

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

**АВТОМАТИЗОВАНІ СИСТЕМИ
УПРАВЛІННЯ ТА ПРИЛАДИ
АВТОМАТИКИ**

Всеукраїнський міжвідомчий
науково-технічний збірник

Заснований у 1965 р.

Випуск 182

Харків
2024

У збірнику наведено результати досліджень, що стосуються використання моделей і методів штучного інтелекту та прийняття рішень під час створення та експлуатації інформаційних систем управління підприємствами та організаціями. Запропоновано нові підходи та рішення в галузі управління IT-проєктами, створення систем пояснювального інтелекту та аналізу функціональних вимог до інформаційних систем управління підприємствами.

Для викладачів університетів, науковців, фахівців, аспірантів.

The collection presents the results of research on the use of artificial intelligence models and methods and decision-making in the creation and operation of information systems for managing enterprises and organizations. New approaches and solutions in the field of IT project management, the creation of explanatory intelligence systems and the analysis of functional requirements for information systems for managing enterprises are proposed.

For university professors, researchers, specialists, graduate students.

Редакційна колегія:

В.В. Семенець, д-р техн. наук, проф. (гол. ред.), *В.М. Левикін*, д-р техн. наук, проф. (відпов. ред.), *М.В. Євланов*, д-р техн. наук, проф. (відпов. секр.), *Є.В. Бодянський*, д-р техн. наук, проф., *І.В. Гребеннік*, д-р техн. наук, проф., *А.Л. Єрохін*, д-р техн. наук, проф., *А.О. Каргін*, д-р техн. наук, проф., *Б.І. Мороз*, д-р техн. наук, проф., *І.Ш. Невлюдов*, д-р техн. наук, проф., *К.Е. Петров*, д-р техн. наук, проф., *І.В. Рубан*, д-р техн. наук, проф., *С.Г. Удовенко*, д-р техн. наук, проф., *О.Є. Федорович*, д-р техн. наук, проф., *В.О. Філатов*, д-р техн. наук, проф., *Г.З. Халімов*, д-р техн. наук, проф.

Рішення Національної ради про реєстрацію
Ідентифікатор медіа

№ 1410 від 25.04.2024 р.
R30-03874

Адреса редакційної колегії: Україна, 61166, Харків, просп. Науки, 14, Харківський національний університет радіоелектроніки, кімн. 254, тел. (057) 70-21-451

© Харківський національний університет
радіоелектроніки, 2024

ЗМІСТ

ЄВЛАНОВ М.В., ЮР'ЄВ І.О., КРУК Б.Є. РОЗРОБКА МЕТОДУ ВИРШЕННЯ ЗАДАЧІ ФОРМУВАННЯ І КЛАСИФІКАЦІЇ ЗАПИТІВ НА ЗМІНИ ІТ-ПРОДУКТУ	5
ЧАЛИЙ С.Ф., ЛЕЩИНСЬКИЙ В.О. ПОБУДОВА ПРОЦЕСНО-ОРІЄНТОВАНИХ ПОЯСНЕНЬ В ІНТЕЛЕКТУАЛЬНИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ НА ОСНОВІ МОЖЛИВІСНИХ КАУЗАЛЬНИХ ЗАЛЕЖНОСТЕЙ	25
ЧАЛА О.В., БОГАТОВ Є.О. РОЗРОБКА ТЕМПОРАЛЬНОЇ МОДЕЛІ ПРЕДСТАВЛЕННЯ ЗНАНЬ ЩОДО БІЗНЕС-ПРОЦЕСІВ ПЕРШОГО РІВНЯ ЗРІЛОСТІ ПРОЦЕСНОГО УПРАВЛІННЯ	34
ВАСИЛЬЦОВА Н.В., БУРКОВСЬКА А.С. ВИБІР ПІДМНОЖИНИ МЕТОДІВ РОЗРОБКИ ІНТЕРФЕЙСУ МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ ВИВЧЕННЯ МОВ	42
ЧАЛИЙ С.Ф., ЛЕЩИНСЬКА І.О. УТОЧНЕННЯ МЕНТАЛЬНОЇ МОДЕЛІ РІШЕННЯ НА ОСНОВІ ДОПОВНЕННЯ ВХІДНИХ ДАНИХ В ЗАДАЧІ ФОРМУВАННЯ ПОЯСНЕНЬ В ІНТЕЛЕКТУАЛЬНІЙ СИСТЕМІ	66
ЄВЛАНОВ М.В., МОРОЗ Б.І., МОРОЗ Д.М., ЛУЧИЦЬКИЙ В.В. ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ВИЯВЛЕННЯ ТЕРМІНІВ ТА АРТЕФАКТІВ ПРОСКТУ У ВИМОГАХ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ	73
БОДЯНСЬКИЙ Є.В., ЧАЛА О.С. ШВИДКА КЛАСИФІКАЦІЯ В ONLINE- ТА NEARLINE-РЕЖИМАХ В УМОВАХ КЛАСІВ, ЩО ПЕРЕТИНАЮТЬСЯ	93
РЕФЕРАТИ	101

CONTENT

YEVLANOV M.V., YUR`EV I.O., KRUK B.YE. DEVELOPMENT OF A METHOD FOR SOLVING THE TASK OF FORMING AND CLASSIFYING REQUESTS FOR IT PRODUCT CHANGES	5
CHALY S.F., LESHCHYNSKYI V.O. DEVELOPMENT OF PROCESS-ORIENTED EXPLANATIONS IN INTELLIGENT INFORMATION SYSTEMS BASED ON POSSIBILISTIC CAUSAL DEPENDENCIES	25
CHALA O.V., BOGATOV YE.O. DEVELOPMENT OF A TEMPORAL MODEL FOR KNOWLEDGE REPRESENTATION REGARDING BUSINESS PROCESSES OF THE FIRST LEVEL OF PROCESS MANAGEMENT MATURITY	34
VASYLTSOVA N.V., BURKOVSKA A.S. SELECTION OF A SUBSET OF METHODS FOR DEVELOPING THE INTERFACE OF A MOBILE APPLICATION FOR LANGUAGE LEARNING	42
CHALY S.F., LESHCHYNSKA I.O. REFINEMENT OF THE MENTAL MODEL OF A SOLUTION BASED ON INPUT DATA AUGMENTATION IN THE TASK OF GENERATING EXPLANATIONS IN AN INTELLIGENT SYSTEM	66
YEVLANOV M.V., MOROZ B.I., MOROZ D.M., LUCHYTSKYI V.V. INFORMATION TECHNOLOGY FOR IDENTIFYING TERMS AND PROJECT ARTIFACTS IN THE REQUIREMENTS FOR THE INFORMATION SYSTEM	73
YE. V. BODYANSKIY, O.S. CHALA FAST CLASSIFICATION IN ONLINE AND NEARLINE MODES IN CONDITIONS OF OVERLAPPING CLASSES	93
ABSTRACTS.....	101

М.В. СВЛАНОВ, І.О. ЮР'ЄВ, Б.С. КРУК

РОЗРОБКА МЕТОДУ ВИРІШЕННЯ ЗАДАЧІ ФОРМУВАННЯ І КЛАСИФІКАЦІЇ ЗАПИТІВ НА ЗМІНИ ІТ-ПРОДУКТУ

Розглянуто основні існуючі інформаційні технології, моделі і методи формування запитів на зміни ІТ-продуктів під час виконання ІТ-проектів. Встановлено, що єдиної сталої системи класифікації RFC, яка охоплювала б увесь життєвий цикл ІТ-проекту, не існує. Розроблено теоретико-множинну модель запиту на зміну, який виникає під час виконання ІТ-проекту розробки чи модифікації ІТ-продукту. На основі цієї моделі розроблено загальний метод вирішення задачі формування і класифікації запитів на зміну ІТ-продукту.

1. Вступ

Управління змінами – це комплексний, циклічний і структурований підхід до зміни окремих осіб, груп, і організації з поточного стану в майбутній стан із передбачуваними бізнес-вигодами [1], [2]. Цей підхід зараз описано у вигляді фреймворку – конвенцій, принципів і методів управління змінами в галузі управління програмами, портфелями проектів та окремими проектами. Цей фреймворк охоплює тему стратегічної гнучкості як відображення здатності організації відчувати зовнішні чи внутрішні загрози і можливості та реагувати на них. Стратегічна гнучкість визначається як здатність бізнесу завчасно оцінювати і реалізувати зміни бізнес-середовища та демонструвати стійкість до непередбачених змін [2].

Фреймворк управління змінами розглядає перехід з поточного стану в майбутній стан із передбачуваними бізнес-вигодами як процес «Управління змінами» із такими підпроцесами [2]:

- а) формулювання змін – підпроцес, який здійснюється шляхом виявлення та уточнення потреби у змінах, оцінки готовності до змін та окреслення масштабу змін;
- б) планування змін – підпроцес, який здійснюється шляхом визначення підходу до змін і планування залучення зацікавлених сторін як переходів та інтеграції;
- в) впровадження змін – підпроцес, який здійснюється шляхом підготовки організації до змін, мобілізації зацікавлених сторін та досягнення результатів проекту;
- г) управління переходом до змін – підпроцес, який здійснюється шляхом переведення результатів у бізнес-операції, вимірювання рівня впровадження та результатів і переваг змін, а також коригування плану для усунення розбіжностей;
- д) підтримка змін на постійній основі – підпроцес, який здійснюється шляхом спілкування, консультації та представництва зацікавлених сторін, проведення смислотворчої діяльності та вимірювання реалізації вигод.

Застосування розглянутого фреймворку управління змінами під час управління таким різновидом проектів, як ІТ-проекти, вимагає приділення особливої уваги підпроцесу «Формулювання змін» і, зокрема, роботі «Визначте/уточніть потребу в змінах». Ця увага обумовлена тим, що саме ця робота є першою роботою, в результаті виконання якої ІТ-проект та система управління цим проектом отримує інформацію про необхідність зміни та перший опис цієї зміни.

Але, на жаль, існуючий фреймворк управління змінами, розглянутий у [2], та стандарти і зведення знань з управління проектами та, зокрема, ІТ-проектами [1], [3] не дають конкретних рекомендацій з виконання підпроцесу «Формулювання змін» і роботи «Визначте/уточніть потребу в змінах». Так у [2] вказано два таких варіанти виконання

цього підпроцесу та роботи:

а) якщо проєкт є частиною програми, визначення та роз'яснення потреби в змінах повинно бути добре розробленою і сформульованою частиною статуту проєкту;

б) якщо проєкт не є частиною програми, необхідно під час управління портфелем окремих проєктів підприємства виконати відповідні дії, щоб переконатися, що уточнення потреб в змінах виконано до завершення статуту проєкту.

Таким чином, незалежно від особливостей ІТ-проєкту, потреби в змінах та перші описи можливих змін згідно з існуючим фреймворком управління змінами повинні бути сформульовані до завершення робіт зі складання та ухвалення статуту ІТ-проєкту. При цьому роботу «Визначте/уточніть потребу в змінах» пропонується розглядати як різновид робіт з формування вимог, які висуваються до ІТ-продукту та ІТ-проєкту перед початком планування та виконання цього проєкту [2]. Таке позиціонування підпроцесу «Формулювання змін» і роботи «Визначте/уточніть потребу в змінах» значно обмежує можливість внесення змін до ІТ-проєкту під час його планування та виконання і, відповідно, зменшує гнучкість проєкту та можливості його адаптації до обставин, що постійно змінюються.

Слід зазначити, що існуючий фреймворк управління змінами передбачає можливість виникнення змін під час виконання проєкту. Зокрема, згідно з цим фреймворком, керівнику проєкту може знадобитися виконати завдання, які виходять за встановлені межі цього проєкту. В ідеалі ці додаткові завдання мають бути частиною узгодженого обсягу проєкту, щоб забезпечити успішне виконання змін. Інакше керівник проєкту повинен попередити спонсора проєкту та вимагати зміни обсягу проєкту [2]. Таким чином, виходить замкнене коло: керівник проєкту повинен прийняти рішення про виконання роботи «Визначте/уточніть потребу в змінах», базуючись на оцінках обсягу ІТ-проєкту до та після внесення зміни, яка пропонується. Але ці оцінки неможливо розрахувати, не знаючи зміст зміни, яка пропонується, тобто не виконавши роботу «Визначте/уточніть потребу в змінах».

Тому виникає необхідність у проведенні науково-прикладних досліджень, які дозволили б формалізувати виконання підпроцесу «Формулювання змін» і роботи «Визначте/уточніть потребу в змінах» під час виконання ІТ-проєкту незалежно від конкретних особливостей цих змін та ІТ-проєктів.

2. Аналіз літературних даних і постановка проблеми дослідження

Особливості виконання процесів, підпроцесів та окремих робіт існуючого фреймворку управління змінами визначаються в загальному випадку прийнятим у [2] життєвим циклом (ЖЦ) управління змінами. Його схема наведена на рис. 1 [2].

За наведеною на рис. 1 схемою, у [2] запропоновано розглядати кожен з розглянутих підпроцесів управління змінами як множину окремих робіт, кожна з яких виконується у встановленому фреймворком порядку (але не обов'язково суворо послідовно одна за одною). Весь процес управління змінами та його підпроцеси у [2] запропоновано описувати з використанням ітераційної моделі ЖЦ з врахуванням можливості постійного виникнення адаптивних змін у відповідь на зміни обставин.

Таке представлення ЖЦ управління змінами дозволяє мінімізувати зв'язки між окремими підпроцесами та роботами існуючого фреймворку управління змінами і розглядати їх як окремі самостійні об'єкти наукових досліджень. На технологічному рівні це означає можливість створення та застосування для автоматизації відповідних підпроцесів та робіт окремих інформаційних технологій (ІТ), які взаємодіють між собою за сервіс-орієнтованою парадигмою.

Для того, щоб можна було оцінити зміну обсягу ІТ-проєкту після внесення зміни, яка пропонується, треба визначити, якої саме частини проєкту стосується ця зміна. Тому виникає необхідність дослідити і проаналізувати існуючі класифікації запитів на зміни (request for change, RFC), які можуть локалізувати місце впливу RFC на обсяг ІТ-проєкту. Такі класифікації є спробою формалізувати різновиди RFC, базуючись, зазвичай, на таких результатах:

- а) досвід управління змінами в окремій українській ІТ-компанії;
- б) досвід управління змінами в закордонних ІТ-проєктах, які відносяться до одного й того ж різновиду.

Типовим прикладом класифікації, яка базується на результатах групи а), є класифікація, запропонована у [4]. За цією класифікацією, уся множина RFC поділяється на такі класи:

- зміни системної логіки/поведінки;
- зміни скоупу (під цим терміном тут слід розуміти обсяг, межі або масштаб ІТ-проєкту чи ІТ-продукту);
- зміни в назвах/формулюваннях;
- зміни в інтерфейсі користувача;
- технічні зміни (інтеграції, мови програмування тощо).

Як можливий окремий клас RFC у [4] пропонується розглядати запит на видалення елементів зі скоупу.

Але запропонована у [4] класифікація характеризується тим, що вона не використовується навіть самим автором під час визначення особливостей виконання робіт процесу «Управління змінами». Так у [4] представлена візуальна модель кастомного процесу управління змінами, адаптованого під команди, з якими працював автор. Але головною проблемою прийняття рішення щодо подальшого управління зміною ІТ-проєкту за цією моделлю визнано проблему відділення потреб зацікавлених сторін проєкту у змінах від потреб цих же сторін у ліквідації знайдених програмних помилок. Виходячи з цього, пропонуються описи дій з управління RFC, які незначно відрізняються від загальних рекомендацій загального фреймворку управління змінами за ступенем деталізації. Тому можна зробити висновок, що запропонований варіант класифікації є декларативним та суб'єктивним, тому що він не впливає на виконання робіт процесу управління змінами та підпроцесу «Формулювання змін».

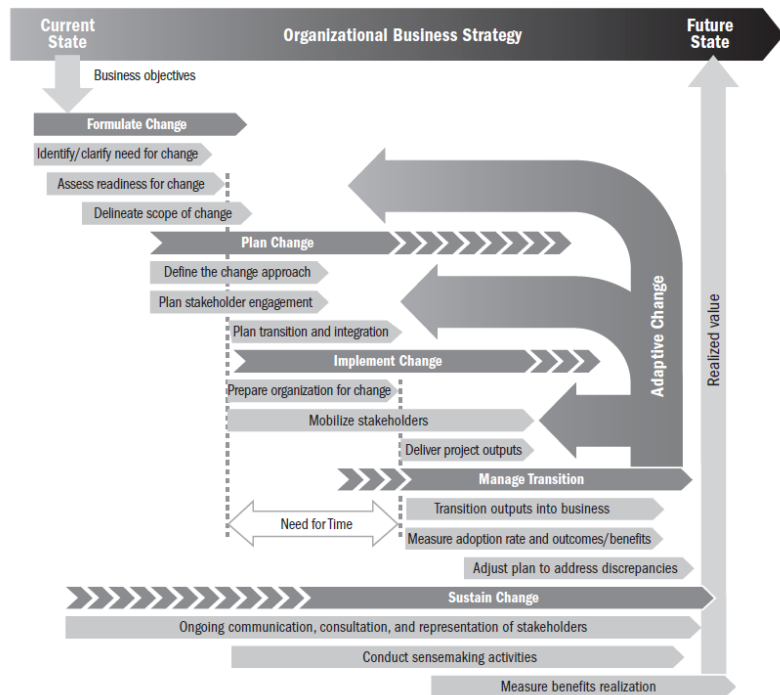


Рис. 1. Схема життєвого циклу управління змінами [2]

Результатом застосування групи б) результатів є класифікації RFC, які виникли та виникають внаслідок узагальнення досвіду закордонних ІТ-компаній. На відміну від результатів групи а), ці класифікації є результатом наукового узагальнення одного з таких варіантів досвіду:

- а) досвід управління змінами на окремій стадії ІТ-проектів;
- б) досвід управління змінами в ІТ-проектах, які належать до одного й того ж різновиду;
- в) узагальнений досвід управління змінами різних ІТ-проектів.

Як приклад класифікації RFC, яка використовується для управління змінами на окремій стадії ІТ-проекту, пропонується розглянути запропоновану у [5] класифікацію технічних (або інженерних) змін. Така зміна у [6] визначається як зміна деталей, креслень або програмного забезпечення, які вже були випущені в процесі розробки продукту. Ця зміна може бути будь-якого розміру або типу; вона може включати будь-яку кількість людей і тривати протягом будь-якого часу [5], [6]. Запит на технічну зміну визначається як RFC, створений для інформування про вихід проекту чи продукту із цільових значень [7]. За іншим визначенням, запит на технічну зміну – це запит на технічну зміну стану моделі продукту, яка вже була випущена для подальшої розробки або виробництва [5].

Для вирішення задачі класифікації авторами у [5] було розроблено атрибутивну модель запиту на технічну зміну за результатами дослідження літературних джерел та великої кількості текстів запитів на технічні зміни. Ця модель наведена на рис. 2 [5].

Attribute		Characteristic value					
Object of change	Concerned object	Document	Component	Process	Module	Equipment	Product
	Concerned object specification	Physical Specification	Functionality	Software	Procedures	Information	
	Extend of change*	Document	Component	Process	Module	Equipment	Product
	Volume of change	Single position		Group		All	
Properties	Phase of life-cycle	Development	Prototyping / Series start-up	Series production	Use phase		
	Architecture	Modular		Partly integrated	Integrated		
	Complexity	Low		Medium	High		
Influencing factors	Root of change	Customer requirement	Law / Certification	Safety	Cost	Technology	Productivity
	Reason of change	Current situation faulty		Optimization of the current situation	Target requirements changed	Incorrect target requirements	
	Urgency	Change to next release		Important, short-term change	Immediate change		
	Concerned Stakeholder	R&D	R&D and other departments		Involvement of the management	Involvement of external stakeholders	
Effect	Effect on cost	Decreasing	Slightly decreasing	Slightly increasing	Increasing		
	Effect on time schedules	Low		Medium	High		
	Effect on the product	None	Necessary correction	Fulfillment of individual customer requirements	Improvement of functionality	Quality improvement	

Descriptive attributes
 Evaluating attributes (expert knowledge required)

*The CVs of concerned object and extend of change are the same. The difference is, that concerned object identifies the objects from where the change originates, while extend of the change describes objects that are impacted by the change of the object. Example: If the dimension of a component (=object) is changed, most likely the drawing document will be changed too (=extend of change)

Рис. 2. Атрибутивна модель запиту на технічну зміну [5]

Для використання моделі, наведеної на рис. 2, у [5] було розроблено рішення задачі автоматичної класифікації запитів на технічні зміни. Загальне графічне представлення цієї задачі наведено на рис. 3 [5].

Для реалізації цієї задачі у [5] запропоновано формалізувати опис запиту на технічну зміну як кортеж таких текстових атрибутів:

- а) «Причина зміни»;
- б) «Компонент/агрегат»;

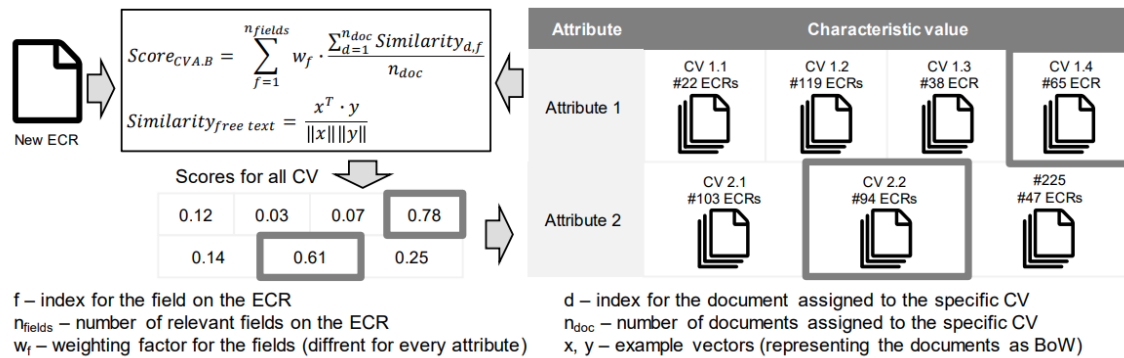


Рис. 3. Графічне представлення задачі автоматичної класифікації запитів на технічні зміни [5]

- в) «Опис поточної ситуації»;
- г) «Вигода/ціль зміни»;
- д) «Запропонована реалізація».

Апробацію ІТ, яка реалізувала цю задачу класифікації, проводили на 40 запитах на технічні зміни. За результатами апробації було встановлено, що ця технологія досягла загальної точності вирішення задачі класифікації запитів на технічні зміни 79 %, а максимальна точність дорівнювала 90 % [5].

Як інший приклад класифікації RFC, яка використовується для управління змінами на окремій стадії ІТ-проекту, розглянемо запропоновану у [8] онтологічну модель. Ця модель описує класи RFC, які виникають під час змін у системних вимогах, що висунуті до створюваного ІТ-продукту. Модель представлено у вигляді діаграми класів, яка відображає ієрархію класів RFC системних вимог (див. рис. 4). Специфікації значень важливих елементів цієї онтологічної моделі наведено у табл. 1.

Як приклад класифікації RFC, яка використовується для управління змінами в ІТ-проектах, які належать до одного й того ж різновиду, розглянемо класифікацію, яку у [9] запропонували використовувати під час управління змінами в ІТ-проектах розробки мобільних застосунків. Схема ЖЦ релізу розробки мобільного застосунку із впровадженою технологією управління змінами наведено на рис. 5 [9]. На цьому рисунку пунктирною лінією виділено основу технології управління змінами та її інтерфейси, які передають встановлені зміни до інструментальних засобів, що використовуються на різних стадіях відповідного ІТ-проекту. Схему підходу, який знаходиться в основі цієї технології, наведено на рис. 6 [9].

Основними обмеженнями системи управління змінами на основі запропонованої на рис. 5 схеми ЖЦ є [9]:

- низька точність класифікації та кластеризації через низьку якість відгуків користувачів;
- труднощі в роботі з різними словниками, які використовуються для опису досвіду використання користувачами застосунків;
- існування розривів в словниках ІТ-проекту між термінами з відгуків користувачів і позначеннями артефактів програмного забезпечення.

Тому в наведеній на рис. 6 технології для подолання цих обмежень запропоновано поділити усі можливі відгуки користувачів на такі чотири категорії [9]:

- а) «Надання інформації» (речення, які інформують або оновлюють інформацію для користувачів або розробників щодо аспекту, пов'язаного з програмою);

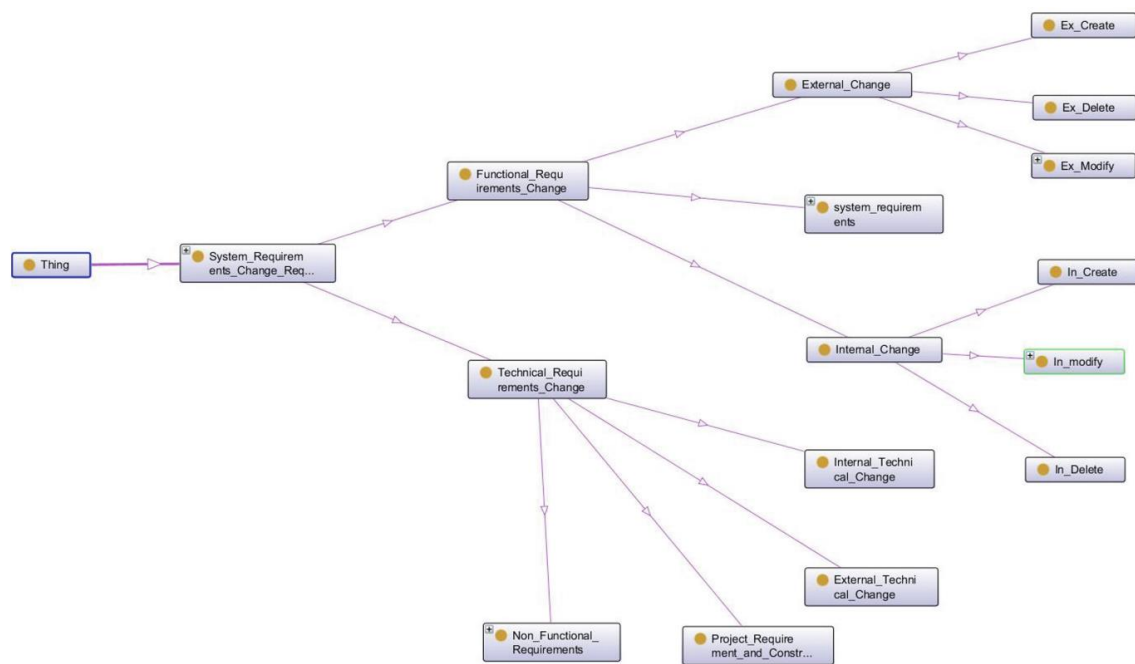


Рис. 4. Онтологічна модель класифікації запитів на зміни системних вимог [8]

Таблиця 1

Специфікації класів онтологічної моделі

Клас	Опис
System requirements change requests (RFC системних вимог)	Запити майбутніх користувачів на зміни системних вимог зі сховища вимог
Functional change (Функціональна зміна)	Запити майбутніх користувачів на зміни, які впливають на функціональні вимоги користувачів
Technical change (Технічна зміна)	Запити майбутніх користувачів на зміни, які впливають на нефункціональні вимоги або проєктні вимоги та обмеження
System requirements (Системні вимоги)	RFC комунікаторів – внутрішніх (системні аналітики, розробники, проєктувальники тощо) або зовнішніх (оператори, користувачі, керівники тощо)
Delete (Видалити)	Містить вимоги, які необхідно видалити
Modify (Модифікувати)	Містить вимоги, які необхідно модифікувати
Create (Створити)	Містить вимоги, які необхідно створити

б) «Пошук інформації» (інформація про тих, хто намагається отримати інформацію або допомогу);

в) «Запит на функцію» (ідеї, пропозиції чи потреби щодо покращення функціональності та продуктивності програми);

г) «Виявлення проблеми» (речення, що описують проблеми з програмами або їхню несподівану поведінку).

