до останнього елемента додається лист, що зберігає в собі інформацію стосовно останньої виконаної користувацької дії, а саме елемент ланцюга на позиції N . Можливий

випадок, коли остання виконана дія вже існує на зазначеному рівні, тоді існуючій дії збільшується пріоритетна одиницю.

Блок-схема зазначеного алгоритму будування дерева прецедентів зображена на рис. 2.8.

Після того, як було завершено побудову дерева прецедентів із додаванням в нього актуальної інформації про останню виконану дію, прогнозуючим модулем виконується процедура пошуку входжень в дереві прецедентів, іншими словами, побудова списку вірогідних до виконання дій.

Для побудови зазначеного списку прогнозуючим модулем запускається пошук входжень ланцюга користувацьких дій в дереві прецедентів. Після того, як входження було знайдене, елементи, що знаходяться на останньому знайденому рівні і утворюють список вірогідних до виконання дій. У випадку, коли входження ланцюга знайдено не було, прогнозуючий модуль повертає порожній список, який символізує недостатню кількість інформації для прогнозування.

Список вірогідних до використання користувачем дій сортується згідно с пріоритетом кожної дії, який визначається кількістю повторень конкретної дії. У випадку, коли пріоритети декількох дій мають однакове значення, утворюється колізія пріоритетів. Для вирішення зазначеної колізії використовується порівняння кожної з дій, що призвели до колізії, згідно з

ланцюгом користувацьких дій. Дія, яка має найбільшу кількість пересічень значень атрибутів із діями з ланцюгу, отримує найбільший пріоритет. Відповідно, дія, яка має найменшу кількість пересічень, отримує найменший пріоритет. У випадку, коли кількість пересічень однакова, діям задається однаковий пріоритет.

Детальну блок-схему алгоритму побудови списку вірогідних до виконання дій наведено на рис. 9.

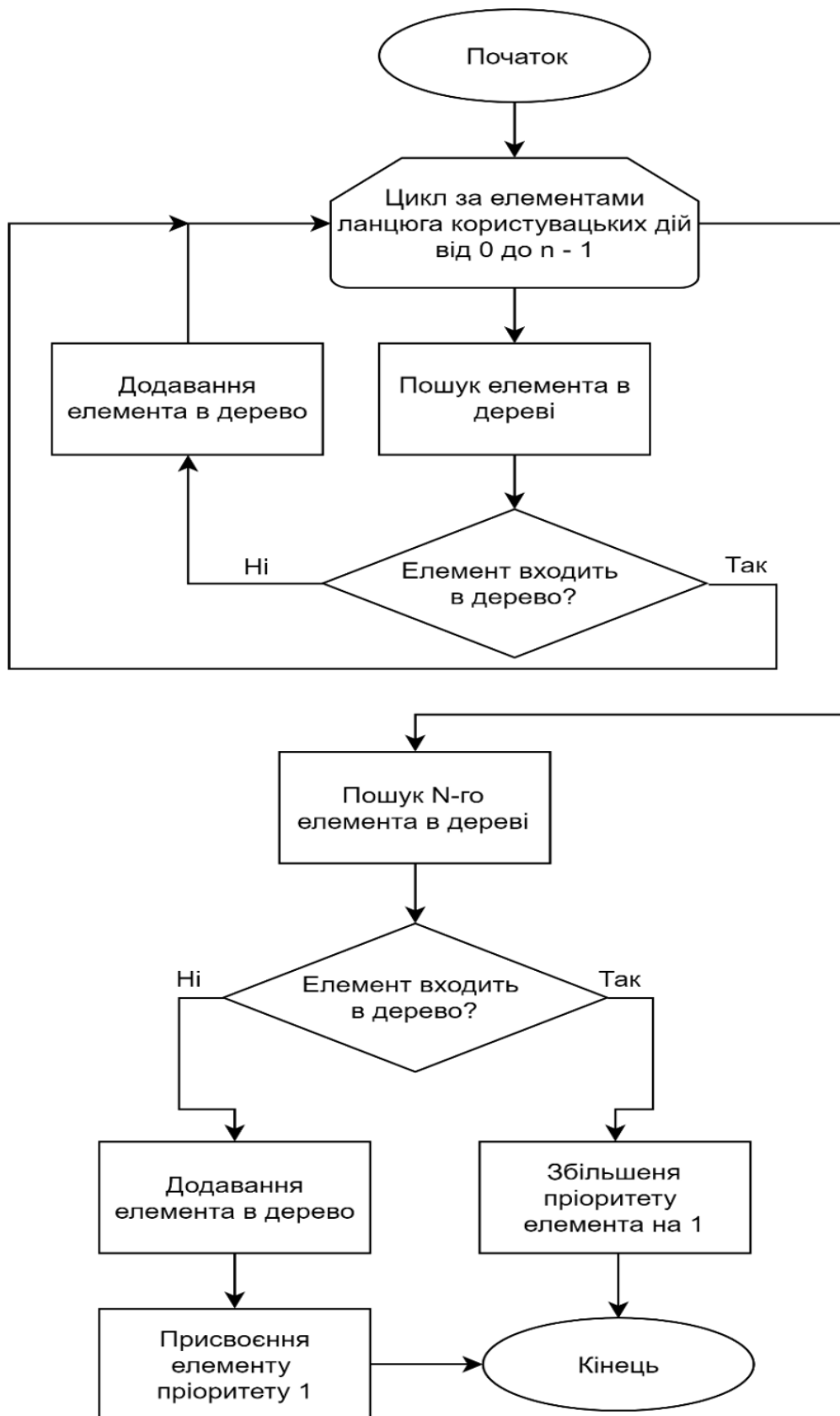


Рис. 2.8 – Блок-схема алгоритму побудови дерева прецедентів

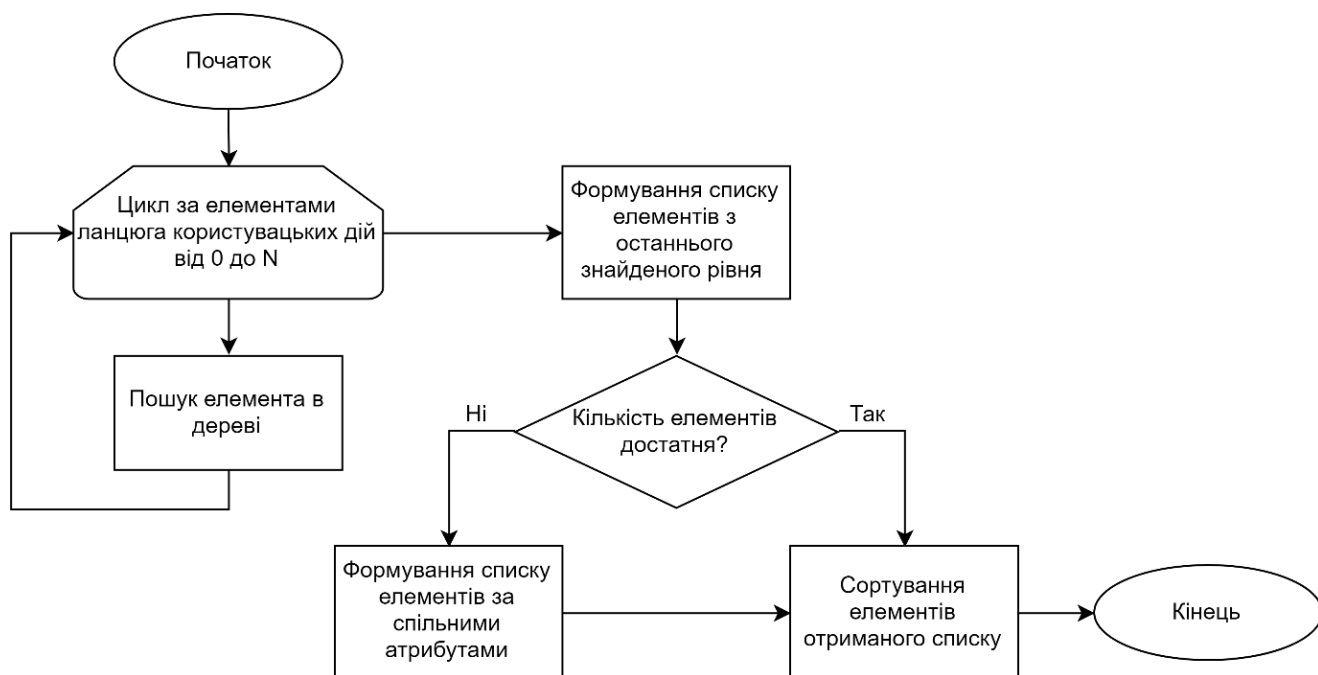


Рис. 2.9 – Блок-схема алгоритму побудови списку вірогідних довиконання дій

Після того, як прогнозуючим модулем було виконано комплекс зазначених дій, в результаті якого побудовано список прогнозованих дій, наведений список повертається до ідентифікаційного модуля для трансформування його на корисну функціональність, після чого відображається користувачеві для виконання.

2.4 Висновки до розділу 2

Беручи до уваги користувацькі вимоги, що були висунуті до способу та програмного забезпечення прогнозування користувацьких дій при

конструюванні схем бізнес-процесів виробництва, можна зазначити що спосіб прогнозування є унікальним та повністю дозволяє реалізувати всі поставлені до нього вимоги, а саме: використання власної активності користувача для побудови майбутніх прогнозів та висока точність прогнозів.

Загальне призначення прогнозуючого модуля – побудова списку прогнозованих дій, який утворюється на основі декількох останніх виконаних користувацьких діях та дереві, що описує взаємозв'язки між виконаними діями.

Для своєї роботи прогнозуючий модуль використовує дві основні структури, що зберігають дані: ланцюг користувацьких дій та дерево прецедентів.

Дерево прецедентів – N-арне дерево, що зберігає в собі інформацію про кожну виконану дію, включаючи інформацію про додаткові атрибути, та послідовності виконання користувацьких дій протягом всієї роботи із програмним застосунком, доповнені інформацією про повторення окремих.

Прогнозуючий модуль після виконання будь-якої дії моделювання схеми бізнес-процесів повинен повертати користувачеві прогнозований список подальших дій. Прогнозування вірогідних дій повинно вираховуватися на основі дерева прецедентів, що будується під час всього часу роботи із програмним забезпеченням на основі описів виконаних користувацьких дій, та послідовності декількох останніх дій.

Стосовно програмного забезпечення можна зазначити, що його архітектура відповідає поставленій задачі, через що програмне забезпечення

відповідає вимогам швидкодії та економічності у використанні машинних ресурсів.

3 ДОСЛІДЖЕННЯ АЛГОРИТМІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРОГНОЗУВАННЯ КОРИСТУВАЦЬКИХ ДІЙ

3.1 Розкриття змісту «Архітектура»

Поняття «архітектура» в контексті даного розділу використовується як загальне і може бути поділено на наступні:

- архітектура взаємодії складових частин;
- архітектура програмного забезпечення.

Основні аспекти, які можуть бути враховані в архітектурному плані дослідження алгоритмів для прогнозування користувацьких дій:

- модульність: Розгляд алгоритмів як окремих модулів з можливістю легкого розширення чи заміни. Це дозволяє покращувати та вдосконалювати алгоритми без зміни всієї системи;

- інтеграція: Вивчення того, як алгоритми взаємодіють між собою та інтегруються в загальну структуру системи. Це може включати аналіз обміну даними та інформацією між різними компонентами;

- ефективність та продуктивність: Розгляд архітектурних рішень, спрямованих на забезпечення оптимальної продуктивності та ефективності в процесі прогнозування користувацьких дій. Це може включати оптимізацію алгоритмів та використання ресурсів;

- розширюваність: Врахування можливості легкого розширення системи для включення нових алгоритмів або підтримки нових функціональностей у майбутньому;

- обробка даних: Розгляд того, як дані про користувача обробляються в системі, включаючи збір, очистку, аналіз та використання для прогнозування.

- масштабованість: Здатність системи ефективно працювати при збільшенні обсягу даних чи користувачів.

3.1.1 Алгоритм дії даної структури:

- збір та підготовка даних:
 - збір необхідних даних про користувачів та їх дії;
 - очищення та підготовка даних для подальшого використання;
- вибір характеристик (features):
 - визначення характеристик, які будуть використовуватися для прогнозування;
 - відбір та інженерія ознак для покращення якості прогнозу;
- розбиття даних – розбиття доступних даних на тренувальний та тестовий набори для оцінки ефективності алгоритму.
- вибір та навчання моделі:
 - вибір конкретного алгоритму машинного навчання чи іншого методу прогнозування;
 - навчання моделі на тренувальних даних;
- оцінка та тестування:
 - оцінка ефективності моделі за допомогою тестового набору даних;
 - виправлення параметрів чи вибір іншого методу, якщо потрібно;
- прогнозування користувацьких дій – використання навченої моделі для прогнозування майбутніх користувацьких дій.
- оцінка результатів:
 - оцінка точності та надійності прогнозування;
 - аналіз результатів та визначення можливостей покращення;
- впровадження та моніторинг:
 - впровадження алгоритму в реальні умови;
 - моніторинг ефективності та оновлення моделі за необхідності.

3.2 Архітектура взаємодії складових частин

Програмний застосунок, що розроблюється, поділено на дві основні частини: графічна та прогноуюча. Зазначені частини є повністю незалежними одна від

іншої, тобто, у випадку, коли необхідно буде, наприклад, деактивувати одну з них, інша зможе працювати в штатному режимі.

Загалом, модульна архітектура складових частин програмного застосунку прогнозування користувацьких дій при моделюванні схем бізнес-процесів була обрана через її мобільність, легкість в тестуванні та розширенні. Ключовим фактором вибору даного типу архітектури стала саме легкість в розширенні, адже з кожним кроком виконання дослідження з'ясовуються нові обставини, що можуть позитивно впливати на роботу прогнозуючого модуля [15]. При кожному виявленні такого фактору необхідно розширювати логіку алгоритму, що, за допомогою модульної архітектури частин програмного забезпечення, робиться доволі швидко та з малою кількістю помилок. Звичайно, легкість в тестуванні також відіграє велику роль в виборі даного типу архітектури, адже протестувавши працездатність один раз можна бути впевненим, що в подальшій роботі не

буде виключних ситуацій, які призведуть до відказу.

Поєднання графічної та прогнозуючої частин програмного забезпечення між собою виконано за допомогою програмного інтерфейсу, що описує просту структуру обміну даних за допомогою публічних методів кожної з частин.

Зображення архітектури взаємодії складових частин програмного забезпечення наведено на рис. 3.1:

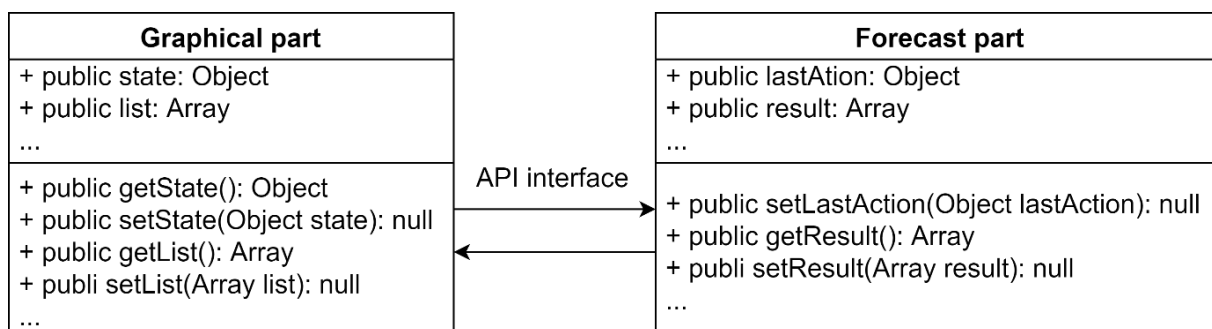


Рисунок 3.1 – Архітектура складових частин програмного забезпечення

3.3 Складові архітектури програмного забезпечення

Розроблюваний програмний застосунок розділено на дві основні складові: графічну та прогнозуючу. Ці складові є повністю автономними, що означає можливість незалежної роботи однієї частини в разі вимкнення або деактивації іншої.

Загалом, модульна архітектура обраних компонентів програмного застосунку для прогнозування користувацьких дій при моделюванні бізнес-процесів була вибрана через її мобільність, легкість у тестуванні та можливість розширення. Ключовим критерієм вибору була легкість у розширенні, оскільки з кожним новим відкриттям досліджень виникає необхідність адаптації алгоритмів, і модульна архітектура дозволяє це робити ефективно та з мінімальною кількістю помилок.

Поєднання графічної та прогнозуючої складових програмного забезпечення здійснено за допомогою програмного інтерфейсу, який визначає просту структуру обміну даними через публічні методи кожної частини.

Розроблений програмний продукт розподіляється на дві основні складові: графічну та прогнозуючу. Обидві частини функціонують автономно, що дозволяє одній частині працювати незалежно в разі відключення або вимкнення іншої.

Загалом, програмне забезпечення, що розроблюється, є клієнт- серверним веб-додатком та складається з двох частин: клієнтської та серверної.

Ключовою особливістю архітектури даного додатку в рамках дослідження, що проводиться, є реалізація всієї необхідної функціональності на клієнтській стороні та відсутність обчислень на серверній стороні за непотрібністю.

У рамках проведеного дослідження основною особливістю архітектури даного додатку є виконання всієї необхідної функціональності на клієнтській стороні та відсутність обчислень на серверній стороні за непотрібністю. Вибір такого типу архітектури є нестандартним для веб-додатків, проте його обґрунтується декількома факторами:

- великий обсяг інформації на клієнтській частині: Збирання значної кількості даних на клієнтській стороні може призвести до великих затрат часу на

комунікацію з сервером;

- складність багатопоточної обробки дерева прецедентів: Організація багатопотокової обробки складних структур, таких як дерева прецедентів, може стати проблематичною на серверній стороні;

- надмірна складність зберігання дерева прецедентів в базі даних: Оптимізоване зберігання складних структур в базі даних може вимагати значних зусиль і ресурсів.

Рішення стосовно вибору такого типу архітектури є нестандартним для веб-додатків, але на користь даного рішення вплинули наступні фактори:

- великий розмір інформації, що збирається на клієнтській частині, впливає на великі затрати часу на спілкування з серверною частиною;

- складність організації багатопоточної обробки дерева прецедентів;

- надлишкова складність організації оптимізованого зберігання дерева прецедентів в базі даних.