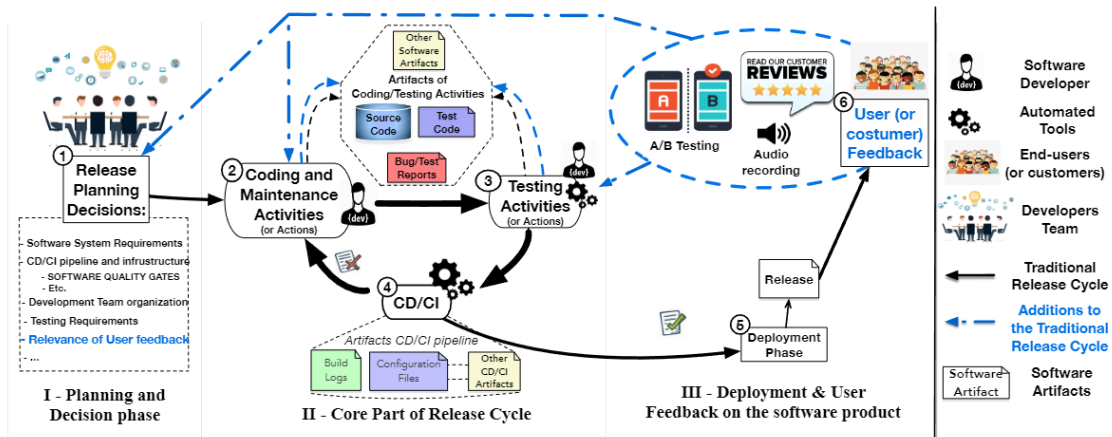


Рис. 5. Схема життєвого циклу розробки релізу мобільного застосунку із вбудованою системою управління змінами [9]

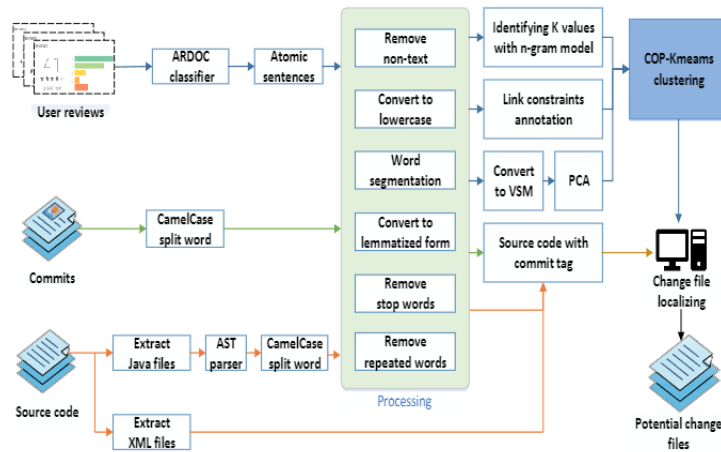


Рис. 6. Схему підходу, який знаходиться в основі запропонованої у [9] технології управління змінами

Для перетворення інформації з відгуків користувачів у RFC у [9] запропоновано вирішити задачу класифікації відгуків користувачів, виділяючи для подальшої обробки тільки ті відгуки, які слід віднести до категорій в) або г), наведених вище. Для цього застосовано ARDOC-класифікатор, який перетворює відгуки користувачів на окремі речення, а потім класифікує ці речення в одну з чотирьох згаданих вище категорій.

Недоліком такої класифікації є надто примітивний поділ усіх RFC на два великих класи (категорії в) та г)). Тому підмножина відгуків користувачів, які за допомогою ARDOC-класифікатора визнали такими, що належать до категорій в) або г), розглянутих вище, далі обробляється шляхом вирішення задачі кластеризації. Результатом вирішення цієї задачі є розподіл усіх відгуків користувачів вихідної підмножини на окремі кластери, які визначають актуальні для поточної ітерації ЖЦ релізу мобільного застосунку класи RFC. Для вирішення задачі кластеризації у [9] запропоновано використовувати алгоритми сімейства k-means із додатковими обмеженнями.

Як приклад класифікації RFC, яка враховує узагальнений досвід управління змінами різних IT-проектів, пропонується розглянути класифікацію RFC, що формуються під час перегляду програмного коду. Для цього у [10] було проаналізовано 10 різних IT-проектів,

які виконувалися за період 2012-2017 рр. за різноманітними замовленнями та які розрізнялися значеннями характеристик. Крім того, автори [10] спиралися на результати визначення таксономії RFC, яку було розроблено у [11]. Результатом дослідження [10] є скоригована модель змін, які виникають під час перегляду коду. Ця модель (з виключенням розгорнутих описів класів RFC) наведена у табл. 2.

Таблиця 2

Скоригована модель змін, які виникають під час перегляду коду

Artifact	Activity	Category	Topic	Detailed Change
1	2	3	4	5
Production & Test Code	Maintainability & Perfective Maintenance	Documentation (D)	- Textual Documentation	(D.1) - Naming. (D.2) - Comments. (D.3) - License Header. (D.4) - Typos. (D.5) - Other.
			- Language Supported Documentation	(D.6) - Immutability. (D.7) - Visibility.
		Style (S)	- Style	(S.1) – Brackets & Braces. (S.2) – Indentation. (S.3) – Blank Lines. (S.4) – Long Lines. (S.5) – Whitespace Usage. (S.6) – Grouping. (S.7) – Commented out code.
	Structure (STR)	- Re-implementation	(STR.1) – Semantic Duplication. (STR.2) – Semantic Dead Code. (STR.3) – Change Function. (STR.4) – Standard Coding Conventions. (STR.5) – New Functionality. (STR.6) – Strings (Wording). (STR.7) – Logging. (STR.8) – Testing.	
	Maintainability & Perfective Maintenance	Structure (STR)	- Organization	(STR.9) – Imports. (STR.10) – Move Functionality. (STR.11) – Long Sub Routine. (STR.12) – Dead Code. (STR.13) – Duplication / Redundant Code. (STR.14) – Complex Code/Simplification. (STR.15) – Statement Issue. (STR.16) – Consistency. (STR.17) – Architectural Changes.
	Functionality / Corrective Maintenance	Interface (I)	- Interface	(I.1) – Function Call. (I.2) – Parameter.

Кінець таблиці 2

1	2	3	4	5
Production & Test Code	Functionality / Corrective Maintenance	Logic (L)	- Logic	(L.1) – Compare. (L.2) – Computation. (L.3) – Wrong Location. (L.4) – Algorithm / Performance.
		Resource (R)	- Resource	(R.1) – Variable Initialization. (R.2) – Memory Management. (R.3) – Data & Resource Manipulation. (R.4) – Security. (R.5) – Concurrency.
		Check (C)	- Check	(C.1) – Check Function. (C.2) – Check Variable. (C.3) – Check User Input.
		Larger Defects (LD)	- Larger Defects	(LD.1) – Completeness. (LD.2) – GUI. (LD.3) – Check outside code / Domino Effects.
Other Changes	–	–	–	(O.1) – Commit Message. (O.2) – Continuous Integration / Continuous Delivery configurations. (O.3) – Automated Static Analysis Tools configurations. (O.4) – Language and Framework specific. (O.5) – External Software Documentation. (O.6) – Runtime Configurations. (O.7) – Other.

Результати порівняльного аналізу розглянутих варіантів класифікацій RFC дозволяють зробити такі висновки:

а) єдиної сталої системи класифікації RFC, яка охоплювала б увесь ЖЦІТ-проєкту, не існує;

б) кожен різновид ІТ-проєкту може мати власну множину типів RFC, яка, в свою чергу, може розглядатися як підмножина загальних моделей класифікації таких RFC;

в) в будь-якому випадку загальний поділ RFC на окремі підмножини виділяє функціональні і технічні RFC;

г) кількість класів RFC може сильно залежати від таких характеристик ІТ-проєкту:

- розмір ІТ-проєкту;
- час, який відведено на виконання ІТ-проєкту;
- необхідна швидкість реагування на потреби у змінах;
- рівень кваліфікації учасників ІТ-проєкту, які мають право визначати потреби у змінах та безпосередньо формувати RFC;
- кількість очікуваних або вже наявних RFC, які вимагають класифікації та ранжування.

Проведений аналіз існуючих класифікацій RFC показав, зокрема, що спроби вирішення задачі класифікації цих запитів приводить до створення складних моделей

класифікації, які є надлишковими з точки зору окремих ІТ-проектів. Цю надлишковість слід вважати проявою складного характеру управління змінами. Тому виникає проблема надання гнучкості існуючим процесам, підпроцесам і роботам за умови, що ця гнучкість допоможе подолати складності управління змінами. Ця проблема вважається одним з найважливіших викликів для такого напрямку робіт, як автоматизація управління змінами [12]. Для подолання цього виклику у [12] запропоновано розглядати автоматизовані системи управління змінами у проектах як різновид систем підтримки гнучкого прийняття рішень, що є одним з напрямів досліджень в галузі штучного інтелекту. Окремі сучасні дослідження [12], [13] вважають, що методи штучного інтелекту здатні не тільки підтримувати окремі роботи, а й керувати комплексом процесів актуального фреймворку управління змінами. Тому виникає необхідність проведення дослідження можливості застосування артефактів штучного інтелекту у елементах актуального фреймворку управління змінами.

Для проведення такого дослідження у [12] було запропоновано розглядати процеси, підпроцеси і роботи актуального фреймворку управління змінами як систему, яка повинна вдовольнити визначені у [14] загальні бізнес-потреби:

- а) бізнес-потреба 1 – автоматизація (як основний спосіб подолання збільшення складності та зростання потужностей, необхідних для втілення управління змінами в систему управління ІТ-проектом);
- б) бізнес-потреба 2 – підтримка прийняття рішень;
- в) бізнес-потреба 3 – оптимізація;
- г) бізнес-потреба 4 – контроль та нагляд.

Для вдоволення цих бізнес- потреб у [12] було досліджено можливості застосування таких артефактів штучного інтелекту:

- алгоритми машинного навчання;
- методи штучного інтелекту, спрямовані на комунікацію, сприйняття та дію (communicating, perceiving and acting AI-methods);
- алгоритми вирішення проблем;
- алгоритми здобування знань, міркування та планування;
- підходи до обробки невизначених знань та міркувань, зокрема, ймовірнісне представлення знань, байєсівські мережі, нечітка логіка, мультиагентні системи.

Крім цих артефактів, в [12] також були розглянуті такі прогнози щодо можливих напрямів розвитку артефактів штучного інтелекту в сфері управління змінами:

- дослідження застосування штучного інтелекту в управлінні змінами, зосереджені в основному на етапах попереднього затвердження RFC та після впровадження цих запитів;
- дослідження мета-евристичних алгоритмів та алгоритм машинного навчання на основі популяції як універсальних алгоритмів штучного інтелекту для оптимізації та автоматизації завдань, що виникають під час управління змінами;
- дослідження мультиагентних систем, потрібні для повної автоматизації процесів управління змінами, оскільки потоки даних у фреймворку управління змінами надто децентралізовані, а процеси управління змінами надто складні;
- дослідження особливостей виконання більшості тестів алгоритмів штучного інтелекту на даних реального світу (тобто окремих проектів) внаслідок відсутності стандартного набору даних для тестування нових алгоритмів в галузі управління змінами;
- розширення впровадження результатів досліджень методів і моделей штучного інтелекту в галузі управління змінами;
- процеси управління знаннями керуються експертними знаннями та дуже індивідуальні, що вимагає значної обробки даних;

– дані, які виникають у фреймворку управління змінами, здебільшого розпливчасті, а результати процесів часто невизначені, що вимагає використання нечіткої логіки та ймовірнісних методів;

– дослідження управління змінами на основі штучного інтелекту рідко порівнюють свій підхід із уже існуючими рішеннями, демонструючи загальну відсутність строгості.

Результати дослідження можливості застосування артефактів штучного інтелекту для задоволення бізнес-потреб управління змінами представлено у табл. 3 [12].

Таблиця 3

Результати дослідження можливості застосування артефактів штучного інтелекту для задоволення бізнес-потреб управління змінами

Бізнес-потреба	Галузь дослідження штучного інтелекту	Можливість застосування знань з інших досліджень штучного інтелекту
Бізнес-потреба 1	Моделювання і розробка мультиагентних систем	Так
Бізнес-потреба 2	Підтримка прийняття рішень шляхом використання об'єктивних даних, машинне навчання без вчителя для автоматичної кластеризації змін у схожі пакети	Так
Бізнес-потреба 3	Мета-евристичний підхід для визначення оптимального терміну набрання чинності	Так
Бізнес-потреба 4	Постійне прогнозування результату виконання елементів фреймворку управління змінами	Ні
	Валідація в польових умовах на прикладі конкретного ІТ-проєкту та порівняння різних підходів	Так

Базуючись на наведених у табл. 3 результатах, можна зробити такі висновки:

а) вибір методів збору та класифікації потреб користувачів ІТ-продукту як інструментальних засобів робіт створення та ранжування RFC як елементів актуального фреймворку управління змінами бажано розглядати як задачу підтримки прийняття рішень на основі об'єктивних даних;

б) рекомендацію найвигіднішого методу збору та класифікації потреб користувачів ІТ-продукту із запропонованих альтернатив бажано розглядати як результат застосування одного із засобів реалізації мета-евристичного підходу;

в) перевірку рекомендованого методу бажано проводити в польових умовах на прикладі конкретного ІТ-проєкту;

г) бажано розглядати рішення задачі вибору методів збору та класифікації потреб користувачів ІТ-продукту як інструментальних засобів робіт створення та ранжування RFC як елементів актуального фреймворку управління змінами у вигляді елементів мультиагентної системи.

Результати проведеного аналізу дозволяють стверджувати, що роботу «Визначте/уточніть потребу в змінах» слід визнати практично недослідженою роботою, виконання якої визначає усі особливості виконання подальших робіт актуального фреймворку управління змінами. Тому проведення досліджень, які дадуть змогу виробити рекомендації з

виконання цієї роботи у конкретних ІТ-проектах та узагальнити ці рекомендації для окремих різновидів ІТ-проектів, слід вважати теоретично та практично актуальним.

3. Мета і задачі дослідження

Метою даного дослідження є розробка методу вирішення задачі формування і класифікації RFC ІТ-продукту. Досягнення цієї мети дозволить автоматизувати вирішення задачі формування і класифікації RFC ІТ-продукту, що значно спростить управління змінами у різних ІТ-проектах однієї й тієї ж ІТ-компанії.

Для досягнення цієї мети треба вирішити такі задачі:

- розробка моделі RFC ІТ-продукту;
- розробка методу вирішення задачі формування і класифікації RFC ІТ-продукту.

4. Матеріали і методи дослідження

Об'єктом дослідження є робота «Визначте/уточніть потребу в змінах» підпроцесу «Формулювання змін» процесу «Управління змінами» [1]. Предметом дослідження є методи збору та класифікації потреб користувачів ІТ-продукту.

Головною особливістю дослідження слід вважати відсутність формального представлення моделей класифікації RFC. Тому як основну гіпотезу даного дослідження запропоновано розглядати модель RFC та метод вирішення задачі формування і класифікації запитів на зміну ІТ-продукту як результати узагальнення існуючих ІТ, які використовувалися для перетворення описів потреб учасників ІТ-проекту у змінах на RFC та подальшої класифікації отриманих RFC.

Результати порівняльного аналізу розглянутих вище варіантів класифікацій RFC дозволяють встановити два основні різновиди таких ІТ.

Перший різновид має в своїй основі апріорно визначену керівництвом ІТ-проекту модель, яка декларативно описує множину припустимих класів RFC. Приклад такої ІТ наведено у [5]. Ця апріорно визначена модель використовується в ІТ для вирішення задачі класифікації, яка встановлює належність кожної окремої потреби у зміні до конкретного класу RFC та дає змогу перетворити опис цієї потреби у публікацію відповідного RFC встановленого класу.

В загальному випадку цей перший різновид ІТ можна представити як послідовність таких кроків.

Крок 1. Учасником ІТ-проекту формується та публікується потреба у зміні ІТ-продукту (у вигляді тексту).

Крок 2. Формується дерево рішень на основі моделі класифікації RFC, яку було апріорно обрано керівництвом ІТ-проекту.

Крок 3. Для публікації потреби у зміні вирішується задача класифікації з використанням дерева рішень, сформованого на Кроці 2.

Крок 4. На основі класифікованої публікації потреби у зміні формується публікація RFC, клас якого визначено як результат виконання Кроку 3. Завершення роботи технології.

Діаграму потоків даних ІТ на основі апріорно визначеної моделі, яка декларативно описує множину припустимих класів RFC, наведено на рис. 7. Для побудови цієї та наступних діаграм потоків даних використано нотацію, яку було запропоновано у онлайн-CASE-засобі Draw.io v. 2.0.4.

Тут і надалі термін «публікація» пропонується застосовувати за аналогією із запропонованим у [15] терміном «публікація вимог до інформаційної системи». Така аналогія викликана загальним представленням RFC як різновидів вимог, які висуваються до

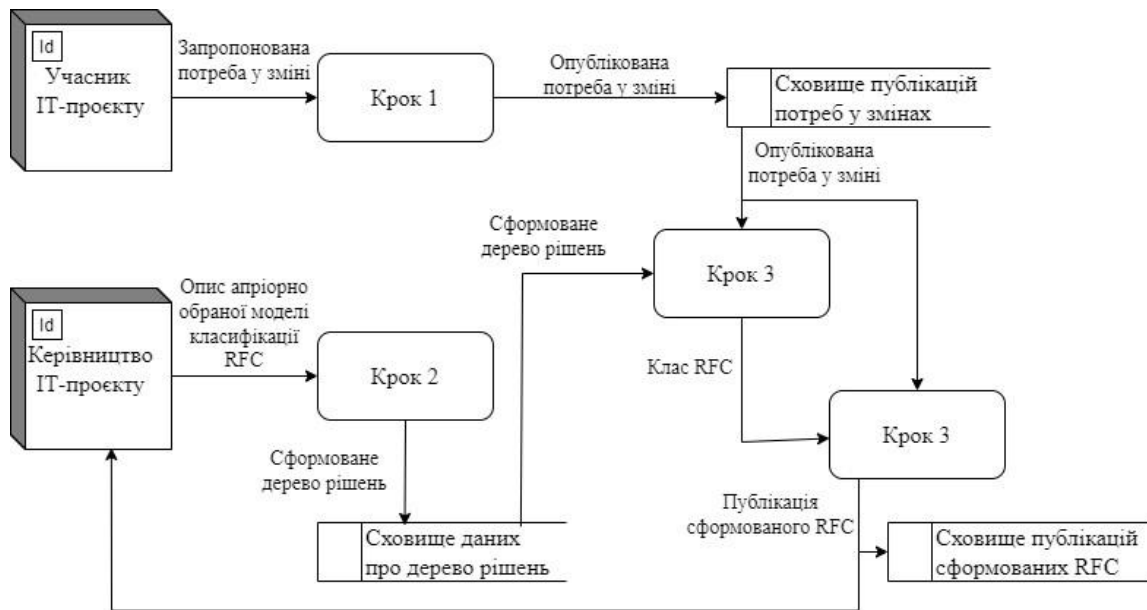


Рис. 7. Діаграма потоків даних інформаційної технології на основі апріорно визначеної моделі, яка декларативно описує множину припустимих класів запитів на зміни

ІТ-продуктів протягом ЖЦ існування цих продуктів.

Під терміном «публікація вимоги до інформаційної системи» слід розуміти [15]:

а) опис умови або можливості, які необхідні Споживачу ІТ-послуг для вирішення проблеми або досягнення мети, виконаний одним з припустимих в рамках методології розробки інформаційної системи способів;

б) опис, виконаний одним з припустимих в рамках методології розробки інформаційної системи способів, умови або можливості, якими повинна володіти інформаційна система або компонент інформаційної системи (ІТ-послуга, ІТ-сервіс) з точки зору Постачальника або Споживача ІТ-послуг та які відповідають договору, стандарту, специфікації або іншому офіційному документу,.

Виходячи з цього визначення, під терміном «публікація потреби у зміні» слід розуміти опис умови або можливості, які необхідні Споживачу ІТ-послуг для вирішення проблеми або досягнення мети, які не були визначені у поточній версії ІТ-продукту, виконаний у вигляді неструктурованого або слабо структурованого тексту. Під терміном «публікація сформованого RFC» слід розуміти:

а) опис, виконаний одним з припустимих в рамках методології розробки інформаційної системи способів, умови або можливості, які необхідні Споживачу ІТ-послуг для вирішення проблеми або досягнення мети та які не були визначені у поточній версії ІТ-продукту,;

б) опис, виконаний одним з припустимих в рамках методології розробки ІТ-продукту способів, умови або можливості, якими повинна володіти наступна версія ІТ-продукту з точки зору Постачальника або Споживача ІТ-послуг та які відповідають договору, стандарту, специфікації або іншому офіційному документу.

Визначення термінів «Споживач ІТ-послуг» та «Постачальник ІТ-послуг» наведено у [15].

Другий різновид ІТ автоматизованого виконання роботи «Визначити/уточнити потреби у змінах» побудовано з врахуванням можливості відсутності під час виконання цієї роботи апріорно визначеної керівництвом ІТ-проєкту існуючої моделі класифікації RFC. Приклад такої технології розглянуто у [9]. В загальному випадку цей другий різновид ІТ можна представити як послідовність таких кроків.

Крок 1. Формування та публікація учасником ІТ-проєкту потреби у зміні ІТ-продукту (у вигляді тексту).

Крок 2. Попереднє розподілення опублікованих потреб у змінах ІТ-продукту на класи функціональних та технічних потреб в змінах.

Крок 3. Для кожного з класів потреб в змінах, визначених на Кроці 2, вирішення задачі виділення множин ключових слів, які характеризують кожну конкретну публікацію потреби в зміні ІТ-продукту.

Крок 4. Для кожного з класів потреб в змінах, визначених на Кроці 2, вирішення задачі кластеризації на основі множин ключових слів, сформованих на Кроці 3.

Крок 5. Формування та публікація RFC з врахуванням результатів виконання Кроку 2 і Кроку 4. Завершення роботи технології.

Діаграму потоків даних ІТ з відсутньою апріорно визначеною моделлю, яка декларативно описує множину припустимих класів RFC, наведено на рис. 8.

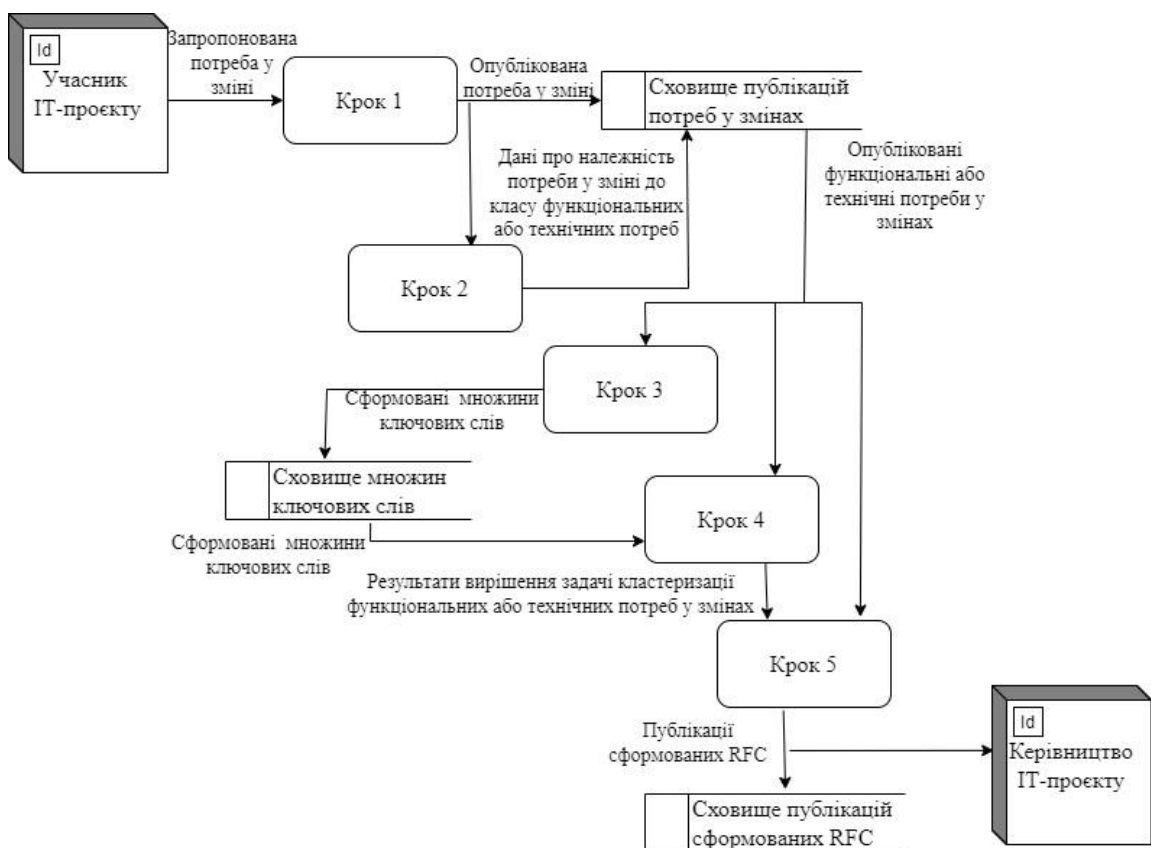


Рис. 8. Діаграма потоків даних інформаційної технології з відсутньою апріорно визначеною моделлю, яка декларативно описує множину припустимих класів запитів на зміни

Подібність розглянутих технологій дозволяє висунути припущення, що ці різновиди є частковими випадками реалізації загального методу вирішення задачі формування і класифікації RFC.

5. Результати дослідження

5.1. Розробка моделі запиту на зміну ІТ-продукту

Першим припущенням, на якому базується модель RFC ІТ-продукту, є припущення про формування і публікацію будь-якої потреби у зміні ІТ-продукту у вигляді

неструктурованого чи слабо структурованого тексту від якогось стейкхолдера ІТ-проекту. Цим стейкхолдером може бути:

- майбутній оператор чи користувач ІТ-продукту;
- керівництво організації, яка виступає в ролі Споживача ІТ-послуг;
- один чи кілька виконавців ІТ-проекту, очікуваним результатом якого є ІТ-продукт;
- керівництво ІТ-компанії, яка виступає в ролі Постачальника ІТ-послуг;
- представник фізичної чи юридичної особи, яка виступає в ролі спонсора ІТ-проекту;
- будь-яка фізична особа чи представник юридичної особи, які приймають на себе одну з ролей в ІТ-проекті, очікуваним результатом якого є ІТ-продукт.

Це припущення дозволяє формально описати будь-яку потребу у зміні, що виникає під час ЖЦІ ІТ-проекту, очікуваним результатом якого є ІТ-продукт, як кортеж

$$N_C = \langle Id_S, Text_C \rangle, \quad (1)$$

де N_C – потреба у зміні (need for change); Id_S – ідентифікатор ролі та/або особи (стейкхолдера), яка виступає як джерело потреби у зміні; $Text_C$ – текстовий опис потреби у зміні.

Розглянуті вище моделі класифікації RFC є частковими випадками деревоподібної структури, корінь якої визначає належність RFC (потреби у зміні), на основі якої формується RFC, до одного з двох базових класів – функціональних або технічних RFC (потреб у зміні). Тому друге припущення, на якому базується модель RFC ІТ-продукту, полягає у можливості формального представлення будь-якої моделі класифікації RFC, яку керівництво ІТ-проекту може апіорно визначити для застосування під час вирішення задачі формування і класифікації RFC, як деревоподібного графу

$$G_{Cl_{RFC}} = (Attr, V), \quad (2)$$

де $G_{Cl_{RFC}}$ – деревоподібний граф, який формально описує моделі класифікації RFC; $Attr$ – множина вершин деревоподібного графу $G_{Cl_{RFC}}$, яка одночасно є множиною атрибутів, що описують ознаки класифікації RFC (потреби у зміні), яка має вигляд

$$Attr = (attr_1, attr_1, attr_2, \dots, attr_i, \dots, attr_z), \quad (3)$$

де $attr_i$ – атрибут, значення якого визначає належність RFC (потреби у зміні) до одного з двох базових класів – функціональних або технічних RFC (потреб у зміні), та який є коренем деревоподібного графу $G_{Cl_{RFC}}$; $attr_i$ – і-й атрибут, що є вузлом розгалуження або листом деревоподібного графу $G_{Cl_{RFC}}$, значення якого визначає належність RFC (потреби у зміні) до окремої підмножини класів (вузол розгалуження) або конкретного класу (лист) RFC (потреби у зміні); z – кількість вершин деревоподібного графу, яка визначається як кількість атрибутів у апіорно обраній моделі класифікації RFC, обраної керівництвом ІТ-проекту; V – множина ребер, які зв'язують окремі пари вершин $(attr_i, attr_j)$ деревоподібного графу $G_{Cl_{RFC}}$ та ідентифікуються окремими значеннями атрибуту $attr_i$, який є початком відповідного ребра.

В результаті вирішення задачі класифікації встановлюється клас RFC (потреби у зміні). Цей клас може бути представлений як послідовність значень атрибутів класифікації, які описують шлях деревоподібним графом (2) від кореневої вершини до одного з листів графа. Але слід врахувати такі умови, які накладаються на визначення цього шляху:

- а) апіорно визначена модель класифікації у вигляді деревоподібного графа (2) в

конкретному ІТ-проекті може бути відсутня;

б) апіорно визначена модель класифікації у вигляді деревоподібного графа (2) може доповнюватися новими вершинами та ребрами, які визначаються за результатами вирішення задачі кластеризації публікацій потреб у змінах ІТ-продукту;

в) апіорно визначена модель класифікації у вигляді деревоподібного графа (2) може створюватися в результаті багатократного вирішення задачі кластеризації публікацій потреб у змінах ІТ-продукту.

Врахування цих умов дозволяє описати будь-який i -й клас RFC (потреби у зміні) як кортеж з двох підмножин, що має вигляд

$$C_{iRFC} = (C_{iPt}, C_{igr}), \quad (4)$$

де C_{iRFC} – i -й клас RFC (потреби у зміні); C_{iPt} – підмножина значень атрибутів апіорно визначеної керівництвом ІТ-проекту моделі класифікації RFC, що має вигляд

$$C_{iPt} = \begin{cases} \langle v_t, v_a, \dots, v_b \rangle \text{ у випадку застосування моделі класифікації;} \\ \langle v_t \rangle \text{ в іншому випадку.} \end{cases}, \quad (5)$$

де $\langle v_t, v_a, \dots, v_b \rangle$ – кортеж ребер, які описують шлях від кореневої вершини до b -го листа графу (2); $\langle v_t \rangle$ – кортеж, який вказує на належність RFC (потреби у зміні) до класу функціональних або технічних RFC (потреб у змінах); C_{igr} – підмножина значень додаткових атрибутів, встановлених у результаті вирішення задачі кластеризації, що має вигляд

$$C_{igr} = \begin{cases} \langle v_{b1}, \dots, v_{bk} \rangle \text{ у випадку вирішення задачі кластеризації;} \\ \langle \emptyset \rangle \text{ в іншому випадку.} \end{cases}, \quad (6)$$

де $\langle v_{b1}, \dots, v_{bn} \rangle$ – кортеж значень атрибутів, які використовуються для опису окремих кластерів, що виникають в результаті вирішення задачі кластеризації публікацій потреб у змінах.

Для забезпечення можливості вирішення задачі кластеризації з метою уточнення існуючої моделі класифікації потреб у змінах введемо множину Tez_{ITP} ключових слів, які описують ІТ-продукт, його елементи, їхні характеристики та властивості. При цьому кожна публікація потреби у змінах буде характеризуватися множиною ключових слів, які однозначно визначають цю публікацію, що має вигляд

$$\begin{bmatrix} kw_1 \\ \dots \\ kw_j \\ \dots \\ kw_p \end{bmatrix} \in Tez_{ITP}, \quad (7)$$

де kw_j – j -те ключове слово, яке характеризує публікацію потреби у зміні, $j=1, \dots, p$.

Розглянуті особливості вирішення задачі формування і класифікації RFC дозволяють представити модель i -го RFC ІТ-продукту M_{RFC_i} як кортеж, що має вигляд:

$$M_{RFC_i} = \left\langle Id_{S_i}, G_{Cl_{RFC}}, \langle C_{ij_{Pt}}, C_{ij_{gr}} \rangle, Tez_{ITP}, \begin{bmatrix} kw_{i1} \\ \dots \\ kw_{ij} \\ \dots \\ kw_{ip} \end{bmatrix}, Text_{RFC_i} \right\rangle, \quad (8)$$

де Id_{S_i} – ідентифікатор джерела i -го RFC (потреби у зміні); $C_{ij_{Pt}}$ – підмножина значень атрибутів апріорно визначеної керівництвом ІТ-проєкту моделі класифікації, які визначають клас C_j для i -го RFC (потреби у зміні), $j=1, \dots, m$; $C_{ij_{gr}}$ – підмножина значень додаткових атрибутів, встановлених у результаті вирішення задачі кластеризації, які визначають клас C_j для i -го RFC (потреби у зміні), $j=1, \dots, m$; m – кількість класів моделі класифікації M_{RFC_i} ; $Text_{RFC_i}$ – публікація i -го RFC (потреби у зміні).

Модель (8) дозволяє розробити загальний метод вирішення задачі формування і класифікації RFC як основний інструмент формування і публікації потреб у змінах ІТ-продукту.

5.2. Розробка методу вирішення задачі формування і класифікації запитів на зміну ІТ-продукту

Розроблена модель (8) дозволяє деталізувати опис змісту етапів загального методу вирішення задачі формування і класифікації RFC. З використанням моделі (8) цей метод може бути представлений як послідовність таких етапів і кроків.

Етап 1. Визначення похідної моделі класифікації RFC, яку слід застосовувати для формування та класифікації потреб у змінах ІТ-продукту в конкретному ІТ-проєкті.

Крок 1.1. Вибір керівництвом ІТ-проєкту похідної моделі класифікації RFC $G_{Cl_{RFC}}$ на основі досвіду виконання попередніх ІТ-проєктів та аналізу результатів науково-прикладних досліджень.

Крок 1.2. Якщо в результаті виконання Кроку 1.1 обрано конкретну модель класифікації RFC $G_{Cl_{RFC}}$, то визначення підмножини атрибутів $Attr_{i_{Pt}}$ та їхніх значень, які утворюють постійну основу дерева рішень з класифікації RFC (потреби у змінах) ІТ-продукту в межах конкретного ІТ-проєкту.

Крок 1.3. Якщо в результаті виконання Кроку 1.1 жодну з існуючих моделей класифікації RFC $G_{Cl_{RFC}}$ не було обрано, то визначення підмножини атрибутів $Attr_{i_{Pt}}$, яка складається з атрибуту $attr_i$, та значень цього атрибуту, які визначають належність об'єкту класифікації до функціональних чи технічних RFC (потреб у змінах).

Крок 1.4. Визначення порогового значення ε_{RFC} , перевищення якого визначає необхідність уточнення моделі класифікації RFC (потреб у змінах) ІТ-продукту у конкретному ІТ-проєкті.

Етап 2. Формування та попередньої класифікації потреб у змінах ІТ-продукту.

Крок 2.1. Формування зацікавленим стейкхолдером ІТ-проєкту потреби в зміні ІТ-продукту N_C за моделлю (1).

Крок 2.2. У випадку, якщо підмножина $Attr_{i_{Pt}}$ атрибутів похідної моделі класифікації RFC складається з одного атрибуту $attr_i$, визначення стейкхолдером ІТ-проєкту типу RFC, якому відповідає сформована потреба у зміні ІТ-продукту.

Крок 2.3. У випадку, якщо підмножина $Attr_{i_{Pt}}$ атрибутів похідної моделі класифікації

RFC складається з двох чи більшої кількості атрибутів, визначення стейкхолдером ІТ-проєкту переліку значень цих атрибутів $\langle v_t, v_a, \dots, v_b \rangle$, які визначають клас сформованої потреби у зміні ІТ-продукту.

Крок 2.4. Публікація сформованої потреби у зміні ІТ-продукту стейкхолдером ІТ-проєкту.

Етап 3. Уточнення та адаптація похідної моделі класифікації RFC до особливостей ІТ-продукту та ІТ-проєкту.

Крок 3.1. Визначити кількість публікацій Q_{N_j} потреб у змінах ІТ-продукту в кожному з класів $C_j, j=1, \dots, m$, виявлених за результатами виконання Кроків 2.2 та 2.3.

Крок 3.2. Якщо хоча б для одного з класів C_j виконується умова

$$Q_{N_j} > \varepsilon_{RFC}, \quad (9)$$

то перейти до Кроку 3.3. В іншому випадку перейти до Кроку 3.9.

Крок 3.3. З кожної публікації потреби в зміні, яку було віднесено до класу C_j , виділити

множину ключових слів $\begin{bmatrix} kw_1 \\ \dots \\ kw_j \\ \dots \\ kw_p \end{bmatrix} \in Tez_{ITP}$, які однозначно характеризують дану публікацію.

Крок 3.4. Сформувати з множин ключових слів $\begin{bmatrix} kw_1 \\ \dots \\ kw_j \\ \dots \\ kw_p \end{bmatrix}$ публікацій потреб у змінах, які

віднесено до класу, вибірку похідних даних для вирішення задачі кластеризації.

Крок 3.5. Вирішити задачу кластеризації для вибірки, сформованої на Кроці 3.4, з додатковим обмеженням на кількість елементів в кожному з кластерів, яка не повинна перевищувати значення ε_{RFC} .

Крок 3.6. Запропонувати множину кластерів, сформованих в результаті виконання Кроку 3.5, керівництву ІТ-проєкту як додаткові підкласи $C_{ij_{gr}}$, сформовані на основі публікацій потреб у змінах, віднесених до класу.

Крок 3.7. Якщо керівництво ІТ-проєкту приймає рішення про доцільність багаторазового використання запропонованих додаткових підкласів, сформувати підмножину атрибутів $Attr_{j_{gr}}$, елементи якої однозначно ідентифікують кожен з цих підкласів.

Крок 3.8. Адаптувати похідну модель класифікації RFC $G_{C_{RFC}}$ до нових особливостей ІТ-продукту та ІТ-проєкту, виконавши операцію

$$Attr_{i_{Pt}} = Attr_{i_{Pt}} \cup Attr_{i_{gr}}, \quad (10)$$

після чого повернутися на Крок 3.1.

Крок 3.9. Сформувати підмножини публікацій потреб у змінах ІТ-продукту, віднесених до виділених класів $C_j, j=1, \dots, m$, RFC.

Етап 4. Публікації та розсилання RFC стейкхолдерам ІТ-проєкту.

Крок 4.1. Перетворення публікацій потреб в змінах ІТ-продукту на публікації RFC за моделлю (8).

Крок 4.2. Визначення переліку стейкхолдерів ІТ-проєкту, яким слід надіслати відповідні публікації RFC, отримані в результаті виконання Кроку 4.1.

Крок 4.3. Розсилання публікацій класифікованих RFC множині стейкхолдерів ІТ-проєкту для наступних робіт з пріоритетизації та оцінювання цих RFC. Завершення застосування методу.

6. Обговорення результатів дослідження

Розроблена модель (8) i -го RFC ІТ-продукту дозволяє описати множину функціональних і технічних RFC, що виникають під час розробки ІТ-продукту. На відміну від існуючих моделей, які зосереджено на описі результатів класифікації вже сформульованих RFC, ця модель описує RFC як результат виконання послідовності робіт з похідного формулювання потреби у зміні та поступового перетворення цієї потреби на RFC, який віднесено до відповідного класу. Ця відмінність дозволяє забезпечити покращення робіт з управління змінами, зокрема, з трасування джерел потреб, сформульованих потреб у змінах, публікацій RFC та зв'язків між ними.

Розроблений метод вирішення задачі формування і класифікації RFC базується на моделі (8) і дозволяє уніфікувати вирішення цієї задачі незалежно від особливостей обраної моделі класифікації, ІТ-проєкту та ІТ-продукту, який розробляється. При цьому, на відміну від існуючих, метод дозволяє не тільки застосовувати апріорно визначену модель класифікації RFC, а й адаптувати існуючу модель до нових особливостей ІТ-проєкту та ІТ-продукту та розробляти власну модель класифікації RFC, що якнайкраще підходить до особливостей діяльності ІТ-компанії, її ІТ-проєктів та ІТ-продуктів.

Використання запропонованих у дослідженні моделі (8) та методу вирішення задачі формування і класифікації RFC дає змогу автоматизувати виконання роботи «Визначте/уточніть потребу в змінах» підпроцесу «Формулювання змін». При цьому застосування розробленого методу дозволяє, на відміну від запропонованих у [5, 9] ІТ, зробити рішення з автоматизації достатньо близьким до універсального, яке не буде вимагати значних змін внаслідок можливої зміни сценарію вирішення задачі формування і класифікації RFC.

Головним недоліком отриманих результатів дослідження слід вважати їхню орієнтацію виключно на RFC, які виникають тільки під час виконання ІТ-проєкту з розробки чи модифікації ІТ-продукту. Це обмежує можливість застосування отриманих результатів і вимагає проведення додаткових досліджень формальних описів RFC, які можуть виникати на інших стадіях ЖЦ ІТ-проєкту та ІТ-продукту.

Виходячи з цього недоліку, головним напрямом подальших досліджень в галузі управління формулюванням змін слід визнати дослідження можливості уніфікації формальних описів RFC на різних стадіях ЖЦ ІТ-проєкту та ІТ-продукту. Підтвердження цієї можливості встановлює, у свою чергу, можливість розробки уніфікованих методів та інструментальних засобів управління змінами в межах відповідних ЖЦ. Спростування цієї можливості дозволить визначити якісні та кількісні ознаки ІТ-проєктів та ІТ-продуктів, наявність яких буде означати необхідність зміни методів та інструментів управління змінами.

Ще одним напрямом подальших досліджень у цій галузі є дослідження особливостей

методів збирання вимог, які слід застосовувати для збирання потреб у змінах ІТ-продукту. При цьому слід виходити з припущення, що потреби у змінах та RFC, створені на їх основі, розглядаються керівництвом ІТ-проєкту як вимоги до ІТ-продукту, що виникають після завершення стадії планування ІТ-проєкту і до моменту початку ініціації нового ІТ-проєкту, який буде стосуватися цього ж ІТ-продукту.

7. Висновки

У ході даного дослідження було вирішено задачу розробки методу вирішення задачі формування і класифікації запитів на зміну ІТ-продукту. Під час вирішення цієї задачі було здійснено:

- розробку моделі (8) *i*-го RFC ІТ-продукту, яка на формальному рівні дозволяє описати будь-які функціональні та технічні RFC, що виникають під час виконання ІТ-проєкту з розробки чи модифікації ІТ-продукту;
- розробку методу вирішення задачі формування і класифікації RFC, який дозволяє уніфікувати вирішення цієї задачі.

Отримані результати дозволяють в подальшому вирішити задачу розробки уніфікованих засобів автоматизованого вирішення задачі формування і класифікації RFC у вигляді елементів мультиагентної чи сервіс-орієнтованої інформаційної системи управління ЖЦ ІТ-продукту.

Перелік посилань:

1. Настанова до Зводу знань з управління проєктами. Настанова РМВОК. Сьоме видання. Стандарт з управління проєктами. Project Management Institute, Inc., 14 Campus Boulevard Newtown Square, Pennsylvania 19073-3299 USA, 2021. 370 с.
2. Managing change in organizations: a practice guide. Project Management Institute, Inc., 14 Campus Boulevard Newtown Square, Pennsylvania 19073-3299 USA. 2013. 122 p.
3. ISO/IEC/IEEE 12207:2017. Systems and software engineering – Software life cycle processes. Чинний від 2017-11-01. 150 p.
4. Віжан К. Процес управління змінами в ІТ-компанії, або Як побороти страх роботи з чендж реквестами. *DOU*. URL: <https://dou.ua/forums/topic/42501/> (дата звернення 27.03.2024).
5. Riesener M, Dölle C, Mendl-Heinisch M, Schuh G, Keuper A. Derivation of Description Features for Engineering Change Request by Aid of Latent Dirichlet Allocation. *Proceedings of the Design Society: DESIGN Conference*. 2020. Vol. 1. P. 697-706. doi:10.1017/dsd.2020.98.
6. Jarratt T.A.W. et al. Engineering change. An overview and perspective on the literature. *Research in Engineering Design*. 2011. Vol. 22, No. 2. P. 103-124. <https://doi.org/10.1007/s00163-010-0097-y>
7. Feldhusen J., Grote K.-H. Pahl/Beitz Konstruktionslehre: Methoden und Anwendung erfolgreicher Produktentwicklung. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. <https://doi.org/10.1007/978-3-642-29569-0>
8. Sakhravi Z., Sellami A., Bouassida N. Requirements Change Requests Classification: An Ontology-Based Approach. In: Abraham A., Siarry P., Ma K., Kaklauskas A. (eds) *Intelligent Systems Design and Applications. ISDA 2019. Advances in Intelligent Systems and Computing*, Vol. 1181. Springer, Cham. 2021. P. 487-496. https://doi.org/10.1007/978-3-030-49342-4_47
9. Zhou Y., Su Y., Chen T., Huang Z., Gall H.C., Panichella S. User Review-Based Change File Localization for Mobile Applications. *IEEE Transactions on Software Engineering*. 2019. Vol. 47 (12). P. 2755-2770.
10. Panichella S., Zaugg N. An Empirical Investigation of Relevant Changes and Automation Needs in Modern Code Review. *Empiric Software Engineering*. 2020. Vol. 25. P. 4833-4872. <https://doi.org/10.1007/s10664-020-09870-3>.
11. Beller M, Bacchelli A, Zaidman A, Jurgens E. Modern code reviews in open-source projects: which problems do they fix? *11Th working conference on mining software repositories, MSR 2014, proceedings*, 2014, Hyderabad, India. P. 202-211. <https://doi.org/10.1145/2597073.2597082>
12. Burggräf P., Wagner J., Saßmannshausen T. et al. AI-artifacts in engineering change management – a systematic literature review. *Research in Engineering Design*. 2024. Vol. 35. P. 215-237. <https://doi.org/10.1007/s00163-023-00430-6>.
13. Burggräf P, Wagner J, Saßmannshausen T.M. Sustainable Interaction of Human and Artificial Intelligence in Cyber Production Management Systems. In: Behrens B-A, Brosius A, Hintze W, Ihlenfeldt S, Wulfsberg JP (eds). *Production at the leading edge of technology*. WGP 2020. Lecture Notes in Production Engineering. Berlin, Heidelberg:

Springer, Berlin Heidelberg, 2021. P. 508–517.

14. Radisic-Aberger O., Weisser T., Saßmannshausen T., Wagner J., Burggräf P. Concept of a multi-agent system for optimized and automated engineering change implementation. *Proceedings of the Design Society*. 2022. Vol. 2. P. 1689–1698. <https://doi.org/10.1017/pds.2022.171/>

15. Левыкин, В.М., Евланов, М.В., Керносов, М.А. Паттерны проектирования требований: моделирование и применение: монография. Харьков: ООО «Компанія «Сміт»», 2014. 320 с.

Надійшла до редколегії 24.04.2024

Євланов Максим Вікторович, доктор технічних наук, професор, професор кафедри ІУС ХНУРЕ, м. Харків, Україна, e-mail: maksym.ievlanov@nure.ua, ORCID: <https://orcid.org/0000-0002-6703-5166> (науковий керівник здобувача вищої освіти Крука Б.Є.).

Юр'єв Іван Олексійович, кандидат технічних наук, доцент кафедри ІУС ХНУРЕ, м. Харків, Україна, e-mail: ivan.iuriev@nure.ua, ORCID: <https://orcid.org/0000-0002-5178-519X>.

Крук Богдан Євгенійович, здобувач вищої освіти, група УПГТМ-22-1, факультет комп'ютерних наук, ХНУРЕ, м. Харків, Україна, e-mail: bohdan.kruk@nure.ua.

УДК 004.8:004.9

DOI: 10.30837/0135-1710.2024.182.025

С.Ф. ЧАЛИЙ, В.О. ЛЕЩИНСЬКИЙ

ПОБУДОВА ПРОЦЕСНО-ОРІЄНТОВАНИХ ПОЯСНЕНЬ В ІНТЕЛЕКТУАЛЬНИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ НА ОСНОВІ МОЖЛИВІСНИХ КАУЗАЛЬНИХ ЗАЛЕЖНОСТЕЙ

Розглянуто процес побудови пояснень в інтелектуальних інформаційних системах. Запропоновано процесно-орієнтовану модель пояснення, яка інтегрує темпоральні та каузальні залежності процесу формування рішення. Модель забезпечує визначення послідовності станів процесу формування рішення на основі можливісних причинно-наслідкових зв'язків між діями такого процесу. Розроблено метод побудови процесно-орієнтованого пояснення на основі можливісних каузальних залежностей, який включає формування каузальних правил, оцінку можливості та необхідності цих правил, а також представлення пояснень з урахуванням атрибутів станів. Метод орієнтований на підтримку внутрішніх користувачів інтелектуальної системи.

1. Вступ

Інтелектуальні інформаційні системи (ІІС) сьогодні знаходять широке застосування у таких галузях, як охорона здоров'я, фінансові послуги, промисловість та управління бізнес-процесами. Ці системи здатні обробляти великі масиви даних, виявляти приховані закономірності та приймати рішення на основі алгоритмів машинного навчання [1]. Однак такі системи зазвичай залишаються «непрозорими» для користувачів внаслідок використання складних алгоритмів формування рішень, що може викликати недовіру до результатів роботи ІІС [2]. Відсутність розуміння того, як система дійшла до певного рішення, не лише знижує рівень довіри до її результатів, але й створює труднощі для виявлення та виправлення потенційних помилок [3]. Для вирішення цієї проблеми необхідно забезпечити користувачам доступ до зрозумілих пояснень щодо процесу прийняття рішень [4]. Такі пояснення повинні бути адаптованими до потреб різних категорій користувачів – як зовнішніх (кінцевих споживачів рішень), так і внутрішніх (розробників та аналітиків). Зовнішні користувачі зазвичай не мають глибоких технічних знань щодо внутрішньої структури ІІС і потребують обґрунтування рішень на концептуальному рівні. Внутрішні користувачі відповідають за налаштування та підтримку ІІС. Для них важливо мати детальне пояснення ключових етапів процесу формування рішень з метою оптимізації роботи системи та виявлення можливих «вузьких

місць» [5]. Процесний рівень деталізації дозволяє відслідковувати послідовність дій і станів у рамках формування рішення, а також визначити причинно-наслідкові зв'язки між цими станами. Тому процесний рівень пояснень дає можливість розробникам і аналітикам краще зрозуміти поведінку системи, виявити можливі помилки або неефективні дії та вдосконалити алгоритми прийняття рішень. Зазначене свідчить про актуальність проблеми побудови процесно-орієнтованих пояснень.