Отже, розташування прогнозуючого модуля на серверній частині веб додатку призвело б до надлишкових затрат часу на прогнозування користувацьких дій та організацію самої серверної частини. Набагато простіше є розташування прогнозуючого модуля та організація зберігання користувацьких даних на клієнтській частині веб додатку та надання серверній частинні ролі файлового серверу, що зберігає статистичні частини веб-додатку.

Описана архітектура зображена на рис. 3.2:

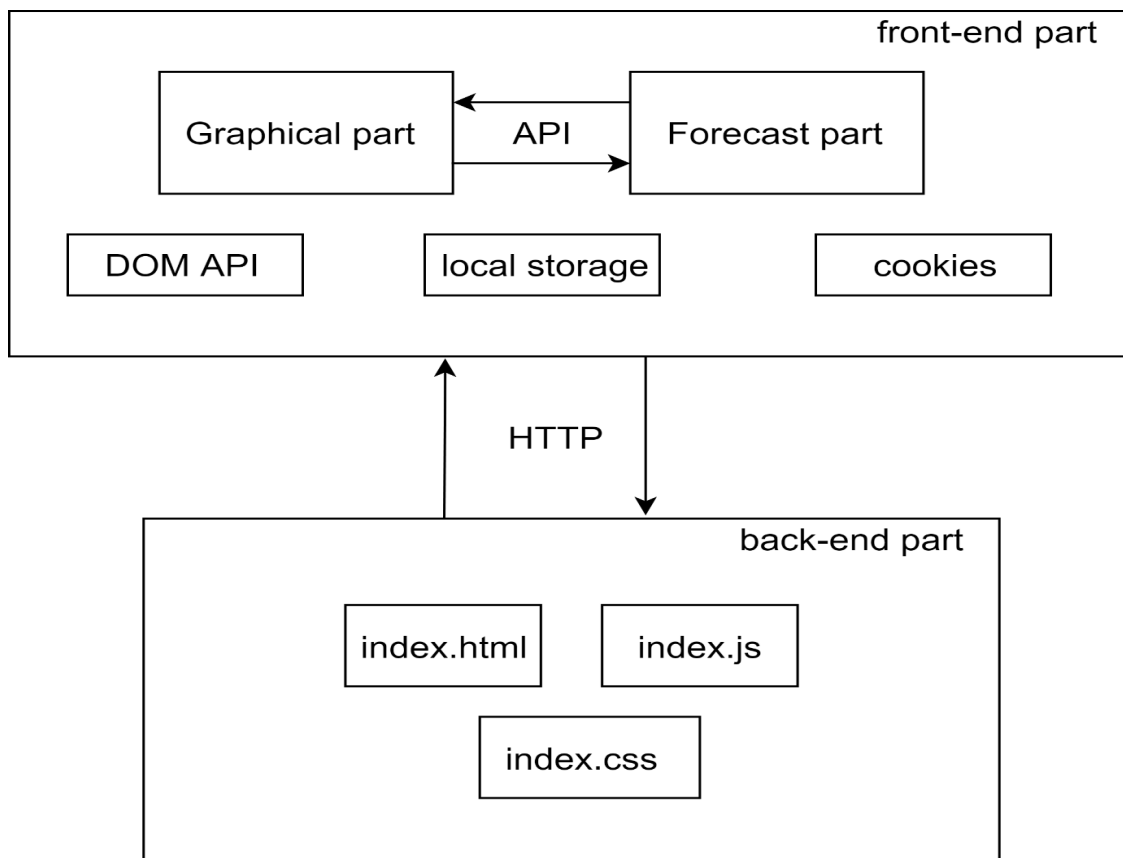


Рисунок 3.2 – Загальна архітектура веб-додатку

3.4 Характеристика інтерфейсу

Розроблений програмний додаток та графічний інтерфейс до нього буде доповнено графічним модулем, що буде демонструвати користувачеві доступні до виконання дії, які є результатом роботи прогнозуючого модуля на кожному кроці конструювання схеми бізнес-процесів. Елементи зазначеного модуля будуть доступні до виконання, про що користувачу буде повідомлено за допомогою стильових особливостей, які змінюються при наведенні курсором миші на елементи. Також, присутню панель управління елементами, буде доповнено кнопкою повернення до попереднього стану схеми бізнес-процесів, реалізація якої буде базуватися на збереженні історії виконаних користувачьких дій та відповідних станів схеми бізнес-процесів, що конструюється.

Структуру блоку, що зберігає запропоновані дії для виконання користувачем, зазначено на рис. 3.3:


- Додати блок "Робота" до схеми
- Додати вхідні дані для створеного блоку "Робота"
- Затунелювати результат блоку "Робота" з наступним рівнем схеми 

Рисунок 3.3 – Структура блоку запропонованих для виконання дії

3.5 Опис програмного застосунку

Оскільки розроблюваний додаток функціонує у веб-середовищі, необхідно розділити його функціональність між клієнтською та серверною частинами. Серверна частина побудована на базі сервера Nginx, призначеного головним чином для надання статичних файлів програмного забезпечення, які виконуються у веб-браузері клієнта.

Клієнтська частина буде реалізована за допомогою мови програмування JavaScript версії ES6, і для створення інтерфейсу користувача буде використана бібліотека ReactJS. Також для збірки та мініфікації вихідних файлів використовуватиметься збірник Webpack.

Застосування збірника Webpack також включає додаткові сценарії використання, такі як:

– використання модуля Webpack-Dev-Server як локального сервера під час розробки програмного забезпечення.

– трансляція сучасної версії мови програмування JavaScript ES6 у більш стару ES5 для розширення підтримки програмного забезпечення різними браузерами.

Для повного функціонування розроблюваного програмного продукту також використовуватимуться додаткові технології, що надаються веб-браузером

користувача, такі як LocalStorage API для локального сховища даних та WebWorker API для інтерфейсу використання додаткових потоків виконання програмних процедур.

Основними вимогами до веб-браузерів, які використовують програмне забезпечення, є наступні:

- підтримка JavaScript ES5 та вище;
- можливість виконання коду на мові програмування JavaScript версії ES5 і більше;
- наявність локального сховища (LocalStorage API);
- забезпечення локального сховища для зберігання користувацьких даних, використовуючи LocalStorage API;
- підтримка HTML5;
- відповідність сучасному стандарту HTML5, включаючи функціональність WebWorker API.

Оскільки аналіз схеми бізнес-процесів для виявлення помилок виконується після кожної реєстрації користувацької дії, критично важливо, щоб цей процес займав мінімальний час. Такий аналіз входить у процес прогнозування користувацьких дій, і весь час, витрачений прогнозуючим модулем, користувач перебуває в очікуванні результатів обробки.

Для розв'язання цієї проблеми я використовував складний вкладений об'єкт, що містить опис елементів, які входять до конкретного рівня побудованої схеми бізнес-процесів. Цей об'єкт також включає в себе описи всіх дочірніх рівнів разом із їхніми елементами.

Такий формат структури даних дозволяє перевіряти кожен елемент на конкретному рівні схеми бізнес-процесів на наявність помилок. Завдяки зберігання певних властивостей елемента в глобальному середовищі щодо заданого елемента, можна перевірити наявність помилок, залежних від конкретного елемента.

Використовуючи рекурсивний алгоритм для обходження цього об'єкта, я отримав можливість швидко виконувати перевірку всієї схеми бізнес-процесів на наявність помилок. У випадку виявлення помилки, система може вказати конкретний рівень вкладеності та елемент, що належить цьому рівню, де виникла помилка.

Основна задача ідентифікуючого модуля – це трансформування початкової інформації про виконану користувачем дію в опис виконаної дії, що має певний формат, зрозумілий прогнозуючому модулю, за допомогою атрибутів, які мають структуру «ключ – значення». Виконання такого трансформування необхідне для приведення інформації про будь-яку зареєстровану дію до єдиного вигляду, так як цього вимагає модульна архітектура прогнозуючого модуля.

Для повноцінної реалізації функціональності ідентифікації користувацьких дій, модуль отримує список усіх можливих дій, які можуть бути зареєстровані у програмному застосунку, при завантаженні застосунку. З цього списку генерується набір методів публічного інтерфейсу, які подальше передаються у глобальне середовище програмного застосунку. Після того, як користувач виконав будь-яку зареєстровану дію, викликається відповідний метод, наданий ідентифікаційним модулем, і йому передаються параметри, які описують виконану дію. Серед цих параметрів можуть бути:

- інформація про браузерну подію, що спричинила виклик методу інтерфейсу модуля.
- часовий проміжок, коли була виконана дія.
- дані про стан схеми бізнес-процесів на конкретному рівні.
- додаткові атрибути.

Важливо відзначити, що у деяких випадках необхідні атрибути можуть бути визначені в початковому списку, за яким генеруються методи публічного інтерфейсу модуля. Цей підхід сприяє підвищенню надійності ідентифікаційного модуля та точності прогнозування, оскільки всі дані, які очікує прогнозуючий модуль, будуть надані йому.