2. Аналіз літературних даних і постановка проблеми дослідження

Дослідження у сфері пояснювального штучного інтелекту (Explainable Artificial Intelligence – XAI) активно розвиваються протягом останніх років [4]. Основними напрямками досліджень є розробка методів побудови пояснень та створення ментальних моделей користувачів. Перший напрямок спрямований на забезпечення прозорості роботи інтелектуальних систем, а другий – на підтримку розуміння пояснень користувачами [6], [7]. При побудові пояснень значна увага приділяється розробці методів, які дозволяють генерувати пояснення на основі аналізу вхідних даних та властивостей моделі. Такі методи, як LIME [2], SHAP [8] та Grad-CAM [9], дозволяють визначити важливість окремих ознак або компонентів моделі для прийняття конкретного рішення. Інший важливий напрямок досліджень пов'язаний з побудовою пояснень з використанням темпоральних та каузальних залежностей. Каузальні моделі дозволяють описати причинно-наслідкові зв'язки між змінними та подіями в процесі прийняття рішень [10]-[13]. Використання каузальних залежностей дозволяє створювати інтерпретованіші та зрозуміліші пояснення, які відображають логіку роботи інтелектуальної системи [14]-[16]. Темпоральні залежності дозволяють описати послідовність станів та подій в процесі прийняття рішень. Особливості побудови темпоральних залежностей та подальшого використання цих залежностей для побудови процесу формування рішення розглядалися в [17]-[19]. Побудову пояснень на основі темпоральних залежностей запропоновано в [12]. Поєднання цих двох типів залежностей дає можливість сформулювати пояснення, які відображають логіку роботи інтелектуальної системи. У сфері інтелектуального аналізу процесів також розроблено ряд методів, які можуть бути застосовані для побудови пояснень. Такі методи дозволяють виявляти реальні процеси шляхом аналізу логів подій та будувати моделі цих процесів на основі визначення залежностей між подіями [20]. Побудова процесних моделей «як є» створює умови для інтерпретації послідовностей робіт цих процесів. Однак у наведених дослідженнях розглядаються як пояснення окремі ключові елементи процесу формування рішень в інтелектуальних системах. Представленню пояснення для внутрішніх користувачів шляхом опису процесу формування рішень в цілому на основі темпоральних та каузальних залежностей не приділяється достатньо уваги, що вказує на важливість задач, які вирішуються в даному дослідженні.

Таким чином, проблема дослідження пов'язана із невідповідністю можливостей сучасних підходів до побудови пояснень та потреб внутрішніх користувачів інтелектуальної системи. Сучасні підходи орієнтовані на побудову пояснення в першу чергу для зовнішніх користувачів, коли ІС представлена як «чорна скринька». Відповідно, пояснення відображають вплив значень вхідних та, частково, проміжних змінних на отримане рішення. Внутрішні користувачі взаємодіють з ІС як із «сірою скринькою», маючи доступ до інформації щодо проміжних станів інтелектуальної системи у процесі формування рішення. Тому вони мають отримати пояснення щодо ключових послідовностей дій в ІС, які привели до поточного рішення з тим, щоб забезпечити можливість удосконалення процесу формування рішень в інтелектуальній системі.

3. Мета і задачі дослідження

Метою даного дослідження є розробка процесно-орієнтованого підходу до побудови пояснень в інтелектуальних інформаційних системах для підтримки та удосконалення процесу формування рішення в інтелектуальних інформаційних системах.

Для досягнення поставленої мети у дослідженні вирішуються такі задачі: розробка

процесної моделі пояснення на основі можливих каузальних залежностей; розробка методу побудови процесно-орієнтованого представлення пояснення.

4. Процесна модель пояснення на основі можливих каузальних залежностей

Пояснення на процесному рівні деталізації описує послідовність дій, які сформуливали рішення інтелектуальної системи. Таке пояснення має процесний характер, оскільки відображає вплив декількох альтернативних варіантів процесу прийняття рішень на отриманий результат. Пояснення відображає причинно-наслідкові зв'язки між станами інтелектуальної системи, оскільки процес формування рішень представляється як упорядкована у часі послідовність станів ПС. Кожен реалізований варіант $E^{(j)}$ процесу формування рішень складається із станів $E_m^{(j)}$:

$$E^{(j)} = \langle E_1^{(j)}, E_2^{(j)}, \dots, E_{m-n}^{(j)}, \dots, E_m^{(j)}, \dots, E_{|E|}^{(j)} \rangle. \quad (1)$$

Сукупність реалізацій $E^{(j)}$ задає повний опис процесу: $E = \{E^{(j)}\}$. Відповідно, стани E_m процесу E охоплюють лише ідентичні стани $E_m^{(j)}$ із різних $E^{(j)}$:

$$E_m \equiv E_m^{(j)} | (\forall i \neq j) E_m^{(i)} \equiv E_m^{(i)}. \quad (2)$$

Кожен стан визначається множиною $\{c_{p,l}\}$:

$$E_m = \{c_{p,l}\}, \quad (3)$$

де l – кількість значень; p – кількість змінних.

Ці змінні характеризують властивості об'єктів або сутностей, використані при формуванні рішення.

Для опису процесу формування рішення у задачі побудови пояснень використовуються темпоральні залежності, які відображають послідовність подій, та каузальні залежності, що пояснюють причинно-наслідкові зв'язки між станами процесу.

Темпоральні залежності $g_{j,m}^{j,m-n}$ визначають порядок у часі для пар станів поточної реалізації процесу формування рішення:

$$g_{j,m}^{j,m-n} : E_{m-n}^{(j)} \rightarrow E_m^{(j)}. \quad (4)$$

В тому випадку, якщо $n=1$, залежність $g_{i,m}^{j,m-1}$ визначає порядок у часі для двох послідовних станів. Оскільки перехід між послідовними станами відбувається в результаті виконання однієї дії, дана залежність відображає причини і наслідки однієї дії процесу формування рішення. Відповідно, залежність $g_{i,m}^{j,m-n}, n > 1$ відображає причини $E_{m-n}^{(j)}$ та наслідки $E_m^{(j)}$ для послідовності із n дій поточної реалізації процесу формування рішення в ПС.

Узагальнена залежність для декількох альтернативних реалізацій процесу формування рішення в ПС має вигляд темпорального правила g_m^{m-n} :

$$g_m^{m-n} : E_{m-n} \rightarrow E_m. \quad (5)$$

У відповідності до (5), залежності із декількох альтернативних реалізацій процесу формування рішення поєднуються в одне правило g_m^{m-n} у випадку співпадіння $E_{m-n}^{(j)}$ та $E_m^{(j)}$ для декількох реалізацій процесу формування рішення.

Каузальні залежності $p_{j,m}^{j,m-n}$ та правила p_m^{m-n} формуються на основі темпоральних правил між станами процесу формування рішення

$$p_{j,m}^{j,m-n} : E_{m-n}^{(j)} \Rightarrow E_m^{(j)} | \exists g_{j,m}^{j,m-n}, \quad (6)$$

$$p_m^{m-n} : E_{m-n} \Rightarrow E_m | \exists p_m^{m-n}. \quad (7)$$

Опис відомих послідовностей дій (станів) процесу формування рішення на основі причинно-наслідкових зав'язків між станами $E_{rule}^{(j)}$ має вигляд:

$$E_{rule}^{(j)} = \left\{ p_{j,2}^{j,1}, \dots, p_{j,m}^{j,1}, \dots, p_{j,|E|}^{j,|E|-1} \right\}. \quad (8)$$

Відповідно, сукупність E_{rule} всіх відомих варіантів процесу формування рішення задається множиною зважених правил:

$$E_{rule} = \left\{ (p_2^1, w_2^1), \dots, (p_m^1, w_m^1), \dots, (p_{|E|}^{|E|-1}, w_{|E|}^{|E|-1}) \right\}. \quad (9)$$

Відмінність між виразами (1) та (8) полягає в тому, що у (1) відображена інформація про послідовність станів цільової реалізації процесу формування рішення, а у (8) – знання про порядок дій даного процесу. Тобто на основі (8) може бути сформовано пояснення виду «причинно-наслідковий зв'язок $p_{j,m}^{j,m-n}$ між станами $E_{m-n}^{(j)}$ та $E_m^{(j)}$ привів до поточного рішення системи».

Для того, щоб сформувати таке пояснення, необхідно визначити найважливіші залежності, які вплинули на рішення ПС, тобто оцінити правила p_m^{m-n} , які у поточному процесі формування рішення реалізовані у вигляді залежностей $p_{j,m}^{j,m-n}$. Загальна ідея такої оцінки полягає в тому, щоб надати найбільше значення ваги тим правилам, які використовуються в більшості реалізацій процесу формування рішення, оскільки такі правила є типовими і впливають не на окремі рішення, а на підмножину рішень ПС. Побудова правил (7) виконується в умовах невизначеності щодо процесу формування рішення, оскільки ПС на процесному рівні деталізації пояснень має вигляд «сірої скриньки». Відповідно, для таких правил використовується можливісна, а не ймовірнісна оцінка. Використання теорії можливостей [21] для оцінки причинно-наслідкових правил пов'язано з такими факторами. По-перше, ймовірнісна оцінка базується на частоті настання певної події у довгостроковій перспективі у межах певного процесу або явища. А можливісні оцінки показують ступінь узгодженості події з наявною інформацією про предметну область, тобто ступінь правдоподібності події з урахуванням доступної інформації. По-друге, можливісні оцінки використовуються в умовах неповноти інформації, на відміну від ймовірнісної оцінки, яка призначена для моделювання випадкових явищ. Тобто можливісні оцінки є загальнішими у порівнянні із ймовірнісними і використовуються в умовах невизначеності щодо станів цільового процесу. Згідно з [21], використовуються дві оцінки: можливість та необхідність, які визначають ступінь

правдоподібності певної події та ступінь впевненості у події.

Вага w_m^{m-n} кожного правила p_m^{m-n} визначається через його функцію можливості $\pi(p_m^{m-n})$. Остання розраховується через частоту використання правила у всіх доступних альтернативах процесу формування рішення. Можливість для набору правил $\Pi(E_{rule})$, що описують процес формування рішення, розраховується традиційно як максимальне значення $\pi(p_m^{m-n})$:

$$\Pi(E_{rule}) = \sup_{p_m^{m-n} \in E_{rule}} \pi(p_m^{m-n}). \quad (10)$$

Можливість для правил поточного процесу $E_{rule}^{(j)}$ розраховується аналогічно за умови, що правила p_m^{m-n} реалізовані в рамках j -го процесу у вигляді залежностей $p_{j,m}^{j,m-n}$:

$$\Pi(E_{rule}^{(j)}) = \sup_{p_m^{m-n} \in E_{rule}} \pi(p_m^{m-n}) | (\forall m, n) \exists p_{j,m}^{j,m-n}. \quad (11)$$

Тоді пояснення $Exp^{(j)}$ щодо поточного процесу $E^{(j)}$ формування рішень на основі каузальних залежностей має вигляд:

$$Exp^{(j)} = p_m^{m-n} : \pi(p_m^{m-n}) = \Pi(E_{rule}^{(j)}). \quad (12)$$

Тобто як пояснення вибирається правило з максимальним значенням функції пояснення $\pi(p_m^{m-n})$.

Для того, щоб користувач міг вибрати детальніший варіант пояснення, $Expl^{(j)}$ може бути представлено у вигляді підмножини правил p_m^{m-n} з найбільшими значеннями функції можливості $\pi(p_m^{m-n})$:

$$\begin{aligned} Expl^{(j)} &= \langle p_m^{m-n}, p_l^{l-k}, \dots \rangle : \\ \pi(p_m^{m-n}) &= \Pi(E_{rule}^{(j)}), \pi(p_m^{m-n}) > \pi(p_l^{l-k}). \end{aligned} \quad (13)$$

Згідно з (12), першою можливою причиною j -го рішення є дії (або послідовність дій), що задаються правилом p_m^{m-n} . Другою можливою причиною є правило p_l^{l-k} .

Пояснення (13) можуть бути деталізовані для користувача з урахуванням властивостей станів (3). Тобто користувачеві в рамках запропонованої моделі можуть бути надані значення атрибутів станів $E_{m-n}^{(j)}$ та $E_m^{(j)}$, які змінились згідно з правилом p_m^{m-n} . Як такі дані може бути представлена, наприклад, назва дії, яка є першою можливою причиною отриманого рішення.

Оцінка необхідності для $\Pi(E_{rule}^{(j)})$ розраховується через можливість доповнення $\neg E_{rule}$ до множини правил E_{rule} . Доповнення містить правила, не реалізовані у поточному

варіанті процесу $E^{(j)}$:

$$\neg E_{rule} = \{P_m^{m-n}\} | ((\forall m, n) \neg (\exists p_{j,m}^{j,m-n})). \quad (14)$$

Оцінка необхідності $N(E_{rule}^{(j)})$ згідно з теорією можливості базується на обчисленні можливості альтернативних правил:

$$N(E_{rule}^{(j)}) = 1 - \Pi(\neg E_{rule}). \quad (15)$$

Згідно з (15), чим нижче можливість альтернативних правил бути причиною рішення ПС, тим вище впевненість, що правила $Expl^{(j)}$ є можливою причиною рішення.

Така парна оцінка створює умови для того, щоб відібрати в $Expl^{(j)}$ підмножину правил з найбільшими значеннями функції можливості. Всі інші правила в такому випадку будуть належати доповненню $\neg E_{rule}$. Тоді користувач може відібрати підмножину правил в пояснення, орієнтуючись на значення необхідності: чим більше правил з високими значеннями функції можливості будуть перенесені в множину $\neg E_{rule}$, тим меншим буде значення необхідності $N(E_{rule}^{(j)})$. В результаті користувач ПС може відібрати в пояснення лише ключові правила – причини отриманого рішення.

5. Метод побудови процесно-орієнтованого пояснення для внутрішніх користувачів інтелектуальної інформаційної системи з використанням каузальних залежностей

Розроблений метод побудови пояснень використовує дані про послідовності станів для відомих реалізацій процесу формування рішення в інтелектуальній системі. Метод складається з фаз побудови та представлення пояснення. На фазі побудови метод послідовно формує темпоральні та каузальні залежності та правила, а також розраховує значення можливості та необхідності для отриманих залежностей. На фазі представлення метод використовує атрибути станів можливісних каузальних залежностей для того, щоб надати пояснення у зрозумілому для користувача вигляді. Як вхідні дані метод використовує інформацію, що містить лог (журнал подій) процесу формування рішень. Кожен такий лог складається з трас. Кожна траса містить послідовність записів про стани процесу для одного виконання такого процесу. Метод містить такі фази та етапи.

Фаза 1. Побудова пояснення.

Етап 1.1. Формування множини E послідовностей станів (1) на основі темпоральних міток у вхідних даних.

Етап 1.2. Формування темпоральних залежностей та правил.

Крок 1.2.1. Формування темпоральних залежностей $g_{j,m}^{j,m-n}$.

Крок 1.2.2. Формування темпоральних правил g_m^{m-n} .

Етап 1.3. Формування каузальних залежностей та правил.

Крок 1.3.1. Формування наборів каузальних залежностей $E_{rule}^{(j)}$ згідно з (6) та (8).

Крок 1.3.2. Формування набору каузальних правил E_{rule} згідно з (7) та (9).

Етап 1.4. Розрахунок можливості $\Pi(E_{rule}^{(j)})$ для правил поточного процесу $E_{rule}^{(j)}$.

Етап 1.5. Побудова пояснення $Exp^{(j)}$ щодо поточного процесу $E^{(j)}$ згідно з (11).

Етап 1.6. Побудова розширеного пояснення у вигляді підмножини правил p_m^{m-n} згідно з (13).

Етап 1.7. Розрахунок необхідності $N(E_{rule}^{(j)})$ для правил поточного процесу $E_{rule}^{(j)}$ згідно з (14), (15).

Фаза 2. Представлення пояснення.

Етап 2.1. Визначення граничних значень для атрибутів станів процесу формування рішення, що змінили своє значення.

Етап 2.2. Формування набору {граничні значення, поточне значення} для атрибутів, що змінили своє значення в рамках правил p_m^{m-n} із пояснення.

Етап 2.3. Представлення пояснення на основі граничних та поточних значень атрибутів.

На другій фазі виконується деталізація правил: на основі залежностей між станами визначаються залежності між атрибутами цих станів, які отримані із вхідних даних і тому є зрозумілими для користувача.

6. Експериментальна перевірка методу

Експериментальна перевірка методу виконана з використанням анонізованого медичного логу [22], що містить 1143 траси. Кожна траса відображає одну послідовність дій з лікування, в тому числі діагностичні процедури з використанням інтелектуальних систем. Одну із трас логу було вибрано як відображення поточного процесу. На базі даних логу було сформовано каузальні правила та розраховано показники можливості й необхідності для цих правил. Поточний процес містить 21 правило. Фрагмент набору правил для поточного процесу з відображенням атрибутів станів наведено у табл. 1.

Таблиця 1

Правила з розрахованим показником можливості

Стан 1	Стан 2	Тривалість (год:хв)	Ресурс 1	Ресурс 2	Кількість правил в журналі	Можливість
Register Patient	Initial Diagnosis	01:30	Admin A	Doctor B	1143	1.0
Register Patient	Request Lab Test	02:00	Admin A	Doctor B	987	0.86
Register Patient	Perform Lab Test	03:30	Admin A	Lab Technician C	987	0.86

Правила, які мають значення можливості 1.0, мають розглядатись як обмеження, оскільки вони виконуються у всіх варіантах реалізації процесу. Ці правила відображаються на локальному рівні представлення пояснення і не враховуються на процесному рівні. Два правила в табл. 1 – «Register Patient → Perform Lab Test» та «Register Patient → Request Lab Test» мають однакове значення можливості 0,86, які є максимальними для даного процесу. Тому обидва правила мають входити до складу пояснення.

Приклади правил із множини доповнення $\neg E_{rule}$ наведено в табл. 2. Ці правила не використовуються у поточному процесі, фрагмент якого представлено у табл. 1.

Правило «Schedule Follow-up → Discharge Patient» із множини доповнення $\neg E_{rule}$ було використано 892 рази. Тому значення можливості для правил, що не

використовуються в поточному процесі, становить $892 / 1143 = 0.78$. Відповідно, пояснення містить 2 правила із $P(E_{rule}^{(j)}) = 0.86$, $N(E_{rule}^{(j)}) = 0,22$.

Таблиця 2

Правила, що не використовуються в поточному процесі

Стан 1	Стан 2	Кількість правил
Emergency Admission	Initial Assessment	156
Perform Surgery	Post-op Care	423
Schedule Follow-up	Discharge Patient	892

На фазі 2 для атрибуту тривалості визначено граничні значення. Наприклад, правило «Register Patient → Perform Lab Test» має граничні значення часу виконання (01:00; 04:00). Тому процесно-орієнтоване пояснення з урахуванням наведених атрибутів має вигляд: дії «Register Patient → Perform Lab Test» виконали за 03:30 години Admin A та Lab Technician C за умови, що максимальний час виконання цих дій 04:00 годин, а мінімальний – 01:00 година.

7. Обговорення результатів дослідження

Розроблені процесно-орієнтована модель та метод побудови пояснень призначені для підтримки задач удосконалення процесу формування рішень в інтелектуальній системі, представленої у вигляді «сірої скриньки», на основі побудови інтерпретованого представлення даного процесу.

Пояснення щодо процесу формування рішення з використанням можливісних каузальних залежностей дає можливість виділити ключові дії процесу, що привели до поточного рішення, а також контекстні умови виконання цих дій.

Подальший розвиток розробленого методу пов'язаний із поєднанням можливісних правил із обмеженнями на допустимі послідовності дій процесу формування рішення.

8. Висновки

Запропоновано процесну модель пояснення в інтелектуальній інформаційній системі на основі можливісних каузальних залежностей. Модель містить набір каузальних залежностей, що відображають процес формування рішень в інтелектуальній системі, а також оцінки можливості та необхідності цих залежностей, що створює умови для визначення дій процесу, які мають найбільше значення можливості і, відповідно, ключовий вплив на рішення інтелектуальної системи.

Запропоновано метод побудови процесно-орієнтованого пояснення в інтелектуальній інформаційній системі на основі можливісних каузальних залежностей. Метод містить фази побудови та представлення пояснення. На фазі побудови виконується формування темпоральних й каузальних правил, що відображають процес формування рішення в інтелектуальній системі, а також визначення можливості та необхідності для цих правил. На фазі представлення виконується визначення граничних значень для атрибутів станів у складі правил та представлення пояснення з урахуванням цих значень. Метод створює умови для удосконалення процесу формування рішень в інтелектуальній системі на основі побудови інтерпретованого опису такого процесу.

Перелік посилань:

1. Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv preprint. <https://doi.org/10.48550/arXiv.1702.08608>
2. Ribeiro, M.T., Singh, S., & Guestrin, C. (2016). «Why should I trust you?»: Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. <https://doi.org/10.1145/2939672.2939778>

3. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM Computing Surveys*, 51(5), 1-42. <https://doi.org/10.1145/3236009>
4. Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, 6, 52138-52160. <https://doi.org/10.1109/ACCESS.2018.2870052>
5. van der Aalst, W.M.P., La Rosa M., & Santoro F.M. (2016). Business process management: Don't forget to improve the process! *Business & Information Systems Engineering*, 58(1), 1-6. <https://doi.org/10.1007/s12599-015-0409-x>
6. Gunning, D., & Aha, D. W. (2019). DARPA's Explainable Artificial Intelligence (XAI) Program. *AI Magazine*, 40(2), 44-58. <https://doi.org/10.1609/aimag.v40i2.2850>
7. Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267, 1-38. <https://doi.org/10.1016/j.artint.2018.07.007>
8. Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30, 4765-4774. <https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf>
9. Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 618-626). IEEE. <https://doi.org/10.1109/ICCV.2017.74>
10. Pearl, J. (2009). *Causality: Models, reasoning and inference* (2nd ed.). Cambridge University Press. <https://doi.org/10.1017/CBO9780511803161>
11. Peters, J., Janzing, D., & Schölkopf, B. (2017). *Elements of causal inference: foundations and learning algorithms*. MIT Press. <https://mitpress.mit.edu/books/elements-causal-inference>
12. Chalyi, S., & Leshchynskiy, V. (2020). Temporal representation of causality in the construction of explanations in intelligent systems. *Advanced Information Systems*, 4(3), 113-117. <https://doi.org/10.20998/2522-9052.2020.3.16>
13. Чалий С. Ф. Метод можливого оцінювання пояснення в системі штучного інтелекту / С. Ф. Чалий, В. О. Лещинський. Вісник Національного технічного університету «ХПІ». Сер. : Системний аналіз, управління та інформаційні технології = Bulletin of the National Technical University «KhPI». Ser.: System analysis, control and information technology: зб. наук. пр. Харків: НТУ «ХПІ», 2023. № 2 (10). С. 95-101.
14. Molnar, C. (2019). *Interpretable machine learning. A Guide for Making Black Box Models Explainable*. <https://christophm.github.io/interpretable-ml-book/>
15. Chalyi, Sergii & Leshchynskiy, V.. (2023). Оцінка чутливості пояснень в інтелектуальній інформаційній системі. Системи управління, навігації та зв'язку. Збірник наукових праць. 2. 165-169. 10.26906/SUNZ.2023.2.165
16. Chalyi, Sergii & Leshchynskiy, Volodymyr. (2023). Інформаційна технологія оцінки пояснень в інтелектуальній інформаційній системі. Системи управління, навігації та зв'язку. Збірник наукових праць. 4. 120-124. 10.26906/SUNZ.2023.4.120
17. Chala O. (2018) Models of temporal dependencies for a probabilistic knowledge base. *Econtechmod. An International Quarterly Journal*. Vol. 7, No. 3. P. 53 – 58.
18. Chala O. (2018) Models of temporal dependencies for a probabilistic knowledge base. *Econtechmod. An International Quarterly Journal*. Vol. 7, No. 3. P. 53 – 58.
19. Levykin V., Chala O. Development of a method of probabilistic inference of sequences of business process activities to support business process management. *Eastern-European Journal of Enterprise Technologies*. 2018. № 5/3(95). P. 16-24. DOI: 10.15587/1729-4061.2018.142664.
20. van der Aalst, W. M. P. (2016). *Process mining: Data science in action* (2nd ed.). Springer.
21. Dubois, D., Prade, H. (2022). Possibility Theory. In: Gläveanu, V.P. (eds) *The Palgrave Encyclopedia of the Possible*. Palgrave Macmillan, Cham. https://doi.org/10.1007/978-3-030-90913-0_175.
22. van Dongen, B. (2011). Real-life event logs - Hospital log [Dataset]. 4TU.ResearchData. <https://doi.org/10.4121/uuid:d9769f3d-0ab0-4fb8-803b-0d1120ffcf54>

Надійшла до редколегії 25.09.2024 р.

Чалий Сергій Федорович, доктор технічних наук, професор, професор кафедри ІУС ХНУРЕ, м. Харків, Україна, e-mail: serhii.chalyi@nure.ua; ORCID: 0000-0002-9982-9091
Лещинський Володимир Олександрович, кандидат технічних наук, доцент, доцент кафедри програмної інженерії ХНУРЕ, м. Харків, Україна, e-mail: volodymyr.leshchynskiy@nure.ua; ORCID: 0000-0002-8690-5702

О.В. ЧАЛА, С.О. БОГАТОВ

РОЗРОБКА ТЕМПОРАЛЬНОЇ МОДЕЛІ ПРЕДСТАВЛЕННЯ ЗНАНЬ ЩОДО БІЗНЕС-ПРОЦЕСІВ ПЕРШОГО РІВНЯ ЗРІЛОСТІ ПРОЦЕСНОГО УПРАВЛІННЯ

Розглянуто процесний підхід до управління на початкових рівнях процесної зрілості. Виконано структурування відмінностей між зрілими та незрілими бізнес-процесами з урахуванням можливостей використання темпоральних знань для переходу до вищих рівнів зрілості процесного управління. Показано, що ключові відмінності між зрілими та незрілими бізнес-процесами полягають у використанні знань для визначення послідовностей дій, що дозволяє адаптувати зрілі процеси до змін у вимогах користувачів та зовнішніх впливів. Запропоновано модель представлення знань щодо бізнес-процесу на основі зважених темпоральних правил, яка дозволяє оцінювати альтернативні варіанти виконання процесу. Модель дає змогу швидко адаптувати бізнес-процес шляхом вибору альтернативи з максимальною вагою за умов дотримання обмежень щодо умов зовнішнього середовища.

1. Вступ

Бізнес-процес (БП) – це послідовність взаємопов'язаних дій, орієнтованих на створення товарів або послуг, що мають цінність для споживача [1]. БП охоплює послідовність операцій зі створення товарів або послуг, ресурси, необхідні для виконання цих операцій, а також постачальників та клієнтів, які, відповідно, забезпечують ресурси для процесу та отримують його результати [1]. Послідовність робіт у БП визначається знаннями у формі бізнес-правил, що встановлюють умови та обмеження на можливі дії, зазвичай залежно від вимог постачальників, клієнтів та наявних ресурсів [3].

Процесне управління підприємством – це управління сукупністю взаємодіючих БП, що описують діяльність підприємства, з метою підвищення ефективності та адаптивності [1]. У процесному управлінні виділяють п'ять рівнів зрілості – від незрілих нестандартизованих БП першого рівня до зрілих інноваційних процесів п'ятого рівня [4], [5].

Процесне управління на початкових рівнях зрілості стикається з рядом проблем, які знижують ефективність діяльності підприємства в цілому. Ключовими серед таких проблем є неповторюваність процесів, відсутність стандартизації, обмежена адаптація до зовнішніх змін та впливів, обмежені можливості реінжинірингу процесів [4]. Згідно з моделлю ВРММ (Business Process Maturity Model), на першому рівні зрілості процеси часто виконуються хаотично, без чіткої структури та стандартизації [5]. Це ускладнює прогнозування результатів і може призводити до перевищення бюджету та термінів виконання проектів. Крім того, на початкових рівнях зрілості різні підрозділи можуть використовувати нестандартизовані підходи для виконання схожих завдань, що ускладнює координацію та інтеграцію діяльності підприємства [5]. Відсутність стандартизації ускладнює впровадження нових технологій та оптимізацію БП.

Сучасні організації працюють у динамічному середовищі, де зміни відбуваються достатньо швидко. Однак на початкових рівнях процесної зрілості ці зміни не завжди враховуються вчасно. Тобто такі процеси не містять правил щодо швидкої інтеграції нової функціональності, які дозволяють оперативно адаптувати послідовність дій до змін у зовнішньому середовищі, що знижує конкурентоспроможність процесно-орієнтованого підприємства. Слід також зазначити, що процеси початкових рівнів зрілості більше

орієнтовані на операційне управління та не містять знань щодо реагування на непередбачувані зовнішні впливи.

Проблема проведення реінжинірингу БП пов'язана із відсутністю у процесних моделях початкових рівнів зрілості знань у формі бізнес-правил, які визначають можливі та допустимі послідовності дій БП.

Таким чином, сукупність розглянутих проблем процесного управління початкових рівнів зрілості обумовлена відсутністю формалізованого представлення знань щодо послідовностей дій таких процесів. Такі знання мають темпоральний характер, оскільки визначають упорядкованість дій БП у часі. Зазначене свідчить про актуальність проблеми формування темпорального представлення знань щодо БП початкових рівнів зрілості.

2. Аналіз літературних даних і постановка проблеми дослідження

Останнім десятиріччям значна увага приділялась вирішенню проблем процесного управління та забезпеченню переходу до вищих рівнів зрілості процесного управління. У [6] розглядається інтеграція управління БП з аналітикою даних. Такий підхід дає можливість вирішувати проблеми нестабільності БП та відсутності стандартизації процесів, а також коригувати процеси у відповідності до вимог користувачів. Комплексний підхід до управління БП в рамках життєвого циклу з метою оптимізації процесів і вирішення проблем нестабільності та недостатньої адаптивності запропоновано в [7]. У [8] сформульовано десять принципів ефективного управління БП з тим, щоб запобігти фрагментації процесів та відсутності координації між підрозділами підприємства, що створює умови для вирішення проблем нестабільності процесів і відсутності стандартизації. В [9] розглядається проблема фрагментації БП внаслідок недостатньої координації між підрозділами. Пропонується використання інформаційних технологій для удосконалення взаємодії між підрозділами. Розглянуті підходи орієнтовані на вирішення проблем процесного управління з позицій інтелектуального аналізу процесів, підтримки життєвого циклу та коригування типових помилок при впровадженні процесного управління. Однак в рамках розглянутих підходів не було виконано узагальнення щодо представлення знань, які визначають як умови, так і обмеження на виконання послідовностей дій БП.

Питання використання знань БП розглянуто в [10] у контексті процесного управління в Індустрії 4.0 та 5.0. Пропонується розширення парадигми процесного управління з тим, щоб врахувати знання інтенсивних БП і створити умови для динамічного управління такими БП. Запропонований підхід орієнтований на вирішення проблеми адаптації до зовнішніх впливів. Проблема застосування багаторівневої моделі зрілості процесного управління у динамічному середовищі розглядається в [4]. Інтеграція динамічних можливостей дозволяє підприємству оперативно реагувати на ринкові зміни та підвищувати конкурентоспроможність. Дані підходи орієнтовані на використання знань в умовах інтенсивних змін зовнішнього середовища. Проте темпоральному аспекту знань не приділено достатньої уваги, тоді як використання темпоральних знань створює умови для коригування ходу робіт БП на основі порівняння ідеальної моделі процесу «як має бути» та реального процесу «як є». Темпоральні знання щодо реального процесу можуть бути отримані на основі аналізу логу (журналу подій), що містить інформацію про виконані у БП дії.

Питання представлення темпоральних знань у вигляді зважених темпоральних правил розглянуто в [11]. Темпоральні правила використовуються для побудови набору альтернативних послідовностей дій із вирішення однієї й тієї ж задачі. Для порівняння послідовностей дій використовується ймовірнісна оцінка правил [12]. В подальшому отримані послідовності дій використовуються для підтримки прийняття рішень [13],

побудови обмежень, шаблонів та каузальних залежностей для пояснень [14]-[17]. Побудова темпоральних правил виконується на основі упорядкованої у часі послідовності подій [18]. Однак в розглянутих дослідженнях не приділяється достатньо уваги задачі підтримки управління БП в умовах динамічних змін зовнішнього середовища на початкових рівнях зрілості процесного управління, тоді як дана задача може бути вирішена шляхом застосування темпоральних знань.

3. Мета і задачі дослідження

Метою дослідження є розробка темпорального підходу до представлення знань щодо БП першого рівня зрілості процесного управління з тим, щоб створити умови для побудови знання-орієнтованої процесної моделі керованого рівня процесної зрілості.

Для досягнення цієї мети в дослідженні вирішуються такі задачі:

- структуризація відмінностей зрілих та незрілих БП з урахуванням особливостей представлення темпорального аспекту БП;
- розробка моделі представлення знань щодо БП першого рівня зрілості процесного управління.

4. Структуризація відмінностей зрілих та незрілих бізнес-процесів

Незрілі процеси є характерними для першого етапу впровадження процесного управління. Вони не мають чіткої структури та не є стандартизованими. Такі БП часто виконуються на основі індивідуальних рішень працівників, що призводить до непослідовних результатів і відхилень у якості продукції або послуг. Натомість зрілі БП мають чітко визначену структуру, задокументовані процедури та стандартизовані підходи до виконання завдань, що забезпечує передбачуваність результатів і стабільну якість [5]. Таким чином, перехід від незрілих до зрілих БП потребує формування темпоральних знань, які визначають чітку послідовність робіт.

У незрілих процесах управління здійснюється реактивно, а не проактивно, що може призвести до затримок та перевищення бюджету [19]. У зрілих БП управління є проактивним і базується на кількісних показниках ефективності та результативності, що дозволяє передбачати проблеми й виконувати попереджувальні та коригувальні дії [20]. Незрілі процеси зазвичай є неефективними через дублювання зусиль і зайві дії, тоді як зрілі процеси орієнтовані на підвищення ефективності та часто підтримуються автоматизованими системами [21]. Для незрілих процесів характерними є слабка передбачуваність результатів, високі ризики помилок, проблеми з управлінням ресурсами. Навпаки, зрілі процеси мають передбачувані результати завдяки використанню статистичних методів управління варіаціями виконання БП [22].

Важливою відмінністю зрілих БП є системна підтримка удосконалення процесів. Незрілі процеси мають слабкі адаптаційні властивості та можуть бути удосконалені лише на евристичній основі через відсутність системного підходу до аналізу їхньої ефективності. У зрілих процесах впроваджуються механізми постійного вдосконалення, що дозволяє організації активно шукати нові можливості для оптимізації та інновацій [23]. Така підтримка реалізується на основі знань про умови та обмеження виконання дій БП.

Таким чином, для зрілих БП визначається набір альтернативних послідовностей робіт. Відповідно, оцінка таких послідовностей може бути виконана з використанням темпоральних підходів, в яких використовується числова оцінка альтернатив з урахуванням частоти фактичного виконання робіт [11].

Використання сучасних технологій також відрізняє зрілі процеси від незрілих. Зрілі процеси інтегрують технологічні рішення для автоматизації рутинних завдань, збору даних і аналізу показників ефективності в реальному часі, що дозволяє швидше реагувати на зміни ринку й покращувати взаємодію між різними підрозділами організації [24].

Порівняльну характеристику зрілих та незрілих БП наведено у табл. 1.

У підсумку, врахування темпорального аспекту при моделюванні БП дає можливість узагальнити підмножини альтернативних послідовностей подій на основі темпоральних знань, оскільки останні використовують відносний час [11]-[13].

Таблиця 1

Порівняльна характеристика зрілих та незрілих бізнес-процесів

Характеристика	Незрілі процеси	Зрілі процеси
Структурованість бізнес-процесів	Процеси є непослідовними, часто виконуються ad hoc, без чіткого плану.	Процеси чітко структуровані, задокументовані та стандартизовані на рівні організації, що свідчить про використання темпоральних знань для визначення послідовності робіт.
Особливості процесного управління	Відсутність систематичного управління; менеджери втручаються лише при виникненні проблем.	Управління є проактивним; бізнес-процеси містять множини альтернативних упорядкованих у часі послідовностей робіт; використовуються кількісні показники ефективності бізнес-процесів.
Ефективність	Низька ефективність через дублювання зусиль, зайві дії та нераціональне використання ресурсів, що збільшує ризик помилок.	Процеси орієнтовані на підвищення ефективності та зменшення витрат ресурсів.
Прогнозованість результатів	Результати є не завжди передбачуваними через відсутність визначеної темпорально упорядкованої послідовності дій бізнес-процесів; порядок дій залежить від людського фактору; відсутній моніторинг кожного варіанту виконання процесу.	Результати передбачувані завдяки моніторингу виконання процесу та вибору відповідної послідовності дій на основі знань про бізнес-процеси.
Адаптивність до змін	Процеси мають слабкі адаптивні властивості до змін у зовнішньому середовищі або ринку.	Процеси гнучкі, містять знання щодо можливих послідовностей дій та здатні адаптуватися до змін у зовнішньому середовищі.

Таке узагальнення створює умови для побудови багатоваріантної моделі реального БП «як є» на основі використання темпоральних знань. В подальшому вказана модель може бути використана для швидкої адаптації апріорно заданої моделі БП «як має бути».

5. Темпоральна модель представлення знань щодо бізнес-процесу

Запропонована модель представлення знань базується на упорядкуванні в часі пар подій БП. Кожна з таких подій фіксує факт завершення однієї з операцій БП. У незрілого БП, як було показано в табл. 1, відсутній моніторинг виконання дій кожного варіанту виконання БП. Зазвичай логування такого процесу відбувається в єдиний журнал, без розділення за окремими альтернативами. Вказаний підхід суттєво утруднює виділення знань щодо особливостей виконання реального БП «як є». Тому упорядкування та подальше узагальнення темпоральних залежностей між подіями у вигляді темпоральних

правил дає можливість визначити умови виконання послідовностей дій у БП «як є». В подальшому за допомогою темпоральних правил можуть бути сформовані нові послідовності дій для адаптації до змін у зовнішньому середовищі.

Кожна подія e_i БП відображає його стан після завершення відповідної дії у вигляді вектору значень властивостей процесу $\overline{v_i}$ та мітки часу t_i :

$$e_i = \langle \overline{v_i}, t_i \rangle. \quad (1)$$

Мітка фіксує у журналі час виникнення події.

Журнал подій E незрілого БП складається із неупорядкованої множини подій e_i :

$$E = \{e_i\}. \quad (2)$$

Оскільки опис кожної події містить мітку часу t_i , для кожної пари подій e_i, e_l може бути задана темпоральна упорядкованість:

$$\langle e_i, e_l \rangle : t_l > t_i. \quad (3)$$

Проте події e_i, e_l можуть належати до різних реалізацій БП. Такі реалізації зазвичай виконуються на різних інтервалах часу T_j . Множина інтервалів $T = \{T_j\}$ визначає множину реалізованих варіантів виконання БП. Тому темпоральна упорядкованість (3) може розглядатись як темпоральна залежність $u_{j,l}^{j,i}$ лише за умови, що пара подій e_i, e_l належить до однієї реалізації БП, тобто виникла на одному інтервалі часу:

$$u_{j,l}^{j,i} = \langle e_{j,i}, e_{j,l} \rangle \mid t_{j,l} > t_{j,i}, t_{j,l}, t_{j,i} \in T_j. \quad (4)$$

Декілька темпоральних залежностей можуть бути поєднані у темпоральне правило за умови співпадіння подій. Співпадіння подій базується на співпадінні значень властивостей $\overline{v_i}$:

$$r_l^i = \bigcup_j u_{j,l}^{j,i} \mid (\forall j \forall k) \overline{v_{j,i}} = \overline{v_{j,k}}. \quad (5)$$

Сукупність темпоральних правил $\{r_l^i\}$ є набором знань щодо БП початкового рівня процесної зрілості. Відповідно, набір темпоральних залежностей $\{u_{j,l}^{j,i}\}$ становить знання щодо одного варіанту виконання такого БП.

Порівняння варіантів виконання БП може бути виконано з використанням ваг темпоральних правил. Вага w_l^i темпорального правила r_l^i може бути отримана на основі співвідношення між кількістю темпоральних залежностей $\left| \{u_{j,l}^{j,i}\} \right|$, що входять до складу правила, та кількістю $|T|$ реалізацій БП:

$$w_l^i = \frac{\left| \{u_{j,l}^{j,i}\} \right|}{|T|}. \quad (7)$$

З урахуванням (1)-(7), темпоральна модель представлення знань БП складається із

множини зважених темпоральних правил:

$$M = \{r_l^i, w_l^i\}. \quad (8)$$

Розроблена модель дає можливість описати та порівняти на основі ваг правил реалізовані послідовності дій БП першого рівня процесної зрілості.

6. Експериментальна перевірка можливостей представлення бізнес-процесів першого рівня процесної зрілості з використанням темпоральних знань

Експериментальну перевірку можливостей запропонованої моделі виконано з використанням логу процесу сервісного обслуговування фірми VolvoIT. Даний лог відображає виконання зрілого БП і тому містить траси із упорядкованих у часі подій. Кожна траса відповідає одній реалізації БП. Ці траси було реалізовано на різних інтервалах часу T_j , оскільки сервісне обслуговування зазвичай проводиться не частіше, ніж раз на рік.

Фрагмент вхідного логу наведено на рис. 1. Траси виділяються тегами `<trace>`, `</trace>`. Події виділяються тегами `<event>`, `</event>`. В рамках останніх тегів задається набір властивостей події та мітка часу.

```
<trace>
  <string key="concept:name" value="1-353818791"/>
  <event>
    <string key="org:group" value="Org line C"/>
    <string key="resource country" value="Sweden"/>
    <string key="organization country" value="se"/>
    <string key="org:resource" value="Martin"/>
    <string key="organization involved" value="G292 3rd"/>
    <string key="org:role" value="C_3"/>
    <string key="concept:name" value="Queued"/>
    <string key="impact" value="Low"/>
    <string key="product" value="PROD815"/>
    <string key="lifecycle:transition" value="Awaiting Assignment"/>
    <date key="time:timestamp" value="2010-02-12T14:00:54+01:00"/>
  </event>
</trace>
```

Рис. 1. Фрагмент вхідного логу

При проведенні експерименту з вхідного логу було видалено траси таким чином, щоб він відповідав логу незрілого процесу у відповідності до (2).

Результати експерименту для 2, 3 та 5 трас вхідного логу наведено в табл. 2.

Таблиця 2

Представлення темпоральних знань

Кількість вхідних трас	Кількість темпоральних правил	Відновлення вхідних трас, %
2	24	64%
3	26	71%
5	31	79,4%

У процесі експерименту було виділено темпоральні правила для підмножини подій, яка складала 2 траси у вхідних даних. Потім отримана множина вигляду (2) послідовно доповнювалась подіями ще з двох трас. З використанням представленого в [12] методу, на основі темпоральних правил було сформовано нові траси. Отримані на нових трасах послідовності подій порівнювались із послідовностями подій вхідного логу. Точність відновлення трас встановлювалась за кількістю співпадаючих темпорально упорядкованих подій.

За результатами експерименту можна зробити висновок, що із збільшенням кількості правил у темпоральній моделі представлення знань БП збільшується відповідність вхідних та відновлених трас. Тому даний підхід дає можливість описати упорядкованість дій БП у часі та виділити окремі варіанти реалізації БП першого рівня процесної зрілості і на цій основі створити умови для переходу до наступного, контрольованого рівня процесного управління.

Подальший розвиток темпорального підходу пов'язаний із формуванням обмежень на можливі послідовності дій БП початкового рівня процесної зрілості. Використання обмежень дасть можливість знизити кількість можливих альтернативних реалізацій БП, що підвищить точність встановлення послідовностей дій процесу «як є» та знизить ризик помилок при виконанні процесу.

7. Висновки

Виконане порівняння зрілих та незрілих БП показало, що незрілі БП відрізняються слабкою передбачуваністю результатів, високими ризиками помилок, в тому числі при управлінні ресурсами, а також слабкими адаптаційними властивостями. Такі недоліки незрілих БП пов'язані з відсутністю формалізованого представлення знань щодо можливих і допустимих послідовностей дій процесу. Використання знань у зрілих БП дає можливість адаптувати БП до змін у вимогах користувачів та інших зовнішніх впливів шляхом вибору відповідної послідовності дій.

Вказані знання визначають послідовність дій БП у часі і тому мають темпоральний характер. Вони узагальнюють підмножини альтернативних варіантів виконання БП на основі використання відносного часу. Дане узагальнення створює умови для побудови моделі реального БП «як є» на основі використання темпоральних знань. Така модель містить альтернативні послідовності дій, реалізовані при імплементації БП. В подальшому модель «як є» може бути використана для швидкої адаптації моделі БП «як має бути».

Запропоновано темпоральну модель представлення знань щодо БП на основі зважених темпоральних правил, яка дає можливість порівняти альтернативні варіанти виконання процесів на основі порівняння ваг темпоральних правил, що створює умови для швидкої адаптації БП на основі вибору альтернативи із максимальною вагою за умови виконання обмежень у поточних умовах зовнішнього середовища.

Перелік посилань:

1. Hammer, M., & Champy, J. (1993). *Reengineering the corporation: A manifesto for business revolution*. Harper Business. https://books.google.com/books/about/Reengineering_the_Corporation.html?id=VpYgWyc16twC
2. Davenport, T. H. (1993). *Process innovation: Reengineering work through information technology*. Harvard Business School Press. https://books.google.com/books/about/Process_Innovation.html?id=9QAHngEACAAJ
3. Weske, M. (2012). *Business process management: Concepts, languages, architectures* (2nd ed.). Springer.
4. Yassien, E. (2020). The challenges of capability maturity model integration application in the dynamic environment. *International Journal of Information Systems and Change Management*, 12(1), 17–34.
5. Object Management Group. (2008). *Business Process Maturity Model (BPMM), Version 1.0*. <https://www.omg.org/spec/BPMM/1.0/PDF>
6. van der Aalst, W.M.P. (2013). *Business process management: A comprehensive survey*. ISRN Software Engineering, 2013.
7. Dumas, M., La Rosa, M., Mendling, J., & Reijers, H.A. (2018). *Fundamentals of Business Process Management* (Vol. 2). Springer.
8. vom Brocke, J., Schmiedel, T., Recker, J., Trkman, P., Mertens, W., & Viaene, S. (2014). Ten principles of good business process management. *Business Process Management Journal*, 20(4), 530–548.
9. Beerepoot, I., Di Ciccio, C., Reijers, H.A., & Rinderle-Ma, S. (2021). The biggest business process management problems of our time. *CEUR Workshop Proceedings*, Vol-2938.
10. Szelągowski, M., & Berniak-Woźny, J. (2024). BPM challenges, limitations and future development directions – a systematic literature review. *Business Process Management Journal*, 30(2), 505–557.
11. Levykin, V., & Chala, O. (2018). Method of determining weights of temporal rules in Markov logic network for building knowledge base in information control system. *EUREKA: Physics and Engineering*, 5, 3-10.

<http://dx.doi.org/10.21303/2461-4262.2018.00713>.

12. Levykin, V., & Chala, O. (2018). Development of a method of probabilistic inference of sequences of business process activities to support business process management. *Eastern-European Journal of Enterprise Technologies*, 5/3(95), 16-24. <https://doi.org/10.15587/1729-4061.2018.142664>.

13. Левикін, В. М., & Чала, О. В. (2018). Підтримка прийняття рішень в інформаційно-управляючих системах з використанням темпоральної бази знань. *Сучасні інформаційні системи*, 2(4), 101-107. <https://doi.org/10.20998/2522-9052.2018.4.17>.

14. Chalyi, S., Leshchynskyi, V., & Leshchynska, I. (2020). Method of forming recommendations using temporal constraints in a situation of cyclic cold start of the recommender system. *EUREKA: Physics and Engineering*, 4, 34-40.

15. Чалий, С. Ф., & Лещинський, В. О. (2020). Темпоральні патерни вподобань користувачів в задачах формування пояснень в рекомендаційній системі. *Бюника інтелекта*, 2 (95), 21-27.

16. Chalyi, S., Leshchynskyi, V., & Leshchynska, I. (2020). Detailing explanations in the recommender system based on matching temporal knowledge. *Eastern-European Journal of Enterprise Technologies*, 4(2 (106)), 6-13.

17. Chalyi, S., & Leshchynskyi, V. (2020). Temporal representation of causality in the construction of explanations in intelligent systems. *Advanced Information Systems*, 4(3), 113-117.

18. Чала, О. В. (2018). Побудова темпоральних правил для представлення знань в інформаційно-управляючих системах. *Сучасні інформаційні системи*, 2(3), 54-59. <https://doi.org/10.20998/2522-9052.2018.3.09>

19. Curtis, B., Alden, J., & Weber, C. V. (2009). The use of process maturity models in business process management. In *Handbook on Business Process Management 1* (pp. 439-458). Springer.

20. Tarhan, A., Turetken, O., & Reijers, H. A. (2016). Business process maturity models: A systematic literature review. *Information and Software Technology*, 75, 122-134.

21. Rosemann, M., & vom Brocke, J. (2015). The six core elements of business process management. In *Handbook on business process management 1* (pp. 105-122). Springer.

22. Van Looy, A., Poels, G., & Snoeck, M. (2017). Evaluating business process maturity models. *Journal of the Association for Information Systems*, 18(6), 461-486.

23. Lehnert, M., Linhart, A., & Röglinger, M. (2021). Advancing business process maturity for data-driven process improvements. *Business & Information Systems Engineering*, 63, 689-710.

24. Röglinger, M., Pöppelbuß, J., & Becker, J. (2012). Maturity models in business process management. *Business Process Management Journal*, 18(2), 328-346.

Надійшла до редколегії 25.09.2024 р.

Чала Оксана Вікторівна, доктор технічних наук, доцент, завідувачка кафедри РТІКС ХНУРЕ, м. Харків, Україна, e-mail: oksana.chala@nure.ua, ORCID: 0000-0001-8265-2480.

Богатов Євген Олегович, асистент кафедри ІУС ХНУРЕ, м. Харків, Україна, e-mail: bogatovevgeniy@gmail.com, ORCID: 0000-0002-0741-7242.

Н.В. ВАСИЛЬЦОВА, А.С. БУРКОВСЬКА

ВИБІР ПІДМНОЖИНИ МЕТОДІВ РОЗРОБКИ ІНТЕРФЕЙСУ МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ ВИВЧЕННЯ МОВ

Розглянуто основні існуючі методи, які використовуються під час розробки інтерфейсів мобільних застосунків для вивчення мов. Встановлено, що жоден з цих методів не має значних переваг, які б дозволили рекомендувати цей метод як найбажаніший інструмент розробки користувацького інтерфейсу мобільного застосунку. Як рішення задачі визначення та планування дій відносно необхідних методів розробки інтерфейсів мобільних застосунків для вивчення мов запропоновано адаптувати метод експертного оцінювання. Проведено апробацію адаптованого методу під час вирішення зазначеної задачі в межах ІТ-проєкту розробки мобільного застосунку з вивчення мов.

1. Вступ

Сучасні стандарти та методології розглядають життєвий цикл ІТ-продуктів як сукупність різних процесів, кожен з яких складається з окремих діяльностей. Але на початку майже кожного з цих процесів пропонується виконати діяльність, суть якої полягає в підготовці до виконання відповідного процесу. При цьому в межах цієї діяльності практично завжди пропонується вирішити задачу визначення та планування дій відносно необхідних забезпечуючих систем та послуг [1]. Суть цієї задачі полягає, зокрема, у виборі тих моделей, методів та інструментальних засобів, які, на думку стейкхолдерів, будуть необхідними для успішного виконання подальших діяльностей та вирішення окремих задач відповідного процесу життєвого циклу ІТ-продукту. В загальному випадку вирішення цієї задачі слід розглядати як послідовність таких дій:

а) вирішення задачі кількісного оцінювання кожного з елементів множини забезпечуючих систем та послуг, які можуть бути застосованими для успішного виконання діяльностей та задачі процесу;

б) вирішення задачі вибору з множини забезпечуючих систем та послуг підмножини необхідних забезпечуючих систем та послуг.

Але конкретні методи вирішення цих задач дуже сильно розрізняються залежно від того, до якого саме процесу життєвого циклу ІТ-продукту здійснюється підготовка. Методи вирішення таких задач можуть суттєво змінюватися, якщо управління цими процесами здійснюється на основі Agile або гібридних методологій управління ІТ-проєктами. Особливістю цих методологій є забезпечення широкої участі в управлінні ІТ-проєктом усіх його стейкхолдерів (а не тільки представників організації-розробника ІТ-продукту) [2], що ускладнює вирішення задачі кількісного оцінювання елементів множини забезпечуючих систем та послуг.

Серед розмаїття ІТ-продуктів, керування створенням яких здійснюється з використанням Agile або гібридних методологій управління ІТ-проєктами, на особливу увагу заслуговують мобільні застосунки. Причина цієї уваги полягає у постійному зростанні кількості людей, які намагаються застосовувати мобільні пристрої й застосунки у своїй бізнес-діяльності та для оптимізації соціальних процесів та відношень. Застосування розумних телефонів і планшетів, оснащених спеціалізованими застосунками, дозволяє прискорити обробку даних, поліпшити комунікацію між командами розробників та користувачами застосунків, ефективно керувати як ресурсами проєктів, так і розподілом часу користувачів [3]-[5]. Загалом, тенденція до переходу з простих мобільних пристроїв на багатофункціональні смартфони з кожним роком тільки посилюється, а створення

мобільної присутності стало питанням виживання й розвитку бізнесу. За даними Statcounter, 55 % від всього вебтрафіку припадає на мобільні телефони та планшети [6].