Після отримання інформації про користувацьку дію ідентифікаційний модуль проводить її перевірку на відповідність шаблону. У випадку успішної перевірки інформації, ідентифікаційний модуль обирає необхідні дані зі стартового складного об'єкта, формує новий об'єкт та присвоює йому унікальний ідентифікатор виконаної дії. Результат роботи ідентифікаційного модуля передається та обробляється прогнозуючим модулем.

Структура дерева, використовуваного для вирішення задачі прогнозування, буде наступною. Кореневим елементом дерева буде спеціальний об'єкт, що вказує на місце початку пошуку в дереві. Глибина вкладеності дерева, тобто кількість рівнів елементів, буде рівна довжині ланцюгу користувацьких задач. На рівнях від першого до передостаннього будуть зберігатися елементи, що відіграють навігаційну роль у процесі побудови списку можливих до виконання дій. Ці елементи служитимуть для пошуку останнього рівня дерева прецедентів, на якому будуть зосереджені описи користувацьких дій, можливих до виконання на останньому кроці користувацької активності.

Для досягнення максимальної продуктивності при додаванні нового елемента або пошуку існуючого елемента слід використовувати пошук по дереву в ширину. Це дозволяє прискорити виконання операцій за рахунок ігнорування сторонніх гілок дерева, які не входять в актуальний маршрут користувацьких дій.

Ще однією особливістю дерева прецедентів є ефективне використання пам'яті у випадках, коли ланцюг актуальних користувацьких дій вказує на вже існуючу дію, збережену в дереві прецедентів. Замість додавання нового елемента на конкретний рівень, існуючому елементу збільшується пріоритет на одиницю, а також додаються додаткові значення атрибутів, що описують конкретну користувацьку дію. В результаті модифікований елемент отримує більший пріоритет відносно інших елементів даного рівня і в подальших прогнозах займатиме вищу позицію в побудованому списку вірогідних до виконання користувацьких дій.

У цій роботі графічним інтерфейсом є компонент, який забезпечує можливість відображення та взаємодії зі списком прогнозованих дій. Для

реалізації цього модуля необхідно створити графічний компонент за допомогою бібліотеки ReactJS, який, при виборі користувачем необхідної прогнозованої дії, буде викликати публічний метод прогнозуючого модуля.

Загальна структура компонента включає наступні частини:

- обгортка: ця частина компонента служить для загального відображення та взаємодії із списком прогнозованих дій;
- список подій: ця частина отримує функціональність та розподіляє її між елементами списку. Вона відповідає за відображення та оновлення списку доступних прогнозованих дій;
- елемент списку подій: ця частина отримує та викликає передану функціональність для окремих елементів списку. Вона взаємодіє з користувачем та ініціює виклики методів прогнозуючого модуля при виборі конкретної прогнозованої дії.

За допомогою методів життєвого циклу, компонент підписаний на зміні загального стану веб-додатку, що дозволяє йому своєчасно оновлювати актуальні дані про список прогнозованих дій під час кожної зміни.

Загальні висновки:

- для розробки програмного забезпечення, що прогнозує користувацькі дії під час конструювання схем бізнес-процесів виробництва, використана клієнт-серверна архітектура для веб-додатку. Ця архітектура містить графічну та прогнозуючу частини, які взаємодіють через програмний інтерфейс взаємодії компонентів;
- вибір такої архітектури зумовлений кількома факторами, включаючи те, що програмне забезпечення працює на платформі веб та враховує існуючий веб-додаток. Модульність архітектури дозволяє робити різні частини незалежними та легко розширювати їхню функціональність;
- програмне забезпечення використовує публічні методи класів для взаємодії між графічною та прогнозуючою частинами, що дозволяє їм спільно працювати та обмінюватися інформацією;

– основним завданням було досягнення оптимальної продуктивності прогнозуючого модуля при ефективному використанні пам'яті. Використання вказаних особливостей архітектури було обрано для досягнення цієї мети;

– практичне тестування підтвердило позитивний вплив програмного забезпечення на продуктивність веб-додатку, що підтримує конструювання схем бізнес-процесів.

Ці висновки свідчать про успішність вибраних архітектурних та технологічних рішень у реалізації функціональності та вдосконаленні ефективності програмного забезпечення.

3.6 Висновки до розділу 3.

Для реалізації програмного забезпечення прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва буде використано клієнт-серверну архітектуру для додатку на платформі веб, що складається з двох основних частин, графічної та прогнозуючої, побудованих за модульною архітектурою та поєднаних між собою за допомогою програмного інтерфейсу взаємодії компонент через публічні методи класів.

Вибір даного архітектурного рішення для заданого програмного забезпечення зумовлено наступними факторами:

– реалізація програмного забезпечення прогнозування користувацьких дій відбувається на основі існуючого програмного застосування на платформі веб;

– окремі частини програмного забезпечення прогнозування користувацьких дій повинні бути незалежними одна від іншої; вибрана архітектура повинна надавати можливість легкого розширення функціональності програмного забезпечення, що розроблюється.

Оскільки досягнення оптимальних показників продуктивності прогнозуючого модуля з малими витратами машинної пам'яті було важливою задачею, що ставилась до розроблення програмного забезпечення, то використання описаних особливостей було необхідним рішенням.

Також слід зазначити, що перевіряючи працездатність програмного застосунку на практиці, спостерігався позитивний вплив на швидкодію веб-додатку.

4 ОЦІНКИ ОТРИМАНИХ РЕЗУЛЬТАТІВ

Тестування розробленого програмного забезпечення проводилося на різних етапах його розробки і включало декілька етапів. По-перше, було проведене тестування функціональності прогнозування користувацьких дій методом білого ящика. По-друге, тестування користувацького інтерфейсу програмного забезпечення здійснювалося методом чорного ящика.

Аналізуючи основні методи тестування для програмного забезпечення з інтерфейсом користувача, було вирішено провести тестування графічної та прогнозуючої частин програмного забезпечення окремо, оскільки для кожної з них встановлені власні вимоги та стандарти.

Для прогнозуючої частини вибраний метод білого ящика, оскільки цей метод є ефективним для виявлення помилок у програмному кодї, коли тестувальнику відома структура коду. Тестування прогнозуючої частини було успішним, виявлено та виправлено декілька важливих для точності прогнозування помилок.

Тестування графічної частини виконувалося методом чорного ящика, оскільки цей метод дозволяє відокремити тестування від реалізації програмного коду та зосередитися на якості взаємодії програмного забезпечення з кінцевим користувачем. Під час цього тестування виявлено та виправлено одну важливу помилку в користувацькому інтерфейсі.

Загальний висновок з результатів тестування свідчить про відсутність критичних помилок у реалізованому програмному забезпеченні та його здатність до якісного виконання завдань у взаємодії з кінцевим користувачем.

Після успішного завершення розробки програмного забезпечення для прогнозування користувацьких дій був проведений аналіз відповідності програми користувацьким вимогам, описаним у пункті 2.1, а також було оцінено якість програмного коду, графічного інтерфейсу та функціональності прогнозуючого модуля.

Розроблене програмне забезпечення:

– забезпечує графічний користувацький інтерфейс для взаємодії зі

згенерованим списком вірогідних дій;

- реалізовано функціонал для аналізу побудованої схеми бізнес-процесів на наявність помилок та можливості відновлення схеми до попереднього стану;

- в автоматичному режимі формує список можливих до виконання користувацьких дій за часовий проміжок менший ніж 1000 мілісекунд, базуючись на попередніх діях.

Таким чином, можна стверджувати, що всі основні вимоги користувачів були виконані в розробленому програмному забезпеченні.

Аналіз програмного коду та результатів тестування підтверджує відсутність критичних помилок та налагодженість всіх аспектів програми. Графічний інтерфейс є дружелюбним для користувача та включає в себе інтерактивні елементи, які чітко вказують користувачеві на функціональність додатку. Висока швидкість реакції та оновлення даних додають комфорту та ефективності при взаємодії з програмним забезпеченням.

У цьому дослідженні розроблене програмне забезпечення для прогнозування користувацьких дій повністю функціональне, проте після проведення аналізу кінцевого продукту виявлено деякі аспекти, які можна вдосконалити для поліпшення ефективності програми, зменшення використання машинної пам'яті та підвищення точності прогнозування користувацьких дій. Серед пропонованих оптимізацій можна виділити наступні:

4.1 Динамічна довжина ланцюгу користувацьких дій та динамічна максимальна вкладеність дерева прецедентів

Розглядається можливість впровадження динамічної довжини ланцюгу користувацьких дій та, відповідно, динамічної максимальної вкладеності дерева прецедентів. Це дозволить системі бути більш гнучкою в реакції на зміни потреб користувача, адаптуючись до різних сценаріїв.

4.2 Події "повернення до попереднього стану"

Розглядається можливість використання події "повернення до попереднього стану" для зменшення пріоритету обраної дії, після виконання якої відбувається повернення. Це може поліпшити точність та динаміку системи, а також дозволить краще керувати вибором та адаптацією системи до поточних потреб користувача.

4.3 Можливість демонстрації результатів прогнозованої дії

Розглядається можливість додавання функціоналу, який дозволить переглядати результати прогнозованої дії без її фактичного виконання. Це може поліпшити зручність використання та прискорити роботу з програмою, дозволяючи користувачеві швидко оцінити очікувані результати.

В теперішній момент у розробленому програмному забезпеченні, зокрема в прогнозуючому модулі, існує умова, яка встановлює обмеження на довжину ланцюгу користувацьких дій і, відповідно, глибину вкладеності дерева прецедентів. Згідно з аналізом, представленим у пункті 4.2, це обмеження акцентує увагу на кількості елементів у межах 6 одиниць, оскільки в цьому випадку прогнозуючий модуль демонструє найвищу ефективність. Однак цей показник стає значущим лише тоді, коли дерево прецедентів заповнене достатньою кількістю елементів для здійснення прогнозів, які повертають від 3 до 5 вірогідних для виконання дій, що виявляється найбільш зручним для кінцевого користувача.

У випадку недостатнього обсягу даних в дереві прецедентів, зокрема на етапі початкового навчання програмного забезпечення, зазначена довжина ланцюга користувацьких дій може призвести до тривалого часу для вибору зручного для користувача прогнозу, оскільки модуль прогнозування вірогідних дій має виконати важку задачу на основі обмеженої інформації. Важливо відзначити, що це не є недоліком роботи програмного засобу, а скоріше є його характеристикою.

Отже, впровадження динамічної довжини ланцюгу користувацьких дій, яка залежить від кількості вже зібраних та структурованих прогнозуючим модулем

елементів, може значно покращити взаємодію та швидкість повернення користувачу зручного набору вірогідних для виконання дій вже через 20–25 виконаних дій у програмному забезпеченні, що відповідає 2–3 хвилинам роботи.

Аналізуючи специфіку взаємодії з розробленим програмним забезпеченням, можна зробити висновок, що подія повернення до попереднього стану конструйованої схеми бізнес-процесів виробництва має винятковий характер серед інших подій, які користувач може викликати при роботі.

Оскільки результат події відміни може бути розглянутий як відмова від здійснених змін через їхню невідповідність очікуванням, можна зробити висновок, що виконана подія, яку користувач відміняє, була невірною і не відповідала вимогам для редагування схеми бізнес-процесів на конкретному кроці.

Отже, для підвищення точності прогнозуючого модуля можна розглянути обробку події повернення до попереднього стану схеми бізнес-процесів, яка викликалася після виконання дії зі списку, побудованого прогнозуючим модулем.

4.4 Висновок до розділу 4

Згідно з результатами тестування та аналізу кінцевого програмного забезпечення для прогнозування користувацьких дій при створенні схем бізнес-процесів у виробництві, можна стверджувати, що розроблений програмний застосунок не виявив помилок, які можуть викликати виняткові ситуації при взаємодії з користувачем. Крім того, він повністю відповідає вихідним вимогам, які були сформульовані під час опитування потенційних користувачів цього програмного засобу.

Також був складений список можливих покращень для розробленого програмного забезпечення з метою зменшення навантаження на апаратну частину виконавчого пристрою та підвищення точності результатів у процесі прогнозування ймовірних дій користувача.

5 ПОБУДОВА БІЗНЕС-МОДЕЛІ

5.1 Аналіз проблеми

В галузі проектування операційної діяльності в наявній момент часу виявлено дві ключові труднощі:

- виявлення семантичних помилок у кінцевих схемах бізнес-процесів;
- зайві витрати часових ресурсів, необхідних для проектування виробничої діяльності.

Ці дві проблеми є високопріоритетними для користувача, оскільки їх ігнорування призводить до надмірних матеріальних витрат при проектуванні виробничої діяльності.

Після вирішення вказаних проблем користувач отримає можливість виявлення семантичних помилок в схемі бізнес-процесів на етапі її конструювання та зможе оптимізувати витрати часових ресурсів, що витрачаються на проектування, що призведе до ефективнішого використання матеріальних ресурсів.

На даний момент відсутнє єдине рішення для обох вказаних проблем. Деякі програмні продукти мають модулі для остаточної перевірки готових схем бізнес-процесів на нотаційні помилки (наприклад, ErWin, Terrasoft), що представляє собою перше існуюче рішення для першої проблеми.

Однак основний недолік цих продуктів полягає в тому, що вони перевіряють схеми після їх реалізації, тобто на етапі, що слідує за проектуванням системи, що призводить до значних витрат на виправлення помилок. Крім того, вони не можуть передбачити виникнення помилок на етапі конструювання схем бізнес-процесів.

Щодо проблеми з економією часових ресурсів, наразі не існує загальноприйнятих рішень через специфічність та неоднозначність цієї сфери. Модуль, який пропонує вирішення цієї проблеми, тісно пов'язаний із специфікою архітектури конструктора схем бізнес-процесів і буде постачатися комплексно з

цим конструктором. Несприятливість такого підходу може виникнути у зв'язку із необхідністю комплексного переходу на новий інтерфейс та особливості роботи з конструктором. Однак переваги в якісній оптимізації та раціональному використанні матеріальних ресурсів мають переважати невеликі труднощі, пов'язані із засвоєнням нового інструменту.

5.2 Створення бізнес-моделі

Під час створення бізнес-моделі я виокремив наступні зацікавлені сторони, які виявили інтерес до реалізації даного проекту:

- керівник проекту;
- інвестор;
- розробник системи;
- конкуренти.

На мою думку, ефективним рішенням вищезазначеної проблеми є збір та обробка інформації про користувацькі дії, а також аналіз результатів виконання цих дій. Під аналізом результатів я розумію оцінку проміжної схеми бізнес-процесів на предмет виявлення допущених помилок.

Таким чином, володіючи таким комплексом даних, система зможе прогнозувати подальші користувацькі дії та передбачати можливі помилки ще на етапі конструювання схем бізнес-процесів. Важливо відзначити, що для використання програмного забезпечення користувачеві не потрібні додаткові навички або знання, оскільки він буде спостерігати лише за кінцевим результатом роботи, а управління процесом прогнозування в програмі буде повністю автоматизоване.

Магістерська робота призвела до створення програмного забезпечення, яке здатне прогнозувати користувацькі дії при конструюванні схем бізнес-процесів у виробництві. Це програмне забезпечення є логічним доповненням до результатів моєї бакалаврської роботи, а саме конструктора схем бізнес-процесів на платформі веб. Серед основних розширень можна виокремити модуль для

прогнозування користувацьких дій та графічний модуль для відображення результатів прогнозування.

Прогнозуючий модуль ґрунтується на дереві прецедентів, структурі даних, яка дозволяє зберігати та зчитувати опис користувацьких дій, впорядкованих з урахуванням хронології виконання дій в залежності від ініціюючої події.

Виходячи з вищесказаного, можна стверджувати, що прогнозуючий модуль здатний до самонавчання при реєстрації користувацьких дій, а саме адаптується до роботи з конкретним користувачем.

Для повноцінного функціонування наукового продукту необхідно розробити набір правил і вимог, на основі яких можна стверджувати з впевненістю, що проміжна схема бізнес-процесів не містить помилок.

Отже, можна констатувати, що кінцевим бізнес-продуктом є програмне забезпечення, яке спрощує конструювання схем бізнес-процесів та прогнозує користувацькі дії на етапі конструювання. Основними перевагами цього продукту перед аналогами є можливість прогнозування користувацьких дій, надання підтримки користувачу при конструюванні схем бізнес-процесів і виконання цих дій на етапі конструювання схем.

Підсумовуючи вищезазначене, можна визначити основні переваги, які користувач отримає при використанні описуваного бізнес-продукту:

- можливість конструювання схем бізнес-процесів в кросплатформенному середовищі: користувач отримає гнучкість та універсальність у роботі з програмним забезпеченням на різних платформах, що сприяє зручності та доступності процесу конструювання;

- отримання миттєвого аналізу схеми бізнес-процесів, що конструюється: користувач може негайно отримувати аналітичну інформацію щодо виробничих процесів, що допомагає в прийнятті обґрунтованих рішень під час конструювання;

- зменшення кількості помилок, що допускаються, при конструюванні схеми: дане програмне забезпечення сприяє автоматизованому прогнозуванню та виявленню помилок, що допомагає уникнути неправильних рішень під час

проектування;

– економія часових і матеріальних ресурсів: використання цього бізнес-продукту спрощує та прискорює процес конструювання схем, що призводить до ефективнішого використання ресурсів, які призначені на розробку та проектування.

Визначимо та класифікуємо конкурентні переваги зазначеного продукту на дві категорії: природні (що базуються на основних перевагах продукту) та штучні (маніпуляційні).

До категорії природних конкурентних переваг можна віднести наступні:

– наявність прогнозуючого модуля: основна перевага продукту полягає в наявності модуля, що дозволяє прогнозувати користувацькі дії та аналізувати схему бізнес-процесів на наявність помилок, зменшуючи їх кількість;

– незалежність від платформи виконання: продукт може працювати в браузері, що робить його гнучким у використанні та дозволяє користувачеві конструювати схеми бізнес-процесів з будь-якого пристрою, підключеного до мережі Інтернет;

– вартість: мінімальна вартість продукту дозволяє користувачам налагоджувати процес конструювання схем без значних матеріальних витрат.

До категорії штучних конкурентних переваг належать:

– можливість динамічного конструювання схем бізнес-процесів: Продукт надає необмеженість в наповненні конкретного рівня схеми, що дозволяє схемі бути гнучкою при конструюванні;

– можливість використання тимчасової підписки: Користувач може спочатку випробувати функціональність продукту за допомогою тимчасової підписки перед тим, як вирішити про придбання, що дозволяє ефективно оцінити його корисність порівняно з конкурентами.