Головна особливість розробки мобільних застосунків полягає в формфакторі самих мобільних пристроїв. Однією з його прояв є маленький розмір екрану таких пристроїв, а в деяких випадках – і нестандартна форма. Тому дуже важливою частиною створення мобільного застосунку є розробка інтерфейсу користувача (User Interface – UI). Цим терміном один з видатних дослідників у галузі розробки UI Джеф Раскін пропонує позначати спосіб, яким людина виконує завдання за допомогою продукту, а саме дії, які вона виконує і що отримує натомість, [7]. У теперішній час UI є не тільки елементом дизайну мобільного застосунку, а й стратегічним інструментом для зростання бізнесу і задоволення потреб користувачів. Тому дослідження методів та способів вирішення задачі визначення та планування дій відносно необхідних методів розробки UI мобільного застосунку (Mobile UI) є актуальним.

2. Аналіз літературних даних і постановка проблеми дослідження

2.1. Аналіз особливостей процесу розробки інтерфейсу користувача мобільного застосунку для вивчення мов

Для вирішення задачі визначення та планування дій відносно необхідних методів розробки Mobile UI для вивчення мов слід розуміти особливості процесу розробки такого UI. Тому, перш за все, треба проаналізувати відмінності цього процесу від процесу проектування UI для інших різновидів ІТ-продуктів.

В рамках мобільних застосунків для вивчення мов можна виділити дві категорії таких застосунків:

- спеціалізовані, які розробляються для курсів з вивчення мов;
- загальні, які розробляються для всіх, хто хоче вивчити мову самостійно [8].

Усі мобільні застосунки для вивчення мов також можна поділити за декількома критеріями, використовуючи іншу класифікацію [9]: за віком, для якого призначено матеріал; за рівнем володіння мовою; за обсягом опрацьованого матеріалу; за експертністю розробника; за популярністю, тобто кількістю завантажень.

Незалежно від різновиду ІТ-продукту, основою процесу розробки UI є ключові принципи, що гарантують легкість використання UI користувачами та їхнього задоволення. До таких принципів належать [10]:

- простота та наочність представленої інформації;
- інтуїтивність, яка дозволяє користувачам діяти без додаткових пояснень (тобто чітка структура інтерфейсу, зрозумілі значки та лаконічні назви тощо);
- доступність (інтерфейс повинен бути доступний для всіх користувачів, включаючи людей з обмеженими можливостями);
- адаптивність (забезпечення зручності використання незалежно від пристрою – комп'ютер це, планшет або смартфон);
- респонсивність (реагування на дії користувача миттєво, забезпечуючи швидкий відгук на натискання кнопок і взаємодію з елементами);
- юзабіліті (ергономічність) та тестування (виявлення слабких місць та гарантування відповідності стандартам простоти, інтуїтивності, доступності, адаптивності та респонсивності).

Mobile UI істотно відрізняються від традиційних десктопних або вебінтерфейсів з ряду причин, які обумовлені специфікою мобільних пристроїв та контекстом їх використання [11]. Можна виділити такі основні відмінності Mobile UI:

- мобільні пристрої мають значно менший екран у порівнянні з десктопами та ноутбуками, що вимагає більшої уваги до економії простору, оптимізації розміщення елементів інтерфейсу користувача та чіткості зображень і тексту;

- основним методом введення є сенсорний екран, що вимагає створення інтерфейсу, який легко керується пальцями, з великими кнопками та іконками, що легко натискати;
- мобільні пристрої підтримують як портретну, так і ландшафтну орієнтацію, що вимагає від Mobile UI бути гнучким й адаптивним до зміни орієнтації;
- мобільні пристрої мають менший запас енергії, ніж десктопи, тому Mobile UI повинні бути оптимізовані для ефективного споживання енергії та ресурсів;
- мобільні пристрої часто використовуються «на ходу», що вимагає від Mobile UI бути простим і зручним для швидкого доступу та взаємодії;
- мобільні пристрої мають вбудовані датчики та можливості (наприклад, GPS, акселерометр, камера), які можуть інтегруватися з мобільними застосунками.

Розробка Mobile UI включає такі чіткі та структуровані етапи, які визначають завдання та ведуть до створення вдалого продукту: створення концепції; брейнсторм (використання методу мозкового штурму) та розробка ескізів; створення діаграми переходів; вибір стилю Mobile UI; проектування Mobile UI; прототипування дизайну та демонстрація; доопрацювання обраного концепту дизайну [12].

Аналіз етапу проектування Mobile UI для вивчення мов показав, що він має специфіку, пов'язану з розумінням логіки користувача та його потреб. На цьому етапі створюється артефакт «персонаж», який описується атрибутами: «ім'я», «вік», «статус», «звички», «потреби» та «інтереси». На основі взаємодії з цим артефактом розробляється сценарій користувача, який передбачає поведінку клієнта. Ця особливість виконання етапу проектування Mobile UI та необхідність врахування на цьому етапі розглянутих вище відмінностей Mobile UI обумовлюють необхідність аналізу існуючих методів розробки інтерфейсу для виявлення особливостей залучення користувачів мобільних застосунків у процес розробки Mobile UI.

2.2. Аналіз особливостей методів розробки інтерфейсу мобільних застосунків для вивчення мов

Загалом до множини методів розробки інтерфейсу мобільних застосунків для вивчення мов пропонується віднести такі методи [13]–[18]:

- метод з використанням гейміфікації;
- метод з використанням інтерактивних вправ;
- метод з використанням мультимедійних матеріалів;
- метод з використанням персоналізації контенту;
- метод з використанням системи обговорень та спільноти;
- метод з використанням подкастів та аудіоуроків;
- метод з використанням мікронавчання (Microlearning);
- метод з використанням регулярних повторень;
- метод з використанням тестування та зворотного зв'язку;
- метод з використанням User Research.

Для порівняльного аналізу переваг і недоліків зазначених вище методів розробки інтерфейсу мобільних застосунків для вивчення мов пропонується використати такі критерії:

- критерій K1, який визначає застосування переваг мобільних платформ (наприклад, таких, як аудіо);
- критерій K2, який визначає залученість користувачів;
- критерій K3, який визначає більшу стислість та структурованість матеріалу;
- критерій K4, який визначає заохочення користувачів до більшої активності через те, що телефон завжди поряд та легко доступний;
- критерій K5, який визначає наявність базової функціональності, грошові витрати та

витрати часу.

Перевіримо кожний з методів на відповідність заданим критеріям й позначимо символом «+» відповідність заданому критерію. Результати цієї перевірки наведено в табл. 1.

Таблиця 1

Порівняння методів розробки інтерфейсу з використанням визначених критеріїв

Методи розробки інтерфейсу	Порівняльні критерії				
	K1	K2	K3	K4	K5
Метод з використанням гейміфікації	-	+	-	+	-
Метод з використанням інтерактивних вправ	-	-	-	+	+
Метод з використанням мультимедійних матеріалів	+	-	-	-	+
Метод з використанням персоналізації контенту	-	+	-	+	-
Метод з використанням системи обговорень та спільноти	-	-	-	+	-
Метод з використанням подкастів та аудіоуроків	+	-	-	-	+
Метод з використанням мікронавчання	-	-	+	-	+
Метод з використанням регулярних повторень	-	+	-	+	-
Метод з використанням тестування та зворотного зв'язку	-	-	-	-	+
Метод з використанням User Research	-	+	-	+	+

Детальний опис переваг та недоліків методів розробки інтерфейсу, які підлягають аналізу, наведено відповідно в табл. 2 та табл. 3.

Таблиця 2

Переваги методів розробки інтерфейсу

Метод розробки інтерфейсу	Переваги методу
1	2
Метод з використанням гейміфікації	забезпечує конкретну та зрозумілу оцінку здобутків користувача; стимулює до продовження активності.
Метод з використанням інтерактивних вправ	сприяє активному засвоєнню матеріалу, оскільки користувачі здійснюють практичну діяльність; заохочує користувачів до активної участі та до вирішення практичних завдань; покращує засвоєння навичок та їх застосування у реальних ситуаціях.
Метод з використанням мультимедійних матеріалів	покращує розуміння та запам'ятовування навчального матеріалу завдяки візуальному та аудіосприйняттю; допомагає вивчати вимову та розвивати навички аудіювання; робить навчальний процес цікавішим завдяки різноманітності мультимедійних форматів.

Продовження таблиці 2

1	2
Метод персоналізації контенту	дозволяє користувачам навчатися за власним графіком та вибирати теми, які їх цікавлять; сприяє ефективному засвоєнню матеріалу, оскільки навчання адаптується до потреб кожного користувача; забезпечує вищий рівень мотивації, оскільки користувачі більше зацікавлені у вивченні тем, які їм потрібні.
Метод використанням аудіоматеріалів	користувачі можуть слухати аудіоуроки під час переміщення, під час фізичної активності, що робить навчання мобільним і доступним у різних ситуаціях; аудіоконтент дозволяє користувачам поліпшувати навички аудіювання, розуміння мови та правильної вимови; аудіоуроки можуть бути більш концентрованими та спрямованими на основний матеріал, що допомагає уникнути розсіювання уваги, яке може бути спричинено відеоконтентом; аудіоматеріали відкривають доступ до навчання для тих, у кого обмежене візуальне сприйняття.
Метод мікронавчання	дозволяє користувачам вивчати новий матеріал в стислих інтервалах часу, що сприяє збереженню уваги і зменшенню втоми; короткі навчальні модулі легко вписуються в розклад занять та можуть бути використані «на ходу», що робить метод мобільним; надає можливість вивчати новий матеріал навіть при обмеженому часі; дозволяє сконцентруватися на важливих аспектах навчання, уникнути інформаційного перенасичення та запам'ятати ключову інформацію.
Метод з використанням тестування та зворотного зв'язку	дозволяє оцінювати, наскільки користувачі засвоїли матеріал та де потрібне покращення; на основі результатів тестів дозволяє надавати користувачам персоналізовані рекомендації та матеріали для вивчення слабких місць; зворотний зв'язок дає можливість користувачам отримати пояснення до неправильних відповідей та вдосконалити свої знання; тестування може стимулювати користувачів до активного навчання та прагнення досягнути кращих результатів.
Метод з використанням системи обговорень та спільноти	забезпечує можливість обговорення складних питань та отримання різних поглядів на матеріал; сприяє вирішенню проблем та уточненню незрозумілих аспектів навчального матеріалу; створює відчуття спільноти та підтримки, що підвищує мотивацію.
Метод з використанням User Research	забезпечує глибоке розуміння потреб і поведінки користувачів, можливість вивчити їхні вподобання та очікування; допомагає зрозуміти, як користувачі взаємодіють з додатком або веб-сайтом, це допомагає вирішенню проблем та збагаченню досвіду користувача, що може призвести до більшої задоволеності користувачів і підвищення їхньої лояльності; допомагає виявляти проблеми та помилки в інтерфейсі або функціональності до того, як вони стануть критичними, це може заощадити час і кошти на подальших виправленнях; розуміння потреб користувачів оптимізує процеси та функціональність застосунку, що може підвищити конверсію і прибутковість; допомагає перевірити гіпотези та ідеї, перш ніж вкладати ресурси у їх розробку, це може запобігти створенню продукту, який буде непопулярним серед користувачів.

Кінець таблиці 2

1	2
Метод з використанням регулярних повторень	ефективний в покращенні запам'ятовування, дозволяє користувачам ефективно засвоювати матеріал і утримувати його в пам'яті протягом тривалого періоду; завдяки систематичному повторенню користувачам не потрібно витратити багато часу на повторне вивчення матеріалу, який вже був вивчений; може бути налаштованим під індивідуальні потреби користувача; вибір інтервалів повторень може залежати від складності матеріалу та інших факторів; користувачі зберігають більше інформації в довгостроковій пам'яті.

Таблиця 3

Недоліки методів розробки інтерфейсу

Метод розробки інтерфейсу	Недоліки методу
1	2
Метод з використанням гейміфікації	може занадто спрощувати складність досягнень.
Метод з використанням інтерактивних вправ	вимагає створення та підтримки багатофункціональних завдань, що може бути об'ємним за часом; не підходить для всіх видів навчального матеріалу та може бути неефективним у випадках, коли необхідно набути теоретичних знань без практичного застосування.
Метод з використанням мультимедійних матеріалів	вимагає наявності відповідних ресурсів для створення та підтримки мультимедійного контенту; може бути неефективним для користувачів з обмеженими можливостями або тих, хто віддає перевагу текстовому сприйняттю; потребує більшого обсягу пропускну здатності для завантаження мультимедійних файлів.
Метод з використанням аудіоматеріалів	відсутність зображень або текстового супроводу може ускладнити розуміння деяких концепцій або відповідності слів та їх вимови; поганий звук, шум або низька якість аудіо можуть негативно вплинути на сприйняття інформації; аудіофайли можуть мати більший об'єм даних порівняно з текстом, що може вимагати більшої пропускну здатності для їх завантаження; у порівнянні з іншими методами розробки інтерфейсу, аудіоуроки можуть бути менш інтерактивними та менше підходити для вправ та тестів.
Метод персоналізації контенту	вимагає розробки та підтримки індивідуальних навчальних матеріалів, що може бути ресурсовитратним; може стати складним у випадках, коли потрібно відслідковувати та оцінювати прогрес кожного користувача окремо; необхідний високий рівень персоналізації, щоб забезпечити ефективність методу.
Метод мікронавчання	в обмеженому часі мікронавчання не завжди дозволяє розглянути тему або матеріал достатньо повно; якість навчального контенту важлива, оскільки короткі сесії вимагають якісного інформаційного матеріалу; деякі теми можуть вимагати тривалих інтервалів навчання й поглибленого вивчення.

Кінець таблиці 3

1	2
Метод з використанням тестування та зворотного зв'язку	тестування може викликати стрес у деяких користувачів, особливо якщо вони відчувають переважання або низьку впевненість у собі; формати тестів можуть бути обмеженими і не завжди відображати всі можливі аспекти матеріалу; для ефективного використання методу потрібна система оцінювання відповідей, що може бути ресурсовитратним завданням; деякі користувачі можуть завищувати свої знання або підсвідомо обирати легші тести, що може впливати на об'єктивність результатів.
Метод з використанням системи обговорень та спільноти	вимагає модерації та контролю, оскільки може виникати небажана агресія або недружелюбне спілкування серед користувачів; не завжди ефективний для навчання, оскільки якість інформації може варіюватися; деякі користувачі можуть не відчувати себе комфортно у спільноті або не бажати спілкуватися з іншими.
Метод з використанням User Research	результати дослідження можуть бути суб'єктивними і залежати від інтерпретації дослідників, це може вплинути на об'єктивність даних.
Метод з використанням User Research	обробка та інтерпретація результатів дослідження може бути складним завданням, особливо у випадку великої кількості даних; у деяких випадках користувачі можуть невідповідально ставитись до дослідження або надавати неправдиву інформацію, що може спотворити результати дослідження; проведення User Research вимагає участі досвідчених спеціалістів, які знають, як правильно ставити питання і аналізувати отримані дані.
Метод з використанням регулярних повторень	вимагає дисципліни з боку користувача, оскільки важливо вчасно виконувати повторення; недотримання графіку може знизити ефективність методу; налаштування оптимальних інтервалів для повторень може вимагати експертного знання або використання спеціалізованих інструментів; підходить для вивчення фактичних даних та визначень, але може бути неефективним для вивчення складних концепцій або творчих навичок; для деяких користувачів постійні повторення можуть виглядати монотонно, що може вплинути на мотивацію навчання.

Базуючись на наведених у табл. 1-3 результатах, можна зробити такі висновки:

а) з розглянутої множини методів розробки інтерфейсу найрелевантнішим визначеним критеріям є метод з використанням User Research, але його релевантність недостатня для однозначного вибору цього методу для розробки Mobile UI (відповідає тільки трьом з п'яти критеріїв);

б) переважна більшість проаналізованих методів розробки інтерфейсу (окрім методу з використанням системи обговорень та спільноти і методу з використанням тестування та зворотного зв'язку) мають однаково низьку релевантність визначеним критеріям (відповідають двом з п'яти критеріїв), недостатню для обґрунтованого вибору цих методів для розробки Mobile UI;

в) метод з використанням системи обговорень та спільноти і метод з використанням тестування та зворотного зв'язку мають найменшу релевантність визначеним критеріям (відповідають одному з п'яти критеріїв), що перешкоджає обґрунтованому вибору цих методів для розробки Mobile UI.

Ці висновки надають можливість стверджувати, що спосіб вирішення задачі визначення та планування дій відносно необхідних методів розробки Mobile UI, який

базується на аналізі переваг і недоліків кожного з методів без врахування думок представників усіх стейкхолдерів, не дає об'єктивно обґрунтованого бажаного результату. Тому проведення досліджень моделей і методів вирішення задачі вибору підмножини методів розробки Mobile UI, що враховували б вимоги та особливості усіх стейкхолдерів IT-проєкту створення мобільних застосунків для вивчення мов, слід вважати актуальним як в теоретичному, так і в прикладному аспектах.

3. Мета і задачі дослідження

Метою даного дослідження є розробка способу вирішення задачі визначення та планування дій відносно необхідних методів розробки Mobile UI, який враховував би відношення до цих методів усіх стейкхолдерів IT-проєкту розробки мобільного застосунку для вивчення мов. Застосування цього способу дозволить покращити об'єктивність вирішення даної задачі в IT-проєктах, управління якими здійснюється із застосуванням Agile та гібридних методологій.

Для досягнення цієї мети у дослідженні вирішуються такі задачі:

- адаптація методу експертного оцінювання до особливостей вирішення задачі визначення та планування дій відносно необхідних методів розробки Mobile UI;
- апробація отриманих результатів в процесі розробки UI мобільного застосунку для вивчення мов.

4. Матеріали і методи дослідження

Об'єктом дослідження є задача визначення та планування дій відносно необхідних забезпечуючих систем та послуг. Ця задача розглядається як частина діяльності «Підготовка до реалізації» процесу реалізації елементів IT-продукту як системи [2]. Предметом дослідження є методи вирішення даної задачі для визначення та планування дій відносно необхідних методів розробки Mobile UI.

Головна особливість даного дослідження полягає у необхідності під час вирішення задачі визначення та планування дій відносно необхідних методів розробки Mobile UI враховувати потреби та вимоги не тільки виконавців IT-проєкту розробки мобільного застосунку, а й інших стейкхолдерів цього проєкту. Тому основна гіпотеза даного дослідження полягає у визнанні відсутності переваг кількісних оцінок методів розробки Mobile UI, визначених представниками розробників мобільного застосунку, перед такими ж оцінками, визначеними представниками майбутніх користувачів цього застосунку.

Для здійснення порівняльної оцінки методів розробки інтерфейсу пропонується використати метод експертної оцінки, залучаючи в ролі експертів як команду розробки застосунку, так і користувачів. Цей метод включає у себе вирішення завдань експертами, їхню аргументацію, визначення кількісних оцінок та подальшу обробку цих оцінок за допомогою формальних методів [19].

Існують різні форми визначення кількісних оцінок досліджуваних методів розробки Mobile UI, такі як дискусія, анкетування, інтерв'ю, мозковий штурм, нарада тощо. Іноді різні форми використовуються в комплексі.

При обробці оцінок експертів для отримання узагальненої інформації про досліджуваний об'єкт або явище, а також для формування рішення, що відповідає меті експертизи, використовують різні методи обробки індивідуальних оцінок. Вибір конкретного методу залежить від складності проблеми, форми представлення експертних оцінок і загальної мети експертизи.

Під час обробки експертних оцінок вирішуються такі завдання:

- формування узагальненої експертної оцінки (зібрані індивідуальні оцінки експертів об'єднуються для створення узагальненої інформації про досліджуваний об'єкт або явище);

– визначення відносних ваг об'єктів (оцінки експертів, в залежності від їхньої кваліфікації або досвіду, можуть мати різну вагу, тому методи визначення відносних ваг допомагають враховувати індивідуальні властивості експертів);

– встановлення ступеня узгодженості думок експертів (для забезпечення надійності узагальнених результатів важливо визначити ступінь узгодженості думок між експертами, тому це може бути важливою інформацією для подальшого аналізу та прийняття рішень).

Для вирішення цих завдань та врівноваження різноманітних експертних оцінок часто використовуються методи математичної статистики [20]. Зазвичай для формування узагальненої оцінки для забезпечення більшої стійкості узагальненої експертної оцінки можна використати:

– метод середніх арифметичних рангів;

– метод медіан рангів.

Метод середніх арифметичних рангів дозволяє обчислювати середнє арифметичне рангів, що були присвоєні об'єктам, за формулою:

$$\bar{x} = \left(\frac{1}{m} \right) \sum_{j=1}^m x_j, \quad (1)$$

де \bar{x} – середнє арифметичне оцінок групи експертів; x_j – оцінка j -го експерта, $j = \overline{1, m}$; m – кількість експертів.

За середніми рангами будується підсумковий ранжований ряд, упорядкований за зростанням середніх рангів.

Метод медіан рангів можна використати таким чином:

– для одного об'єкта відзначаються ранги, отримані експертами, і розташовуються в порядку зростання;

– медіаною призначається оцінка, для якої кількість експертів, що отримали більший ранг, дорівнює кількості експертів, що отримали менший ранг, тобто визначається значення рангу, розташоване на центральному місці всієї послідовності рангів;

– за медіанами рангів будується підсумковий ранжований ряд, упорядкований за зростанням медіан, тобто чим менша медіана рангу, тим кращий об'єкт.

Для більшої об'єктивності та стійкості узагальненої експертної оцінки рекомендується використовувати обидва методи одночасно.

Суб'єктивізм особи, яка вибирає метод обробки вихідних експертних оцінок, може впливати на загальні висновки, тому використання обох розглянутих методів дозволяє отримати повніший та збалансованіший результат.

Для визначення ваги кожного з досліджуваних об'єктів пропонується використовувати такий алгоритм.

Крок 1. Розрахунок ваги кожного об'єкта для кожного експерта за формулою:

$$w_{ij} = \left(\frac{1}{n} \right) \sum_{i=1}^n x_{ij}, \quad (2)$$

де w_{ij} – вага i -го об'єкта, розрахована за оцінками j -го експерта; x_{ij} – оцінка об'єкта i , яка дається j -им експертом; n – кількість об'єктів, що порівнюються.

Крок 2. Розрахунок середньої ваги кожного об'єкта за формулою:

$$w_i = \left(\frac{1}{m} \right) \sum_{j=1}^m w_{ij}, \quad (3)$$

де w_{ij} – вага i -го об'єкта, розрахована за оцінками j -го експерта; w_i – середня вага i -го об'єкта; n – кількість експертів.

Отримані таким чином ваги можна використовувати для подальших розрахунків або для прийняття рішень щодо важливості кожного об'єкта з множини об'єктів, яка розглядається.

Визначення рівня узгодженості або розбіжностей між оцінками різних експертів є важливою частиною групової експертизи. Для цього можна використовувати різні метрики або методи. Однією з таких метрик є коефіцієнт конкордації. Він дозволяє визначити міру згоди або узгодженості між різними оцінками, наданими експертами. Цей коефіцієнт приймає значення від 0 до 1, де 0 означає повну відсутність згоди, а 1 – повну узгодженість [20].

Розрахунок коефіцієнту конкордації W має здійснюватися за формулою:

$$W = \frac{12 * S}{m^2(n^3 - n)}, \quad (4)$$

де $S = \sum_{i=1}^n \left[\sum_{j=1}^m x_{ij} - \frac{1}{2} m(n+1) \right]^2$; m – кількість експертів; n – кількість об'єктів, що порівнюються.

5. Результати дослідження

5.1. Результати адаптації методу експертного оцінювання до особливостей вирішення задачі визначення та планування дій відносно необхідних методів розробки Mobile UI застосунку для вивчення мов

Основну увагу під час адаптації методу експертного оцінювання до особливостей вирішення задачі визначення та планування дій відносно необхідних методів розробки Mobile UI застосунку для вивчення мов слід було приділити вибору методу збору оцінок окремих експертів. Для цього запропоновано створити опитувальники для обох груп експертів (користувачів і команди розробки).

Питання для цих опитувальників відносно методів, які аналізуються, для користувачів застосунку та команди його розробників було запропоновано сформулювати окремо.

Сформований опитувальник для користувачів мобільного застосунку представлено в табл. 4. Сформований опитувальник для представників команди розробки мобільного застосунку представлено в табл. 5.

В опитувальнику для користувачів для кожного методу розробки інтерфейсу представлено конкретні питання, спрямовані на оцінку його зручності, ефективності та задоволення від взаємодії. Індивідуальні оцінки для кожного методу надаються за шкалою від 1 до 10, де 1 вказує на найменший рівень задоволення, а 10 – на найвищий [21].

В опитувальнику для користувачів для кожного методу розробки інтерфейсу представлено конкретні питання, спрямовані на оцінку його технічної складності, вартості реалізації, можливостей майбутнього розширення та підтримки. Оцінки також надаються за шкалою від 1 до 10, проте з урахуванням технічних аспектів розробки, де 1 вказує на низьку складність реалізації функціоналу, а 10 – на високу.

Таблиця 4

Сформований опитувальник для користувачів мобільного застосунку

Номер питання	Питання
1	2
Метод з використанням гейміфікації	
П1	Як впливає можливість отримання балів та винагород у вигляді досягнень на вашу мотивацію вивчати іноземну мову?
П2	Чи хотіли б ви змагатися з іншими користувачами для покращення ваших здібностей вивчення мови?
Метод з використанням інтерактивних вправ	
П3	Чи хотіли б ви вирішувати інтерактивні вправи для покращення розуміння іноземної мови?
П4	Наскільки інтерактивні вправи сприяють вашому зануренню в навчання мови?
Метод з використанням мультимедійних матеріалів	
П5	Як впливає використання аудіо- та відеоматеріалів на ваше сприйняття та вивчення іноземної мови?
П6	Чи полегшує перегляд навчальних матеріалів освоєння вимови та нюансів іноземної мови?
Метод з використанням персоналізації контенту	
П7	Наскільки важливо для вас самому підбирати види завдань, складність і підходи до вивчення іноземних мов?
П8	Чи сприяє персоналізація контенту вашому індивідуальному прогресу у вивченні мови?
Метод з використанням системи обговорень та спільноти	
П9	Як впливає участь у форумах та обговореннях на ваше розуміння та використання іноземної мови?
П10	Чи покращує обмін досвідом та порадами з іншими користувачами ваші мовні навички?
Метод з використанням подкастів та аудіоуроків	
П11	Як впливає прослуховування подкастів та аудіоуроків на ваше вивчення мови?
П12	Чи полегшує можливість слухати носіїв мови розуміння інтонації та вимови?
Метод з використанням мікронавчання	
П13	Як ви оцінюєте ефективність мікрівправ для вивчення нових слів або граматичних правил?
П14	Чи полегшує можливість вивчення мови у невеликих порціях засвоєння матеріалу?
Метод з використанням регулярних повторень	
П15	Як оцінюєте регулярні повторення для закріплення навчального матеріалу?
П16	Як часто використовуєте повторення для закріплення вивченого матеріалу?