Для максимізації ефективності роботи команди проекту рекомендовано виконати сегментацію ринку клієнтів, що означає виділення груп користувачів, які мають схожі інтереси щодо пропонованого продукту.

У випадку конструктора схем бізнес-процесів, сегментацію ринку клієнтів можна провести за двома категоріями:

- обсяг запланованих робіт:
 - користувачі, які готові оплатити ліцензію для використання продукту, тим самим отримуючи необмежений доступ до робіт та рівня їх вкладеності;
 - користувачі, які готові переглядати рекламу та обмежувати обсяг робіт та рівень їх вкладеності під час користування продуктом.
- Можливість купити ліцензію на використання продукту.

Відповідно до цих двох критеріїв, потенційних клієнтів можна розділити на 2 групи:

- група 1: користувачі, чия професійна діяльність пов'язана з проектуванням операційної діяльності. Ця категорія включає користувачів, які є основним джерелом доходу для компанії;
- група 2: користувачі, які знайомляться або вивчають проектування операційної діяльності.

Такий підхід дозволяє зорієнтувати маркетингові та рекламні стратегії на відповідних аудиторій та забезпечити більш ефективну взаємодію з різними групами користувачів.

При аналізі витрат на проект важливо виділити дві ключові категорії: стартові витрати, які здійснюються одноразово на початковому етапі, та щомісячні витрати, які виникають регулярно для подальшого розвитку проекту.

Стартові витрати:

- витрати на юридичні послуги;
- витрати на рієлторські послуги, меблі та канцелярські товари;
- витрати на комп'ютери, комплектуючі та програмне забезпечення;
- витрати на транспорт.

Щомісячні витрати:

- зарплати за посадами;
- оренда офісного приміщення;
- витрати на комунікації;
- податки;
- непередбачені витрати.

Ця розрізнена категоризація дозволяє ефективніше керувати фінансовими потоками та точніше планувати витрати для досягнення узгодженого фінансового плану проекту.

Висновки: опис бізнес-моделі продукту відображено у формі основних блоків, які будуть узагальнені у компактну і інформативну графічну схему бізнес-моделі.

Ціннісна пропозиція: пропонований продукт - конструктор схем бізнес-процесів виробництва з можливістю прогнозування користувацьких дій. Призначений для комерційних організацій, зайнятих проектуванням операційної діяльності. Забезпечує економію матеріальних і часових ресурсів при конструюванні схем бізнес-процесів.

Відміни, що ведуть до переваг: основною перевагою продукту є прогнозуючий модуль, який дозволяє передбачати користувацькі дії під час конструювання схем бізнес-процесів. Це допомагає економити час і матеріальні витрати на конструювання схем.

Сегменти споживачів: споживачами продукту є фахівці, чиї професійні діяльності пов'язані з проектуванням операційної діяльності. Додатково, продукт може бути корисним для тих, хто вивчає або знайомиться з цією галуззю.

Канали продажів: основний канал продажів - продаж ліцензій кінцевому користувачеві. Додатковий канал - продаж місць для контекстної реклами.

Структура доходів компанії: основний дохід компанії - оплата за використання продукту через придбання ліцензій. Плата здійснюється періодично.

Ключові партнери: необхідні матеріальні інвестиції для початку роботи та партнерство з послугами, що надають серверні ресурси для розміщення продукту.

Ключові ресурси: фінансові ресурси для найму персоналу та розробки продукту, обладнання для розробки та оренда офісного приміщення.

Структура витрат: основні витрати включають заробітну плату, утримання комунікацій і серверних служб, а також витрати на оренду офісу.

Ключові вимірювачі: рентабельність продажів і розвиток корисної функціональності продукту є ключовими факторами для вимірювання успіху проекту.

ВИСНОВКИ

Виходячи з загального концепту прецедентного опису об'єктів, завдання прогнозування користувацьких дій при конструюванні схем бізнес-процесів зводиться до динамічного формування моделі поведінки конкретного користувача та подальшого аналізу цієї створеної моделі. Прецедентний опис ознак розглядається як потужна технологія порівняно з іншими методами аналізу, використовуваними раніше, що дозволяє значно спростити аналіз інформації в завданнях високих розмірностей.

Враховуючи користувацькі вимоги, які були висунуті до методу та програмного забезпечення прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва, можна зауважити, що цей метод є унікальним і повністю відповідає всім поставленим вимогам, а саме: використання власної активності користувача для прогнозування майбутніх подій та висока точність прогнозів.

Для реалізації програмного забезпечення прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва буде використана клієнт-серверна архітектура для веб-додатку. Цей додаток складається з двох основних частин - графічної та прогнозуючої, обидві побудовані за модульною архітектурою та пов'язані між собою за допомогою програмного інтерфейсу взаємодії компонентів через публічні методи класів.

На основі результатів тестування та аналізу кінцевого програмного забезпечення прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва можна заявити, що розроблений програмний застосунок не містить помилок, які можуть призвести до виняткових ситуацій під час роботи користувача, і повністю відповідає початковим вимогам, що були висунуті під час опитування потенційних користувачів програмного застосунку.

В результаті проведеної роботи було сформовано та побудовано бізнес-модель кінцевого продукту, яка описує ключові аспекти в організації діяльності, пов'язаної з впровадженням розробленого програмного забезпечення

прогнозування користувацьких дій при конструюванні схем бізнес-процесів виробництва в комерційному середовищі організації проектної діяльності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) Розпізнавання. Математичні методи. програмна система. Практичні застосування. [Електронний ресурс] - Журавльов Ю. І., Рязанов В. В., Сенько О. Ст – М.: Фазіс, - 2006;
- 2) Прикладний методи аналізу даних та знань. [Електронний ресурс] – Загоруйко Н.Г. -Новосибірськ: ІМ СО РАН, - 1999;
- 3) Статистичне моделювання та прогнозування. [Електронний ресурс] - Єріна А.М. - К.: КНЕУ, - 2001;
- 4) Моделювання процесів виділення ознак із використанням латентно-семантичного аналізу [Електронний ресурс] - Соловйов А.М. - СпбГУ, - 2008;
- 5) Feature engineering for machine learning. [Електронний ресурс] - A.Zheng, A.Casazi, - O'Reilly, – 2018;
- 6) Прикладна статистика. [Електронний ресурс] - Орлов А.І. – Видавництво «Іспит», - 2004;
- 7) Введення у аналіз даних. Підручник для бакалаврату та магістратури [Електронний ресурс] - Міркін Б.Г. - М: Фаза, - 2016;
- 8) Управління виробничим підприємством [Електронний ресурс] //ІС. - 2016.
- 9) Guide to the Software Engineering Body of Knowledge (SWEBOOK), IEEE 2004 Version [Електронний ресурс] - Посібник до Зводу Знань з Програмної Інженерії, надалі просто "SWEBOOK" [SWEBOOK, 2004];
- 10) Єліферов В.Г., Рєпін В.В. Бізнес процеси. Регламентація та управління [Електронний ресурс] - видавництво Інфра-М, Київ, 2005.
- 11) Аналіз вимог за Вігерсом [Електронний ресурс] // Business Analysis. - 2004.70
- 12) Башмаков А.І., Башмаков І.А. Інтелектуальні інформаційні технології [Електронний ресурс] // -М.: Вид-во МДТУ ім. н.е. Баумана, 2005.
- 13) Прикладний методи аналізу даних та знань. [Електронний ресурс] - Загоруйко Н.Г. - Новосибирськ: ІМ СО РАН, - 2005;
- 14) Методики збору вимог до програмного забезпечення [Електронний

ресурс] // CASEClub. – 2003

15) Тестування програмного забезпечення [Електронний ресурс]
//Вікіпедія. - 2015.

16) Достовірний бізнес-план [Електронний ресурс] // Egorfine. -2017

17) Машинне навчання [Електронний ресурс] // Розпізнавання. -2016.

ДОДАТОК А

Технічне завдання

ЗАТВЕРДЖУЮ
Проректор Українського державного
університету науки і технології
Анатолій РАДКЕВИЧ

ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ ТА ЗАСОБІВ ПРОГНОЗУВАННЯ КОРИСТУВАЦЬКИХ ДІЙ ПРИ КОНСТРУЮВАННІ БІЗНЕС ПРОЦЕСІВ ВИРОБНИЦТВА

Технічне завдання

ЛИСТ ЗАТВЕРДЖЕННЯ

44165850.1334-01-ЛЗ

Завідувач кафедри КІТ

_____ Вадим ГОРЯЧКІН

Керівник розробки

_____ Вадим ГОРЯЧКІН

Виконавець

_____ Володимир ПОГРЕБНЯК

Нормоконтролер

_____ Світлана ВОЛКОВА

ЗАТВЕРДЖЕНО

44165850.1334-01-ЛЗ

ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ ТА ЗАСОБІВ ПРОГНОЗУВАННЯ
КОРИСТУВАЦЬКИХ ДІЙ ПРИ КОНСТРУЮВАННІ БІЗНЕС
ПРОЦЕСІВ ВИРОБНИЦТВА

Технічне завдання

Листів 10

2024

АНОТАЦІЯ

Програмний застосунок, розроблений в рамках даного проекту, є інноваційним інструментом для дослідження технологій та засобів прогнозування користувацьких дій при конструюванні бізнес-процесів виробництва. Застосунок призначений для проведення експериментів, аналізу та порівняння різних технологій, спрямованих на оптимізацію та прогнозування користувацької взаємодії та подій у виробничому середовищі.

Основні функціональності:

- збір та обробка даних – забезпечення можливості імпорту та обробки різноманітних даних щодо взаємодій користувачів та виробничих процесів.
- експерименти та пілотні проекти – розробка інструментів для проведення пілотних проектів та експериментів з метою ефективного тестування обраних технологій.
- візуалізація результатів – виведення результатів досліджень.

Очікувані результати - очікується, що програма дозволить здійснювати об'єктивний аналіз, оптимізованих виробничих процесів з урахуванням впливу прогнозування користувацьких дій.

ЗМІСТ

Вступ.....	5
1 Підстава для розробки	6
2 Призначення розробки.....	7
3 Вимоги до програми.....	8
4 Вимоги до програмної документації.....	9
5 Стадії та етапи розробки.....	10
6 Порядок і контроль приймання	11
7 Економічно-технічні показники	12

ВСТУП

В умовах швидкої технологічної трансформації та постійної динаміки виробничих процесів виникає актуальна задача оптимізації бізнес-процесів у сфері виробництва. З метою підвищення ефективності та адаптації до змін, виникає необхідність в розробці та впровадженні передових технологій прогнозування користувачьких дій при конструюванні бізнес-процесів.

Цей проект спрямований на розробку програмного застосунку, який дозволить досліджувати та аналізувати різноманітні технології прогнозування в контексті їхнього впливу на виробничі процеси. Застосунок буде відігравати ключову роль у проведенні експериментів, виборі оптимальних стратегій та підтримці прийняття рішень для оптимізації бізнес-процесів у виробництві.

В рамках даного проекту буде вивчено різні аспекти використання технологій прогнозування, зокрема їхня точність, масштабованість, адаптабельність та інтеграцію з виробничим середовищем. Очікується, що результати досліджень сприятимуть впровадженню ефективних стратегій та технологій для вдосконалення виробничих процесів, підвищуючи тим самим конкурентоспроможність підприємств у галузі виробництва.

1 ПІДСТАВА ДЛЯ РОЗРОБКИ

Основою для розробки є наказ ректора Українського державного університету науки і технології Радкевич А.В. «Про затвердження тем та призначення керівників дипломних проєктів» №1334 ст від 05.12. 2022 року.

Тема проєкту: «Дослідження технологій та засобів прогнозування користувацьких дій при конструюванні бізнес процесів виробництва».

Керівник дипломного проєкту: Горячкін Вадим Миколайович

2 ПРИЗНАЧЕННЯ РОЗРОБКИ

Розробка програмного забезпечення для прогнозування користувацьких дій під час конструювання схем бізнес-процесів у сучасному виробничому середовищі, яка має важливе значення в зв'язку з поєднанням динамічного розвитку технологій та необхідності вдосконалення продуктивності й ефективності виробництва.

3 ВИМОГИ ДО ПРОГРАМИ

Візуалізація результатів: - забезпечення зручного інтерфейсу для візуалізації та аналізу результатів експериментів.

Мова програмування: - розробка на мові програмування javascript, з використанням бібліотеки react.

Інтерфейс: - розробка веб-інтерфейсу, побудованого з використанням react для зручності взаємодії з користувачем.

4 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

До складу документації мають входити:

- текст програми;
- керівництво користувача для користування мобільними додатками.

Вся документація програмного додатку повинна задовольняти вимоги до програмної документації.

5 СТАДІЇ ТА ЕТАПИ РОЗРОБКИ

Стадія	Зміст	Строки виконання
Технічне завдання	Постановка задачі, збір інформації, вибір та обґрунтування критеріїв дослідження.	01.09.23 – 14.09.23
	Попередній вибір методів рішення задач.	14.09.23
	Визначення вимог до технічних засобів.	15.09.23 – 17.09.23
	Узгодження і затвердження технічного завдання.	04.10.23

6 ПОРЯДОК І КОНТРОЛЬ ПРИЙМАННЯ

Контроль за виконанням роботи здійснює керівник дипломного проекту:
доц. Горячкін В. М.

7 ЕКОНОМІЧНО-ТЕХНІЧНІ ПОКАЗНИКИ

Економічно – технічні показники та їх обчислення не були визначені у документах через те, що розробка та дослідження спрямовані на освітні цілі, а не має комерційного характеру

затверджено

44165850.1334 -01

ДОДАТОК Б

Специфікації

**ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ ТА ЗАСОБІВ ПРОГНОЗУВАННЯ
КОРИСТУВАЦЬКИХ ДІЙ ПРИ КОНСТРУЮВАННІ БІЗНЕС**

Специфікація

Листів 2

2024

СПЕЦИФІКАЦІЯ

Позначення	Найменування	Примітка
Документація		
44165850.01334-01-ЛЗ	Лист затвердження	
44165850.01334-01	Специфікація	
44165850.01334-01 12 01-ЛЗ	Лист затвердження	
44165850.01334-01 12 01	Текст програми	
44165850.01334-01 13 01-ЛЗ	Лист затвердження	
44165850.01334-01 13 01	Опис програми	
44165850.01334-01 ІЗ 01-ЛЗ	Лист затвердження	
44165850.01334-01 ІЗ 01	Керівництво користувача.	

ДОДАТОК В

Технічне завдання

ЗАТВЕРДЖУЮ
Проректор Українського державного
університету науки і технологій
Анатолій РАДКЕВИЧ

АНАЛІЗ МОЖЛИВОСТЕЙ МОБІЛЬНИХ ІНТЕРФЕЙСІВ НАВЧАЛЬНИХ
ПРОГРАМ ДЛЯ ДІТЕЙ

Текст програми

ЛИСТ ЗАТВЕРДЖЕННЯ
44165850.01334-01 13 01

Завідувач кафедри КІТ
_____Вадим ГОРЯЧКІН
Керівник розробки
_____Вадим Горячкін
Виконавець
_____Володимир ПОГРЕБНЯК
Нормоконтролер
_____Світлана ВОЛКОВА

2024

1 ТЕКСТ ПРОГРАММЫ

```

Index.js
import { StrictMode } from "react";
import { createRoot } from "react-dom/client";

import App from "./App";

const rootElement = document.getElementById("root");
const root = createRoot(rootElement);

root.render(
  <StrictMode>
    <App />
  </StrictMode>
);

App.js
import React, { Component } from "react";

const Item = ({
  color,
  id,
  className = "",
  onClick,
  onMouseEnter,
  onMouseLeave,
}) => {
  return (
    <div
      className={['item item--${color}', className].join(" ")
      onClick={onClick}
      onMouseEnter={onMouseEnter}
      onMouseLeave={onMouseLeave}
    >
      {id}
    </div>
  );
};

export default class App extends Component {
  constructor(props) {
    super(props);

    this.actions = [

```

```

    { id: 0, color: "red" },
    { id: 1, color: "yellow" },
    { id: 2, color: "green" },
    { id: 3, color: "blue" },
  ];

  this.state = {
    history: [],
    hovered: null,
  };

  this.makeAction = this.makeAction.bind(this);
}

makeAction(action) {
  this.setState((ps) => ({
    ...ps,
    history: [...ps.history, action],
  }));
}

render() {
  const { history, hovered } = this.state,
    lastAction = history[history.length - 1],
    lastActionId = lastAction ? lastAction.id : null;

  return (
    <div className="wrapper">
      <div className="actions">
        <div className="available">
          <h1>Доступні дії</h1>
          <div className="items">
            {this.actions.map((action) => (
              <Item {...action} onClick={() => this.makeAction(action)} />
            ))}
          </div>
        </div>
        <div className="prognosed">
          <h1>Спрогнозовані дії</h1>
          <div className="items">
            {history
              .reduce((res, item, index, array) => {
                if (item.id === lastActionId && index + 1 < array.length) {
                  let newElem = array[index + 1],
                      existedIndex = res.findIndex((_) => _.id === newElem.id);

```

```

    if (existedIndex >= 0) {
      return res
        .slice(0, existedIndex)
        .concat(
          Object.assign({}, res[existedIndex], {
            priority: res[existedIndex].priority + 1,
          })
        )
        .concat(res.slice(existedIndex + 1, res.length));
    } else {
      return res.concat(
        Object.assign({}, newElem, { priority: 1 })
      );
    }
  } else {
    return res;
  }
}, [])
.sort((a, b) => b.priority - a.priority)
.map((item) => (
  <Item
    {...item}
    onClick={() => {
      this.setState(
        (ps) => ({ ...ps, hovered: null }),
        () => this.makeAction(item)
      );
    }}
    onMouseEnter={() =>
      this.setState((ps) => ({ ...ps, hovered: item.id }))
    }
    onMouseLeave={() =>
      this.setState((ps) => ({ ...ps, hovered: null }))
    }
  />
  )))
</div>
</div>
</div>
<div className="history">
  <h1>Історія викликів</h1>
  <div className="items">
    {history.map((item, index, array) => {
      let prevItem = array[index - 1];