Кінець таблиці 4

1	2
Метод з використанням тестування та зворотного зв'язку	
П17	Чи хотіли б ви тестувати новий функціонал для вивчення мов?
П18	Як ви оцінюєте важливість зворотного зв'язку для вашого подальшого вивчення?
Метод з використанням User Research	
П19	Наскільки ви вважаєте важливим залучення користувачів до досліджень та опитувань під час створення та подальшого розвитку додатку?
П20	Чи сприяє покращенню якості навчального процесу врахування потреб та думок користувачів у процесі розробки?

Таблиця 5

Сформований опитувальник для представників команди розробки мобільного застосунку

Номер питання	Питання
1	2
Метод з використанням гейміфікації	
П1	Яка складність реалізації функціоналу отримання балів та винагород у вигляді досягнень?
П2	Яка складність реалізації функціоналу змагання користувачів між собою?
Метод з використанням інтерактивних вправ	
П3	Яка складність реалізації функціоналу інтерактивних вправ, що виконуються?
П4	Яка складність реалізації функціоналу виконання інтерактивних вправ різного рівня складності, різних видів та різного рівня цікавості?
Метод з використанням мультимедійних матеріалів	
П5	Яка складність реалізації функціоналу використання аудіо- та відеоматеріалів?
П6	Яка складність реалізації функціоналу перегляду навчальних матеріалів?
Метод з використанням персоналізації контенту	
П7	Яка складність реалізації функціоналу вибору виду завдань, складності і підходів до вивчення іноземних мов?
П8	Яка складність реалізації функціоналу персоналізації контенту?
Метод з використанням системи обговорень та спільноти	
П9	Яка складність реалізації функціоналу участі користувачів у форумах та обговореннях?
П10	Яка складність реалізації функціоналу обміну особистими повідомленнями між користувачами?

Кінець таблиці 5

1	2
Метод з використанням подкастів та аудіоуроків	
П11	Яка складність реалізації функціоналу прослуховування подкастів та аудіоуроків?
П12	Яка складність реалізації функціоналу прослуховування аудіоносіїв мови, якщо робиться акцент на інтонацію та вимову?
Метод з використанням мікронавчання	
П13	Яка складність реалізації функціоналу мікровправ для вивчення нових слів або граматичних правил?
П14	Яка складність реалізації функціоналу самостійного розподілу завдань, вправ та задач?
Метод з використанням регулярних повторень	
П15	Яка складність реалізації функціоналу регулярних повторень навчального матеріалу?
П16	Яка складність реалізації функціоналу закріплення вивченого матеріалу?
Метод з використанням тестування та зворотного зв'язку	
П17	Яка складність реалізації функціоналу надання доступу до тестування окремим користувачам?
П18	Яка складність реалізації функціоналу зворотного зв'язку?
Метод з використанням User Research	
П19	Яка складність реалізації функціоналу залучення користувачів до досліджень та опитувань?
П20	Яка складність реалізації функціоналу, якщо в процесі розробки враховуються потреби та думки користувачів?

5.2. Результати апробації адаптованого методу експертного оцінювання в процесі розробки UI мобільного застосунку для вивчення мов

З огляду на досліджувану предметну область, пов'язану з необхідністю і можливістю вивчення мов, UI відповідного мобільного застосунку може включати такі елементи, як відокремлені секції для різних мов, можливість вибору рівня складності, інтерактивні вправи та тестування, а також зручний механізм відстеження прогресу користувача. Тому під час вирішення задачі визначення та планування дій відносно необхідних методів розробки Mobile UI вдовольнитися вибором лише одного з методів розробки інтерфейсу стає принципово неможливим, оскільки жоден з цих методів не може забезпечити сам по собі реалізацію усієї сукупності потреб та вимог, які висуваються до Mobile UI цього застосунку.

Експертне оцінювання методів розробки Mobile UI застосунку для вивчення мов проводилося на етапі вибору і розробки концепції застосунку, при проведенні мозкового штурму та розробці ескізів. Узагальнена експертна оцінка формувалася з використанням середніх арифметичних рангів та медіан рангів за формулами (1)-(4). Для формування даних з експертного оцінювання було використано опитувальники користувачів та команди розробників, наведені в табл. 4 і табл. 5 відповідно.

Опитування для користувачів проходили 10 незалежних користувачів. Опитування для команди розробників проходили 5 експертів, серед яких були дизайнер, менеджер,

бізнес-аналітик, бекенд-розробник та фронтенд-розробник.

Результати опитування користувачів наведено в табл. 6.

Таблиця 6

Результати опитування користувачів мобільного застосунку з вивчення мов

Номер експерта	Метод з використанням гейміфікації		Метод з використанням інтерактивних вправ		Метод з використанням мультимедійних матеріалів		Метод з використанням персоналізації контенту		Метод з використанням системи обговорень та спільноти	
	П1	П2	П3	П4	П5	П6	П7	П8	П9	П10
1	9	9	9	9	2	2	9	8	1	3
2	10	8	10	10	3	4	9	10	3	3
3	10	8	9	8	3	1	8	9	5	5
4	8	9	9	8	2	3	9	9	5	2
5	9	9	9	9	5	4	10	9	3	1
6	8	10	9	10	1	3	8	10	2	4
7	10	9	10	9	3	5	10	9	3	1
8	10	9	9	8	2	4	10	8	2	3
9	9	10	8	9	5	2	10	9	4	5
10	8	9	8	10	5	1	9	10	2	1
Номер експерта	Метод з використанням подкастів та аудіоуроків		Метод з використанням мікронавчання		Метод з використанням регулярних повторень		Метод з використанням тестування та зворотного зв'язку		Метод з використанням User Research	
	П11	П12	П13	П14	П15	П16	П17	П18	П19	П20
1	9	9	9	8	10	9	4	3	10	8
2	10	10	10	9	8	10	5	2	10	9
3	9	8	10	10	8	8	1	1	10	8
4	9	8	8	8	10	9	5	4	8	9
5	9	9	8	9	9	9	1	4	8	9
6	9	10	10	8	8	9	3	5	8	10
7	10	8	8	9	10	10	3	1	8	9
8	9	8	9	8	8	9	2	4	8	9
9	8	9	9	8	9	10	5	2	10	8
10	8	10	10	8	10	9	5	5	9	10

Для детального відображення оцінок також представлено діаграму (див. рис. 1) з результатами опитування користувачів з кожного питання. На діаграмі представлено інформацію про питання, методи та загальну суму оцінок з кожного питання, які поставили експерти.

Результати обчислення узагальненої експертної оцінки з використанням середніх арифметичних рангів досліджуваних методів при опитуванні користувачів надані в табл. 7.

Результати обчислення узагальненої експертної оцінки методом медіан рангів досліджуваних методів надаються в табл. 8.

Рейтинг методів розробки інтерфейсу, сформований з використанням експертного оцінювання для користувачів, представлений у табл. 9.

Базуючись на даних, наведених у табл. 6 та табл. 7, було перевірено узгодженість ранжованих рядів з використанням коефіцієнта конкордації. Результати перевірки представлено в табл. 10.

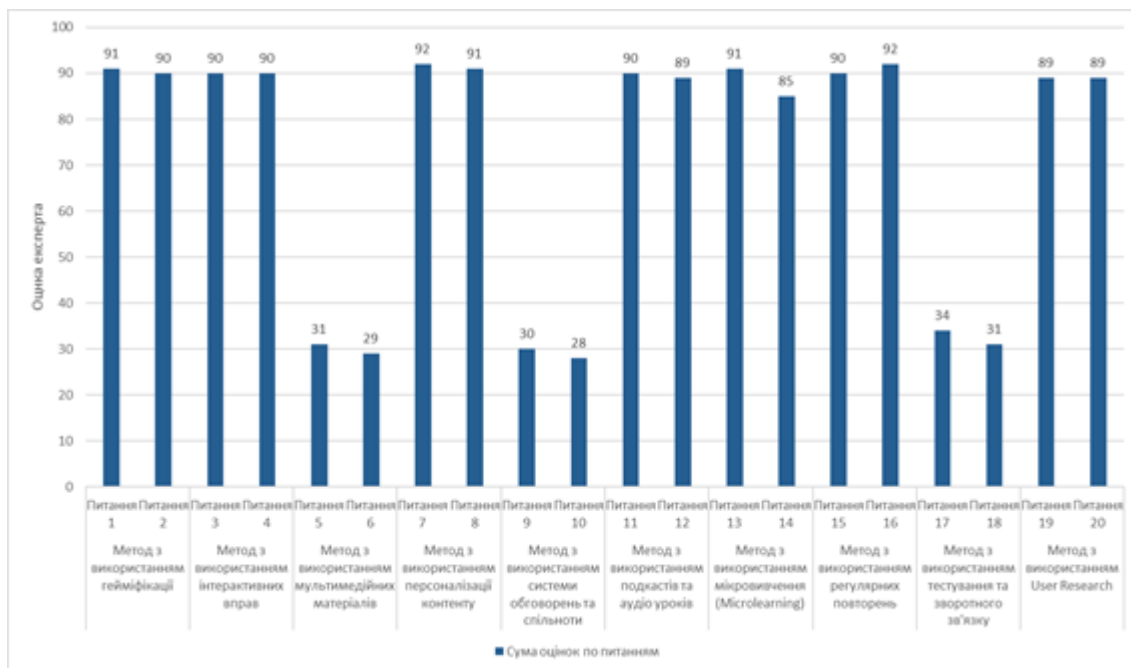


Рис. 1. Результати опитування користувачів з кожного питання

Таблиця 7

Результати обчислення узагальноної експертної оцінки з використанням середніх арифметичних рангів досліджуваних методів (для оцінок користувачів)

Досліджувані методи та питання	Метод з використанням гейміфікації		Метод з використанням інтерактивних вправ		Метод з використанням мультимедійних матеріалів		Метод з використанням персоналізації контенту		Метод з використанням системи обговорень та спільноти	
	П1	П2	П3	П4	П5	П6	П7	П8	П9	П10
1	2	3	4	5	6	7	8	9	10	11
Сума рангів питань	91	90	90	90	31	29	92	91	30	28
Сума рангів методів	90,5		90		30		91,5		29	
Середній арифметичний ранг	9,05		9		3		9,15		2,9	
Підсумковий ранг	3		4		9		1		10	
Досліджувані методи та питання	Метод з використанням подкастів та аудіоуроків		Метод з використанням мікронавчання		Метод з використанням регулярних повторень		Метод з використанням тестування та зворотного зв'язку		Метод з використанням User Research	
	П11	П12	П13	П14	П15	П16	П17	П18	П19	П20
Сума рангів питань	90	89	91	85	90	92	34	31	89	89

Кінець таблиці 7

1	2	3	4	5	6	7	8	9	10	11
Сума рангів методів	89,5		88		91		32,5		89	
Середній арифметичний ранг	8,95		8,8		9,1		3,25		8,9	
Підсумковий ранг	5		7		2		8		6	

Таблиця 8

Результати обчислення узагальненої експертної оцінки методом медіан рангів досліджуваних методів (для оцінок користувачів)

Досліджувані методи та питання	Метод з використанням гейміфікації		Метод з використанням інтерактивних вправ		Метод з використанням мультимедійних матеріалів		Метод з використанням персоналізації контенту		Метод з використанням системи обговорень та спільноти	
	П1	П2	П3	П4	П5	П6	П7	П8	П9	П10
1	2	3	4	5	6	7	8	9	10	11
Медіана рангів питань	9,1	9	9	9	3,1	2,9	9,2	9,1	3	2,8
Медіана рангів за методом	9,05		9		3		9,15		2,9	
Підсумковий ранг	3		4		9		1		10	
Досліджувані методи та питання	Метод з використанням подкастів та аудіоуроків		Метод з використанням мікронавчання		Метод з використанням регулярних повторень		Метод з використанням тестування та зворотного зв'язку		Метод з використанням User Research	
	П11	П12	П13	П14	П15	П16	П17	П18	П19	П20
Медіана рангів питань	9	8,9	9,1	8,5	9	9,2	3,4	3,1	8,9	8,9
Медіана рангів за методом	8,95		8,8		9,1		3,25		8,9	
Підсумковий ранг	5		7		2		8		6	

Середня сума рангів становить 72,1 та сума квадрата відхилення становить 7431,9. Значення коефіцієнта конкордації $W \approx 0,901$ є близьким до 1, що вказує на узгодженість експертних оцінок.

Результати опитування команди розробників наведено в табл. 11.

Для детального відображення оцінок також представлено діаграму (див. рис. 2) з результатами опитування команди розробників з кожного питання.

Таблиця 9

Рейтинг методів розробки інтерфейсу, сформований з використанням експертного оцінювання для користувачів

№ у рейтингу	Назва методу
1	Метод з використанням персоналізації контенту;
2	Метод з використанням регулярних повторень;
3	Метод з використанням гейміфікації;
4	Метод з використанням інтерактивних вправ;
5	Метод з використанням подкастів та аудіоуроків;
6	Метод з використанням User Research;
7	Метод з використанням мікронавчання;
8	Метод з використанням тестування та зворотного зв'язку;
9	Метод з використанням мультимедійних матеріалів;
10	Метод з використанням системи обговорень та спільноти.

Таблиця 10

Результати розрахунків коефіцієнта конкордації при опитуванні користувачів

Досліджувані методи та питання	Метод з використанням гейміфікації		Метод з використанням інтерактивних вправ		Метод з використанням мультимедійних матеріалів		Метод з використанням персоналізації контенту		Метод з використанням системи обговорень та спільноти	
	П1	П2	П3	П4	П5	П6	П7	П8	П9	П10
1	2	3	4	5	6	7	8	9	10	11
Сума рангів питань	91	90	90	90	31	29	92	91	30	28
Сума рангів методів	90,5		90		30		91,5		29	
Середній арифметичний ранг	9,05		9		3		9,15		2,9	
Відхилення від середньої суми	18,4		17,9		-42,1		19,4		-43,1	
Квадрат відхилення	338,56		320,41		1772,41		376,36		1857,61	
Досліджувані методи та питання	Метод з використанням подкастів та аудіоуроків		Метод з використанням мікронавчання		Метод з використанням регулярних повторень		Метод з використанням тестування та зворотного зв'язку		Метод з використанням User Research	
	П11	П12	П13	П14	П15	П16	П17	П18	П19	П20
Сума рангів питань	90	89	91	85	90	92	34	31	89	89
Сума рангів методів	89,5		88		91		32,5		89	

Кінець таблиці 10

1	2	3	4	5	6	7	8	9	10	11
Середній арифметичний ранг	8,95		8,8		9,1		3,25		8,9	
Відхилення від середньої суми	17,4		15,9		18,9		-39,6		16,9	
Квадрат відхилення	302,76		252,81		357,21		1568,16		285,61	

Таблиця 11

Результати опитування команди розробників мобільного застосунку з вивчення мов

Номер експерта	Метод з використанням гейміфікації		Метод з використанням інтерактивних вправ		Метод з використанням мультимедійних матеріалів		Метод з використанням персоналізації контенту		Метод з використанням системи обговорень та спільноти	
	П1	П2	П3	П4	П5	П6	П7	П8	П9	П10
1	9	10	9	9	9	9	9	9	5	5
2	8	9	8	8	9	8	9	10	6	6
3	8	10	8	9	8	9	8	9	6	3
4	10	10	10	9	10	8	9	9	4	2
5	9	9	8	10	9	10	9	10	6	1
Номер експерта	Метод з використанням подкастів та аудіоуроків		Метод з використанням мікронавчання		Метод з використанням регулярних повторень		Метод з використанням тестування та зворотного зв'язку		Метод з використанням User Research	
	П11	П12	П13	П14	П15	П16	П17	П18	П19	П20
1	9	9	9	8	10	9	4	3	10	8
2	10	10	10	9	8	10	5	2	10	9
3	9	8	10	10	8	8	1	1	10	8
4	9	8	8	8	10	9	5	4	8	9
5	9	9	8	9	9	9	1	4	8	9

Результати обчислення узагальненої експертної оцінки з використанням середніх арифметичних рангів досліджуваних методів при опитуванні команди розробників наведено в табл. 12.

Результати обчислення узагальненої експертної оцінки методом медіан рангів методів наведено в табл. 13.

Рейтинг методів розробки інтерфейсу, сформований з використанням експертного оцінювання для команди розробників, наведено у табл. 14.

Базуючись на даних, наведених у табл. 11 та табл. 12, було перевірено узгодженість ранжованих рядів з використанням коефіцієнта конкордації. Результати перевірки представлені в табл. 15.

Значення коефіцієнта конкордації $W \approx 0,551$ є достатньо близьким до 1, що вказує на можливість визнання експертних оцінок узгодженими.

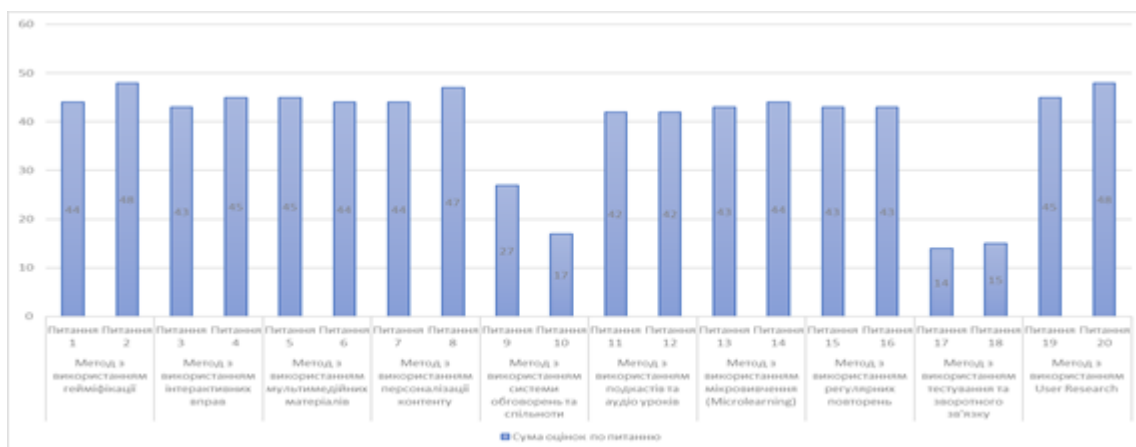


Рис. 2. Результати опитування команди розробників з кожного питання

Таблиця 12

Результати обчислення узагальноної експертної оцінки з використанням середніх арифметичних рангів досліджуваних методів (для оцінок команди розробників)

Досліджувані методи та питання	Метод з використанням гейміфікації		Метод з використанням інтерактивних вправ		Метод з використанням мультимедійних матеріалів		Метод з використанням персоналізації контенту		Метод з використанням системи обговорень та спільноти	
	П1	П2	П3	П4	П5	П6	П7	П8	П9	П10
Сума рангів питань	91	90	90	90	31	29	92	91	30	28
Сума рангів методів	46		44		44,5		45,5		22	
Середній арифметичний ранг	4,6		4,4		4,45		4,55		2,2	
Підсумковий ранг	2		5		4		3		9	
Досліджувані методи та питання	Метод з використанням подкастів та аудіоуроків		Метод з використанням мікронавчання		Метод з використанням регулярних повторень		Метод з використанням тестування та зворотного зв'язку		Метод з використанням User Research	
	П11	П12	П13	П14	П15	П16	П17	П18	П19	П20
Сума рангів питань	90	89	91	85	90	92	34	31	89	89
Сума рангів методів	42		43,5		43		14,5		46,5	
Середній арифметичний ранг	4,2		4,35		4,3		1,45		4,65	
Підсумковий ранг	8		6		7		10		1	

Таблиця 13

Результати обчислення узагальненої експертної оцінки методом медіан рангів досліджуваних методів (для оцінок команди розробників)

Досліджувані методи та питання	Метод з використанням гейміфікації		Метод з використанням інтерактивних вправ		Метод з використанням мультимедійних матеріалів		Метод з використанням персоналізації контенту		Метод з використанням системи обговорень та спільноти	
	П1	П2	П3	П4	П5	П6	П7	П8	П9	П10
Медіана рангів питань	8,8	9,6	8,6	9	9	8,8	8,8	9,4	5,4	3,4
Досліджувані методи та питання	Метод з використанням гейміфікації		Метод з використанням інтерактивних вправ		Метод з використанням мультимедійних матеріалів		Метод з використанням персоналізації контенту		Метод з використанням системи обговорень та спільноти	
	П1	П2	П3	П4	П5	П6	П7	П8	П9	П10
Медіана рангів за методом	9,2		8,8		8,9		9,1		4,4	
Підсумковий ранг	2		5		4		3		9	
Досліджувані методи та питання	Метод з використанням подкастів та аудіоуроків		Метод з використанням мікронавчання		Метод з використанням регулярних повторень		Метод з використанням тестування та зворотного зв'язку		Метод з використанням User Research	
	П11	П12	П13	П14	П15	П16	П17	П18	П19	П20
Медіана рангів питань	8,4	8,4	8,6	8,8	8,6	8,6	2,8	3	9	9,6
Медіана рангів за методом	8,4		8,7		8,6		2,9		9,3	
Підсумковий ранг	8		6		7		10		1	

Результати вирішення задачі визначення та планування дій відносно необхідних методів розробки Mobile UI для мобільного застосунку з вивчення мов наведено у табл. 16. Ці результати представлені у вигляді підсумкового рейтингу методів розробки інтерфейсу, в якому враховано отримані результати визначення рейтингу цих методів з точок зору користувачів та команди розробників.

За результатами, наведеними у табл. 16, для використання у процесі розробки Mobile UI слід рекомендувати:

Таблиця 14

Рейтинг методів розробки інтерфейсу, сформований з використанням експертного оцінювання для команди розробників

№ у рейтингу	Назва методу
1	Метод з використанням User Research;
2	Метод з використанням гейміфікації;
3	Метод з використанням персоналізації контенту;
4	Метод з використанням мультимедійних матеріалів;
5	Метод з використанням інтерактивних вправ;
6	Метод з використанням мікронавчання;
7	Метод з використанням регулярних повторень;
8	Метод з використанням подкастів та аудіоуроків;
9	Метод з використанням системи обговорень та спільноти;
10	Метод з використанням тестування та зворотного зв'язку.

Таблиця 15

Результати розрахунків коефіцієнта конкордації при опитуванні команди розробників

Досліджувані методи та питання	Метод з використанням гейміфікації		Метод з використанням інтерактивних вправ		Метод з використанням мультимедійних матеріалів		Метод з використанням персоналізації контенту		Метод з використанням системи обговорень та спільноти	
	П1	П2	П3	П4	П5	П6	П7	П8	П9	П10
Сума рангів питань	44	48	43	45	45	44	44	47	27	17
Сума рангів методів	90,5		90		30		91,5		29	
Досліджувані методи та питання	Метод з використанням гейміфікації		Метод з використанням інтерактивних вправ		Метод з використанням мультимедійних матеріалів		Метод з використанням персоналізації контенту		Метод з використанням системи обговорень та спільноти	
	П1	П2	П3	П4	П5	П6	П7	П8	П9	П10
Середній арифметичний ранг	46		44		44,5		45,5		22	
Відхилення від середньої суми	6,85		4,85		5,35		6,35		-17,15	
Квадрат відхилення	46,9225		23,5225		28,6225		40,3225		294,1225	

Кінець табл. 15

Дослід- жувані методи та питання	Метод з використанням подкастів та аудіоуроків		Метод з використанням мікронавчання		Метод з використанням регулярних повторень		Метод з використанням тестування та зворотного зв'язку		Метод з використанням User Research	
	П11	П12	П13	П14	П15	П16	П17	П18	П19	П20
Сума рангів питань	42	42	43	44	43	43	14	15	45	48
Сума рангів методів	89,5		88		91		32,5		89	
Середній арифме- тичний ранг	42		43,5		43		14,5		46,5	
Відхи- лення від середньої суми	2,85		4,35		3,85		-24,65		7,35	
Квадрат відхи- лення	8,1225		18,9225		14,8225		607,6225		54,0225	

Таблиця 16

Результати визначення підсумкового рейтингу методів розробки інтерфейсу мобільного застосунку з вивчення мов з точок зору користувачів та команди розробників

Назва методу	Рейтинг користува- чів	Рейтинг команди розробників	Сума рейтингів користувачів та команди розробників	Підсумковий рейтинг методу
Метод з використанням персоналізації контенту	1	3	4	1
Метод з використанням гейміфікації	3	2	5	2
Метод з використанням User Research	6	1	7	3
Метод з використанням регулярних повторень	2	7	9	4-5
Метод з використанням інтерактивних вправ	4	5	9	4-5
Метод з використанням подкастів та аудіоуроків	5	8	13	6-8
Метод з використанням мікронавчання	7	6	13	6-8
Метод з використанням мультимедійних матеріалів	9	4	13	6-8
Метод з використанням тестування та зворотного зв'язку	8	10	18	9
Метод з використанням системи обговорень та спільноти	10	9	19	10

а) як підмножину методів розробки інтерфейсу, що викликають найбільше задоволення усіх стейкхолдерів, – метод з використанням персоналізації контенту та метод з використанням гейміфікації;

б) як додаткову підмножину методів розробки інтерфейсу, які слід задіяти у випадках, коли застосування методів з першої підмножини не дає бажаного ефекту, – метод з використанням User Research, метод з використанням регулярних повторень та метод з використанням інтерактивних вправ.

6. Обговорення результатів дослідження

Адаптований до особливостей вирішення задачі визначення та планування дій відносно необхідних методів розробки Mobile UI метод експертного оцінювання дозволяє визначити кількісні оцінки доцільності застосування в процесі розробки Mobile UI окремих методів розробки інтерфейсу. На відміну від існуючих методів вирішення цієї задачі, метод експертного оцінювання дозволяє враховувати оцінки зазначених методів як з боку користувачів мобільного застосунку, так і з боку представників команди розробника цього застосунку. Ця відмінність дає змогу враховувати в процесі розробки мобільного застосунку рекомендації не тільки команди розробників, а й інших стейкхолдерів відповідного IT-проєкту, що дає змогу покращити вирішення задачі визначення та планування дій відносно необхідних методів розробки Mobile UI в умовах використання для управління IT-проєктами Agile та гібридних методологій.