```

```

    let nextItem = array[index + 1];
    return (
      <Item
        {...item}
        className={
          hovered === null
            ? ""
            : (item.id === lastActionId &&
              (nextItem && nextItem.id) === hovered) ||
              (item.id === hovered &&
              (prevItem && prevItem.id) === lastActionId) ||
              index === array.length - 1
            ? ""
            : "unhovered"
          }
      />
    );
  }
}
</div>
</div>
</div>
);
}
}

```

Package.json

```

{
  "name": "react",
  "version": "1.0.0",
  "description": "",
  "keywords": [],
  "main": "src/index.tsx",
  "dependencies": {
    "react": "18.2.0",
    "react-dom": "18.2.0",
    "react-scripts": "5.0.1"
  },
  "devDependencies": {
    "@types/react": "18.2.38",
    "@types/react-dom": "18.0.9",
    "loader-utils": "3.2.1",
    "typescript": "4.4.2"
  },
  "scripts": {
    "start": "react-scripts start",

```

```

    "build": "react-scripts build",
    "test": "react-scripts test --env=jsdom",
    "eject": "react-scripts eject"
  },
  "browserslist": [">0.2%", "not dead", "not ie <= 11", "not op_mini all"]
}

```

Index.html

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="utf-8" />
```

```
<meta
```

```
  name="viewport"
```

```
  content="width=device-width, initial-scale=1, shrink-to-fit=no"
```

```
</>
```

```
<meta name="theme-color" content="#000000" />
```

```
<!--
```

manifest.json provides metadata used when your web app is added to the homescreen on Android. See

<https://developers.google.com/web/fundamentals/engage-and-retain/web-app-manifest/>

```
-->
```

```
<link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
```

```
<link rel="shortcut icon" href="%PUBLIC_URL%/favicon.ico" />
```

```
<!--
```

Notice the use of `%PUBLIC_URL%` in the tags above.

It will be replaced with the URL of the `public` folder during the build.

Only files inside the `public` folder can be referenced from the HTML.

Unlike `/favicon.ico` or `favicon.ico`, `%PUBLIC_URL%/favicon.ico` will work correctly both with client-side routing and a non-root public URL.

Learn how to configure a non-root public URL by running `npm run build`.

```
-->
```

```
<title>React App</title>
```

```
</head>
```

```
<body>
```

```
<noscript> You need to enable JavaScript to run this app. </noscript>
```

```
<div id="root"></div>
```

```
<!--
```

This HTML file is a template.

If you open it directly in the browser, you will see an empty page.

You can add webfonts, meta tags, or analytics to this file.

The build step will place the bundled scripts into the `<body>` tag.

To begin the development, run ``npm start`` or ``yarn start``.

To create a production bundle, use ``npm run build`` or ``yarn build``.

```
-->  
</body>  
</html>
```

ДОДАТОК Г

Тези доповіді

Міністерство освіти і науки України
 Міністерство інфраструктури України
 Український державний університет науки та технологій
 Східний науковий центр транспортної академії наук



TEMPUS: CITISET & SEREIN & CRENG

ТЕЗИ
XVII Міжнародної науково-практичної конференції
«СУЧАСНІ ІНФОРМАЦІЙНІ ТА КОМУНІКАЦІЙНІ
ТЕХНОЛОГІЇ НА ТРАНСПОРТІ, В ПРОМИСЛОВОСТІ ТА ОСВІТІ»

ABSTRACTS
of the XVII International Conference
«MODERN INFORMATION AND COMMUNICATION TECHNOLOGIES
ON A TRANSPORT, IN INDUSTRY AND EDUCATION»

13.12.2023 – 14.12.2023

Дніпро
2023

Розробка програмного забезпечення для прогнозування користувацьких дій у виробничому середовищі.....	87
Погребняк В. М., Горячкін В. М., Український державний університет науки і технологій	
Що привносить когнітивність до основ нечіткої математики?.....	88
Прокопчук Ю. О., Інститут технічної механіки НАН і ДКА України, Україна	
Дослідження методів та засобів аутентифікації користувача у реальному часі шляхом розпізнавання обличчя.....	89
Різниченко М. О., Горбова О. В., Український державний університет науки і технологій, Україна	
Поведінкові стратегії Інтернет бізнесу.....	90
Савчук Л. М., Ковальчук Є. В. Український державний університет науки і технологій, Інститут промислових та бізнес технологій, Україна	
Вибір цілі, критерію і норми при проектуванні інтелектуальної системи управління організаційно-технічними процесами	91
Самойлов С.П., Український державний університет науки і технологій, Україна	
Розпізнавання рукописних символів за допомогою нейронної мережі	92
Середа О.А., Горячкін В.М., Український державний університет науки і технологій	
Підвищення якості коду мобільних застосунків методами рефакторингу з використанням генеративного штучного інтелекту.....	93
Сирота О. А., Горячкін В.М., Український державний університет науки і технологій	
Багатовимірні нечіткі моделі моніторингу і реабілітації з нерівномірним інтервалом спостережень.....	94
Скалозуб В.В., Горячкін В.М., Клименко І.В., Мурашов О.В., Український державний університет науки і технологій	
Процедури редукції і каппа статистики у завданнях формування достовірних математичних моделей класифікації невизначених даних.....	95
Скалозуб В.В., Терлецький І.А., Дудник І.О., Український державний університет науки і технологій, Скалозуб М. PayPal, Швеція	
Порівняльний аналіз комп'ютерних моделей прогнозування часових послідовностей з нерівномірним і нечітким кроком спостережень	96
Скалозуб В.В., Медич С.О., Старина А.К., Український державний університет науки і технологій, Україна	
Розробка користувацької частини системи моніторингу пасіки	97
Терещенко Д. І., Дзюба В. В., Український державний університет науки і технологій, Україна	
Цифрові двійники (digital twin).....	98
Тимашов О.О., Інститут кібернетики імені В.М.Глушкова НАН України, Україна, Самойлов С.П., Український державний університет науки і технологій	
Використання спрощеної форми проєкту предметно-орієнтованого програмного забезпечення.....	99
Чигір Р.Р., Шинкаренко В.І., Український державний університет науки і технологій, Україна	

Розробка програмного забезпечення для прогнозування користувацьких дій у виробничому середовищі

Розробка програмного забезпечення для прогнозування користувацьких дій під час конструювання схем бізнес-процесів у сучасному виробничому середовищі має важливе значення в зв'язку з поєднанням динамічного розвитку технологій та необхідності вдосконалення продуктивності й ефективності виробництва.

Застосування такого програмного забезпечення дає можливість адаптуватися до змінних умов ринку та вимог споживачів, прогнозуючи їхні взаємодії з бізнес-системою. Розуміння користувацької поведінки стає вирішальним фактором у вирішенні викликів, пов'язаних з оптимізацією процесів, підвищенням рівня персоналізації та покращенням якості продукції чи послуг.

Розробка моделей прогнозування користувацьких дій є складним процесом, який включає кілька ключових етапів для ефективного прогнозування поведінки користувачів.

- збір з різних джерел (сайти, додатки, опитування) та їх обробка для подальшого використання в моделях, їх очищення, а також виявлення та виправлення пропущених значень та аномалій;

- вибір моделей, під час якого після підготовки даних обираються методи для розробки моделей прогнозування, такі як машинне навчання, статистичні моделі або глибинне навчання;

- тренування моделей на наявних даних для вивчення патернів та залежностей, підбір параметрів моделі;

- валідація та оцінка точності та ефективності прогнозування на тестових даних, виявлення можливостей для їх вдосконалення.

- реалізація та впровадження моделі для прогнозування користувацьких дій у реальних умовах або тестових середовищах.

Цей цикл може бути ітеративним, з постійним удосконаленням моделі через оновлення даних, додавання нового функціоналу та вдосконалення алгоритмів.

Відповідне програмне забезпечення надає можливість не лише прогнозувати користувацькі дії, але й попередньо аналізувати та планувати свої дії відповідно до цих прогнозів, оперативно мінімізуючи ризики, збільшуючи ефективність та забезпечуючи конкурентні переваги. Зокрема у виробничому середовищі, де кожний крок у бізнес-процесі має вагому значимість, вірогідність передбачити дії користувачів може позитивно вплинути на планування ресурсів, оптимізацію виробництва та зниження витрат.

Тим самим, розробка програмного забезпечення для прогнозування користувацьких дій є важливим кроком у підвищенні продуктивності, відповіді на вимоги ринку та забезпеченні конкурентоздатності виробничих підприємств у сучасному динамічному економічному оточенні.

Програмне забезпечення для прогнозування користувацьких дій розкриває можливості для оперативного аналізу та планування дій компаній на основі цих прогнозів. Цей підхід дозволяє мінімізувати ризики, підвищувати ефективність та надавати конкурентні переваги. Навіть у виробничому середовищі, де кожен етап бізнес-процесу має велике значення, передбачення користувацьких дій сприяє точному плануванню ресурсів, оптимізації виробництва та ефективному використанню ресурсів.

Таким чином, програмне забезпечення для прогнозування користувацьких дій стає ключовим інструментом для підвищення продуктивності та конкурентоздатності виробничих підприємств у сучасному економічному середовищі, сприяючи їх успішному розвитку.