Використання запропонованого у дослідженні методу експертного оцінювання для вирішення задачі формування і класифікації RFC дає змогу автоматизувати вирішення відповідних задач під час виконання діяльностей, пов'язаних з підготовкою до виконання процесів життєвого циклу мобільного застосунку. Відповідна інформаційна технологія є, значною мірою, універсальною, оскільки для різних процесів та діяльностей будуть розрізнятися тільки питання опитувальників, а не алгоритми та моделі обробки зібраних даних.

Як головні недоліки отриманих результатів дослідження треба вказати:

а) необхідність задіяти для обробки отриманих результатів опитування спеціальну групу фахівців, відволікаючи їх від виконання інших робіт IT-проєкту;

б) відсутність рекомендацій з визначення, яким саме методом формування узагальненої оцінки користуватися краще;

в) відсутність рекомендацій з пошуку компромісного рішення задачі, яке максимально можливою мірою вдовольнило б усіх стейкхолдерів IT-проєкту.

Виходячи з цих недоліків, головним напрямом подальших досліджень в галузі управління підготовкою до виконання процесів життєвого циклу IT-продукту слід визнати дослідження можливостей застосування моделей і методів пошуку компромісних рішень з вибору обмежених підмножин інструментів виконання діяльностей процесів життєвого циклу за результатами кількісного оцінювання існуючих альтернатив. Дослідження цих можливостей дозволить автоматизувати вирішення задачі визначення та планування дій відносно необхідних забезпечуючих систем та послуг, використовуючи відомий механізм отримання так званих «фідбеків» від усіх стейкхолдерів відповідного IT-проєкту.

7. Висновки

Під час виконання даного дослідження було отримано рішення задачі визначення та планування дій відносно необхідних методів розробки Mobile UI для мобільного застосунку. Основою цього рішення є метод експертного оцінювання, адаптований до особливостей порівняння та вибору підмножини методів розробки інтерфейсу, які рекомендується використовувати під час розробки Mobile UI. В процесі адаптації було розроблено опитувальники для користувачів та команди розробки мобільного застосунку

як основних стейкхолдерів IT-проєкту створення подібного застосунку.

Адаптований метод експертного оцінювання було апробовано в процесі вирішення задачі визначення та планування дій відносно необхідних методів розробки Mobile UI для мобільного застосунку з вивчення мов. Результати апробації показали, що найкращими для усіх стейкхолдерів цього IT-проєкту є метод з використанням персоналізації контенту та метод з використанням гейміфікації. У випадках, коли ці методи можуть не дати очікувані результати, обмежену підмножину рекомендованих методів запропоновано розширити шляхом додавання до неї методу з використанням User Research, методу з використанням регулярних повторень та методу з використанням інтерактивних вправ.

Результати дослідження дозволяють отримувати рішення задачі визначення та планування дій відносно необхідних методів розробки Mobile UI для IT-проєктів розробки мобільних застосунків за умови, що управління цими проєктами здійснюється на основі Agile та гібридних методологій.

Перелік посилань:

1. DSTU ISO/IEC/IEEE 15288:2016 Інженерія систем і програмного забезпечення. Процеси життєвого циклу систем (ISO/IEC/IEEE 15288:2015, IDT). [Чинний від 2018-10-01]. Вид. офіц. Київ: ДП «УкрНДНЦ», 2018. 80 с.
2. Настанова до Зводу знань з управління проєктами. Настанова РМВОК. Сьоме видання. Стандарт з управління проєктами. Project Management Institute, Inc., 14 Campus Boulevard Newtown Square, Pennsylvania 19073-3299 USA, 2021. 370 с.
3. Мобільні інформаційні системи [Електронний ресурс]: опорн. консп. лекцій / уклад. Н. Г. Яцків. Тернопіль : THEU, 2016. 60 с. URL: <http://dspace.wunu.edu.ua/handle/316497/25381>
4. Clark J. Designing for Touch. 2015. 169 p.
5. Nalwaya A., Paul A. React Native for Mobile Development: Harness the Power of React Native to Create Stunning iOS and Android Applications. APress, 2019. 237 p.
6. Мобільний веб-додаток, який тип обрати для розробки. *WEB(CASE)*. URL: <https://webcase.com.ua/uk/blog/cho-takoe-mobilnoe-web-prilozhenie/> (дата звернення: 28.09.2024).
7. Raskin J. The Humane Interface: New Directions for Designing Interactive Systems. Addison-Wesley Professional. 2000. 233 p.
8. 12 best language exchange apps and websites in 2024. *Preply Blog*. URL: <https://preply.com/en/blog/language-exchange-app/> (дата звернення: 28.09.2024).
9. Блинова Н.М., Кирилова О.В., Долженко М.В. Дидактичний потенціал мобільних застосунків для вивчення англійської мови як іноземної. Вісник університету ім. Альфреда Нобеля. Серія «Педагогіка і психологія». Педагогічні науки, 2023, №1 (25). С. 184-193.
10. User interface це не просто прикраса продукту. *FoxmindEd*. URL: <https://foxminded.ua/user-interface-tse/> (дата звернення: 28.09.2024).
11. Васильцова Н.В., Бурковська А.С. Аналіз структури та методів побудови інтерфейсів мобільних застосунків для користувачів веб-сайтів вивчення мов. *Proceedings of the XV International Scientific and Practical Conference «Distance education as the main problem of young people»* (December 26 - 29, 2023) Madrid, Spain: International Science Group. 2023. P. 293-302. URL: <https://isg-konf.com/distance-education-as-the-main-problem-of-young-people/>
12. Інтерфейс мобільних додатків: ключові принципи, критерії оцінки та помилки. *Wezom*. URL: <https://wezom.com.ua/ua/blog/dizajn-interfejsov-mobilnyh-prilozhenij> (дата звернення: 28.09.2024).
13. Microlearning: як просто та ефективно організувати навчання для команди. *Cases Media*. 2023. URL: <https://cases.media/article/microlearning-yak-prosto-ta-efektivno-organizuvati-navchannya-dlya-komandi> (дата звернення: 28.09.2024).
14. The Power of Customer Feedback in Improving Website Performance. *SurveySensum*. 2023. URL: <https://www.surveysensum.com/blog/power-of-customer-feedback> (дата звернення: 28.09.2024).
15. Common UI Elements and How to Use them in Mobile App Development. *CodeCoda*. 2021. URL: <https://codecoda.com/en/blog/entry/common-ui-elements-and-how-to-use-them-in-mobile-app-development> (дата звернення: 28.09.2024).
16. Mobile App Architecture & HoW To Start Building One. *Intellectsoft*. 2023. URL: <https://www.intellectsoft.net/blog/mobile-app-architecture/> (дата звернення: 28.09.2024).
17. In-App Survey Design: Tips, Best Practices And Great Examples From SaaS. *Userpilot Blog*. 2022. URL: <https://userpilot.com/blog/in-app-survey-design-best-practices/> (дата звернення: 28.09.2024).

18. Gamification in Education: Benefits, Risks, Examples. *KeenEthics*. 2023. URL: <https://keenethics.com/blog/simulation-gamification-of-education> (дата звернення: 28.09.2024).
19. Paired Comparison (Methods, Examples, Tools). *OpinionX*. 2020. URL: <https://www.opinionx.co/research-method-guides/paired-comparison> (дата звернення: 28.09.2024).
20. Пономарьов А.В. Встановлення узгодженості результатів експертного оцінювання / А.В. Пономарьов, науковий керівник – доцент Козлов Ю.В. *Радиоелектроніка і молодь у XXI столітті : матеріали 25-го Міжнародного форуму, 20-22 квітня 2021 р.* Харків : ХНУРЕ, 2021. Т. 4. С. 165–166.
21. Labijak-Kowalska A., Kadziński M. Experimental comparison of results provided by ranking methods in Data Envelopment Analysis. *Expert Systems with Applications*. 2021. Vol. 173, № 114739. DOI: <https://doi.org/10.1016/j.eswa.2021.114739>

Надійшла до редколегії 24.06.2024 р.

Васильцова Наталія Володимирівна, кандидат технічних наук, доцент, професор кафедри ІУС ХНУРЕ, м. Харків, Україна, e-mail: natalia.vasylytsova@nure.ua, ORCID: <https://orcid.org/0000-0002-4043-487X> (науковий керівник здобувачки вищої освіти Бурковської Анни Сергіївни).

Бурковська Анна Сергіївна, здобувачка вищої освіти, група ІУСТМ-22-1, факультет комп'ютерних наук, ХНУРЕ, м. Харків, Україна, e-mail: anna.burkovska@nure.ua

УДК 004.8:004.9

DOI: 10.30837/0135-1710.2024.182.066

С.Ф. ЧАЛИЙ, І.О. ЛЕЩИНСЬКА

УТОЧНЕННЯ МЕНТАЛЬНОЇ МОДЕЛІ РІШЕННЯ НА ОСНОВІ ДОПОВНЕННЯ ВХІДНИХ ДАНИХ В ЗАДАЧІ ФОРМУВАННЯ ПОЯСНЕНЬ В ІНТЕЛЕКТУАЛЬНІЙ СИСТЕМІ

Предметом дослідження є процес побудови ментальної моделі рішення для користувача інтелектуальної системи. Метою є розробка підходу до уточнення ментальної моделі користувача на основі принципу доповнення вхідних даних для того, щоб врахувати як позитивні, так і негативні аспекти рішення. Структуровано принцип доповнення вхідних даних з урахуванням особливостей формування пояснень в інтелектуальних системах. Запропоновано метод уточнення ментальної моделі рішення на основі доповнення вхідних даних. Метод включає етапи виявлення властивостей рішення, оцінки їхньої важливості, доповнення моделі та балансування позитивних і негативних аспектів рішення. Метод орієнтований на побудову пояснень, які враховують як переваги, так і обмеження рішення, що підвищує довіру користувачів до рішень інтелектуальних систем.

1. Вступ

Інтелектуальні інформаційні системи (ІС) використовують методи машинного навчання для аналізу даних та прийняття рішень. Проте через складність і непрозорість алгоритмів машинного навчання користувачі не завжди можуть зрозуміти, яким чином система дійшла кінцевого рішення. Це може привести до зниження довіри до ІС [1]. Для вирішення даної проблеми в галузі штучного інтелекту активно розвивається науковий напрямок пояснювального штучного інтелекту (Explainable Artificial Intelligence – ХАІ), орієнтований на розробку моделей і методів, що забезпечують прозорість та зрозумілість рішень для користувачів [2].

Однією із задач, що вирішуються в рамках побудови пояснень у ХАІ, є узгодження пояснення з ментальною моделлю користувача. Ментальна модель – це відображення зовнішньої реальності в свідомості людини. Таке відображення допомагає людині зрозуміти та інтерпретувати навколишні події [3]. У контексті ІС ментальна модель має відображати вхідні дані та рішення системи таким чином, щоб користувач міг їх коректно інтерпретувати та зрозуміти, як їх використовувати для вирішення практичних задач у

своїй професійній сфері діяльності.

Проте існуючі підходи до побудови ментальних моделей часто не враховують специфічні потреби зовнішніх користувачів ІС. Зокрема, вони не завжди відображають обмеження або негативні аспекти рішення, які можуть бути важливими для користувача. Така невизначеність може призводити до неповних або некоректних висновків при використанні інтелектуальної системи [4]. Тому розробка ментальних моделей, що враховують негативні аспекти рішення ІС, є ключовою умовою для створення зрозумілих пояснень в інтелектуальних системах.

2. Аналіз літературних даних і постановка проблеми дослідження

Напрямок ХАІ побудови пояснень у системах штучного інтелекту був започаткований у агентстві DARPA. ХАІ орієнтований на розробку методів, які дають можливість розуміти причини рішення ІС з тим, щоб підвищити довіру користувачів до таких систем [1]. Дослідження в галузі ХАІ зосереджені на створенні інтерпретованих моделей, розробці методів пост-пояснень та побудові інтерфейсів пояснень [2].

У напрямку досліджень ХАІ розроблено ряд підходів до побудови пояснень щодо рішень інтелектуальних систем. Метод контрфактуальних пояснень [3] забезпечує для користувача можливість отримати відповідь на питання: «Що потрібно змінити у вхідних даних, щоб отримати інше рішення?». Метод відображає альтернативні сценарії, що допомагає користувачам оцінити вплив різних значень вхідних даних на рішення ІС. Однак контрфактуальні пояснення часто не враховують негативні аспекти рішення.

Метод LIME (Local Interpretable Model-agnostic Explanations) [5] формує локальну інтерпретовану модель формування рішення для конкретного прикладу. Проте LIME фокусується лише на локальних аспектах рішення, що не дає можливості відобразити загальні закономірності побудови рішення, в тому числі обмеження на вхідні дані.

Метод SHAP (SHapley Additive exPlanations) використовує теорію ігор для розподілу важливості кожної ознаки у прогнозі моделі [6]. Метод забезпечує узгоджені пояснення на локальному рівні. Проте SHAP може бути обчислювально складним для великих моделей.

Метод пошарового розподілу релевантності (LRP) [7] розподіляє релевантність від виходу моделі до її вхідних даних. Перевагою LRP є можливість отримання детальних пояснень про внесок кожного шару мережі у кінцевий результат. Однак LRP не завжди враховує негативні аспекти рішення або його обмеження.

Каузальні методи побудови пояснень з можливістю безпосередньої інтерпретації користувачем було розроблено у [8]-[10]. Можливість безпосередньої інтерпретації каузальних залежностей було показано в [11].

Проте в розглянутих методах не приділяється достатньо уваги узгодженню пояснень із ментальними моделями рішень, які використовують користувачі ІС.

Ментальні моделі відіграють ключову роль у розумінні та інтерпретації рішень інтелектуальних систем користувачами [3]. Основи теорії ментальних моделей було закладено в [12], [13]. Ментальні моделі було визначено як внутрішні репрезентації зовнішнього світу, які люди використовують для розуміння, міркування та прогнозування. Принципи побудови ментальних моделей, розроблені в цих роботах, було розширено принципом доповнення вхідних даних в [14].

Роль ментальних моделей у взаємодії людини з технологіями проаналізовано в [15]. В даній роботі було обґрунтовано важливість розуміння ментальних моделей користувачів для проєктування інтуїтивно зрозумілих інтерфейсів. Особливості представлення ментальної моделі для побудови пояснень в ІС розглянуто в [16].

Існуючі методи побудови ментальних моделей, такі як метод структурного відображення [17] та метод каузального моделювання [18], можуть представляти

структуру та причинно-наслідкові зв'язки у рішеннях ІС. Однак ці методи не враховують обмеження або негативні аспекти рішення. Метод побудови ментальних моделей на основі спільного розуміння [19] розглядає ІС як члена команди людей, що створює умови для ефективної взаємодії між людиною та інтелектуальною системою. Однак побудова точної ментальної моделі ІС може бути складною для користувачів, особливо коли йдеться про врахування потенційних негативних аспектів рішень системи.

Аналіз існуючих методів побудови пояснень та ментальних моделей у ХАІ дає можливість зробити висновок про те, що в існуючих підходах, зазвичай, не приділяється достатньо уваги доповненню ментальної моделі відображенням негативних аспектів рішення ІС. Це обмеження може призвести до формування неповної або некоректної ментальної моделі.

Таким чином, актуальною є задача розробки методу уточнення ментальної моделі, який би враховував як позитивні, так і негативні аспекти рішень ІС. Вирішення даної задачі дає можливість отримати збалансоване представлення рішення, включаючи його потенційні обмеження та ризики. Формування узгодженого із потребами користувача рішення підвищує довіру до інтелектуальних систем та ефективність їх використання [20].

3. Мета і задачі дослідження

Метою даного дослідження є розробка підходу до уточнення ментальної моделі користувача інтелектуальної системи на основі принципу доповнення вхідних даних з тим, щоб враховувати не лише позитивні властивості рішення, але й можливі обмеження або негативні аспекти для підвищення рівня довіри до пояснень в інтелектуальних системах.

Для досягнення поставленої мети у роботі вирішуються такі задачі: структуризація принципу доповнення вхідних даних; розробка методу уточнення ментальної моделі користувача інтелектуальної системи на основі доповнення вхідних даних.

4. Принцип доповнення вхідних даних ментальної моделі користувача інтелектуальної системи

Принципи побудови ментальних моделей, що розглядалися в [12]-[14], включають в себе такі: відповідності структури ментальної моделі та рішення; множинності ментальних моделей, що дає можливість врахувати окремі варіанти рішення; неповноти даних для прийняття рішення; доповнення вхідних даних ментальної моделі.

Принцип доповнення вхідних даних ментальної моделі користувача інтелектуальної системи дає можливість користувачеві уточнити бажане рішення з урахуванням властивостей, які не були враховані в початковій ментальній моделі.

Користувачі можуть формувати неповні ментальні моделі рішень, враховуючи лише їхні позитивні (бажані) властивості. Такий підхід відображає когнітивні упередження користувачів та може призвести до прийняття нерациональних рішень внаслідок ігнорування потенційних обмежень. Наприклад, при виборі товару у системі e-commerce користувач може зосередитися на окремих технічних характеристиках товару, таких як розмір пам'яті або ціна, але не звернути увагу на такі важливі обмеження, як тривалість гарантії або існуючі проблеми з обслуговуванням.

Принцип доповнення вхідних даних спрямований на подолання цього обмеження шляхом включення додаткової інформації, яка може бути важливою для повного розуміння рішення. Згідно з цим принципом, необхідно провести додаткову оцінку рішення з урахуванням обмежень, що представляють значення властивостей, які раніше не були включені до ментальної моделі користувача.

Даний принцип доповнює принцип відповідності структури, забезпечуючи повніше відображення структури рішення в ментальній моделі. Принцип множинності ментальних моделей розширюється за рахунок включення альтернативних моделей, які враховують

різні комбінації позитивних та негативних аспектів рішення.

Принцип доповнення вхідних даних є комплементарним до принципу неповноти вхідних даних. Якщо останній принцип призводить до спрощення ментальної моделі шляхом виключення «неважливої» інформації, то принцип доповнення вхідних даних компенсує це спрощення, включаючи в модель важливі, але раніше проігноровані аспекти рішення.

У практичному плані застосування принципу доповнення вхідних даних може базуватися на аналізі відгуків інших користувачів інтелектуальної системи. Такі відгуки часто містять інформацію про властивості рішення, які не були очевидними або важливими для користувача на етапі початкового формування ментальної моделі.

5. Метод уточнення ментальної моделі користувача інтелектуальної системи на основі доповнення вхідних даних

Повна ментальна модель користувача M , згідно з принципами множинності, доповнення та неповноти вхідних даних, складається з множини локальних ментальних моделей M_i : $M = \{M_i\}$. Кожна з M_i використовується для визначеної підмножини вхідних даних. Наприклад, при використанні рекомендаційної підсистеми в системі електронної комерції повна ментальна модель смартфона може складатись з локальних ментальних моделей окремих моделей смартфонів, які рекомендовано користувачеві (або які переглядав користувач). Згідно з принципом неповноти вхідних даних, початкова модель M_i містить лише бажані для користувача властивості:

$$M_i = \{V_i^+\}. \quad (1)$$

Згідно з принципом доповнення вхідних даних, модель має характеризуватись неперетинними наборами позитивних V_i^+ та негативних V_i^- властивостей:

$$M_i = \{V_i^+, V_i^- \mid V_i^+ \cap V_i^- = \emptyset\}. \quad (2)$$

Ключова ідея методу уточнення ментальної моделі користувача полягає у визначенні та оцінці позитивних та негативних властивостей рішення ПС на основі аналізу відгуків користувачів такої системи. В подальшому негативні властивості, вага яких перевищує заданий поріг, включаються до складу ментальної моделі.

Метод уточнення ментальної моделі користувача містить такі етапи.

Етап 1. Аналіз відгуків користувачів щодо реалізованих рішень ПС.

Крок 1.1. Виявлення підмножини позитивних властивостей V_i^+ рішення.

Крок 1.2. Виявлення підмножини негативних властивостей V_i^- рішення.

Етап 2. Оцінка важливості властивостей рішення.

Крок 2.1. Для кожної властивості $v_{i,j}^+ \in V_i^+$ або $v_{i,j}^- \in V_i^-$ розрахунок ваги $w_{i,j}^+$ на основі кількості $n_{i,j}$ згадувань про цю властивість у відгуках. Розрахунок ваги i для позитивних, i для негативних властивостей має вигляд:

$$w_{i,j} = \frac{n_{i,j}}{|V_i^+ \cup V_i^-|}. \quad (3)$$

Таким чином, вага відображає частоту згадувань про властивість у відгуках

користувача.

Крок 2.2. Ранжування негативних властивостей за зменшенням ваги.

Етап 3. Доповнення ментальної моделі негативними властивостями рішення.

На даному етапі до ментальної моделі, крім позитивних властивостей V_i^+ рішення, включаються властивості з вагою, більшою порогового значення ε :

$$M_i = \{V_i^+, V_i^- \mid w_{i,j}^- > \varepsilon\}. \quad (4)$$

Етап 4. Балансування позитивних і негативних властивостей рішення.

Крок 4.1. Обчислення сумарних ваг $\sum_j w_{i,j}^+$ та $\sum_j w_{i,j}^-$ відповідно позитивних і негативних властивостей доповненої ментальної моделі.

Крок 4.2. Перевірка умови балансу ваг:

$$\sum_j w_{i,j}^+ > \sum_j w_{i,j}^-. \quad (5)$$

Дана умова визначає переваги рішення для користувача, тобто можливість використання рішення на практиці. Сенс принципу доповнення полягає в тому, щоб користувач враховував негативні властивості рішення, а не відкидав рішення повністю. Тому сумарна вага позитивних властивостей має переважати вагу негативних.

Крок 4.3. Якщо умова (5) не виконується, то збільшення порога на етапі 3, після чого – повторне виконання етапу 4.

Результатом застосування методу є ментальна модель користувача, що містить набори позитивних (бажаних для користувача) та найсуттєвіших негативних властивостей, вага яких перевищує порогове значення. Використання даної моделі для пояснень щодо рішення ПС дає можливість представити користувачеві не лише переваги рішення, а й можливі обмеження при його використанні.

6. Експериментальна перевірка методу доповнення ментальної моделі користувача інтелектуальної системи

Експериментальну перевірку методу орієнтовано на доповнення локальних ментальних моделей M_1 та M_2 двох смартфонів вищого цінового сегменту для однієї цільової аудиторії. Використано відгуки покупців цих смартфонів на платформі електронної комерції. Властивості базової ментальної моделі (без негативних властивостей) наведено в табл. 1.

Таблиця 1

Позитивні властивості двох моделей смартфонів M_1 та M_2

Властивість	Моделі	
	M_1	M_2
Кількість відгуків користувачів	231	66
Кількість позитивних властивостей	7	6
Сумарна вага позитивних властивостей	0,8398	0,7121

Ці моделі мають схожі набори позитивних властивостей: співпадають 5 базових властивостей і не співпадають 3 другорядні властивості.

Зважені негативні властивості смартфонів представлено в табл. 2. Вага враховує кількість відгуків за кожною позицією.

Таблиця 2

Зважені негативні властивості двох моделей смартфонів

Властивість	Вага	
	M_1	M_2
Вартість	0,0303	0,1060
Значне нагрівання корпусу	0,0389	–
Час (автономність) роботи	0,0346	–
Завеликі вага та розмір	0,0259	–
Не задовольняє якість звуку	0,0173	–
Проблеми з прийомом сигналу	0,0129	–
Відсутність або висока вартість аксесуарів	–	0,0757
Проблеми з автоматичними оновленнями	–	0,0606
Якість матеріалів корпусу	–	0,0454
Сумарна вага	0,1601	0,2878

Сумарна вага позитивних властивостей значно перевищує вагу негативних. Тому поріг $\varepsilon = 0$ і всі негативні властивості включені до складу обох ментальних моделей.

6. Обговорення результатів

Результати експериментальної перевірки показали, що ментальні моделі для схожих смартфонів мають аналогічні набори позитивних властивостей, але суттєво різні підмножини негативних властивостей. Тобто доповнення ментальних моделей негативними властивостями дає можливість обґрунтувати вибір конкретного рішення ПС в рамках пояснення.

Експериментальна перевірка показала, що пояснення має відповідати повній ментальній моделі та представляти позитивні й негативні властивості рішення за умови більшої ваги позитивних властивостей. В позитивному аспекті пояснення обґрунтовує відповідність вимогам користувача до рішення ПС, а в негативному – визначає можливість використання рішення при вирішенні задач користувача у відповідній предметній галузі.

7. Висновки

У дослідженні запропоновано підхід до удосконалення ментальної моделі рішення інтелектуальної системи шляхом урахування негативних властивостей даного рішення.

З метою обґрунтування запропонованого підходу виконано структурування принципу доповнення вхідних даних та визначено його зв'язки з іншими принципами побудови ментальних моделей, що визначають їхню структуру, множинність та неповноту.

Запропоновано метод доповнення вхідних даних ментальної моделі рішення інтелектуальної системи, що базується на принципі доповнення вхідних даних та включає етапи аналізу відгуків користувачів, оцінки важливості виявлених властивостей рішення, доповнення ментальної моделі з урахуванням негативних властивостей та балансування позитивних і негативних аспектів рішення. Метод дає можливість формувати пояснення, які враховують як переваги, так і обмеження рішення інтелектуальної системи, що створює умови для вилучення типових когнітивних упереджень, пов'язаних із ігноруванням негативних аспектів рішення. Експериментальна перевірка запропонованого методу показала можливість автоматизованого доповнення ментальної моделі рішення на основі аналізу відгуків користувачів інтелектуальної системи.

Перелік посилань:

1. Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI). IEEE Access, 6, 52138-52160. <https://doi.org/10.1109/ACCESS.2018.2870052>.

2. Gunning, D., & Aha, D. W. (2019). DARPA's Explainable Artificial Intelligence (XAI) program: A retrospective. *AI Magazine*, 40(2), 44-58. <https://doi.org/10.1609/aimag.v40i2.2850>.
3. Rook, L. (2021). Mental models: A robust definition. *The Learning Organization*, 28(1), 6-17. <https://doi.org/10.1108/TLO-09-2019-0136>.
4. Byrne, R. M. (2002). Mental models and counterfactual thoughts about what might have been. *Trends in Cognitive Sciences*, 6(10), 426-431. [https://doi.org/10.1016/S1364-6613\(02\)01974-5](https://doi.org/10.1016/S1364-6613(02)01974-5).
5. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). «Why should I trust you?»: Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135-1144.
6. Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30, 4765-4774.
7. Samek, W., Montavon, G., Lapuschkin, S., Anders, C. J., & Müller, K. R. (2022). Explaining deep neural networks and beyond: A review of methods and applications. *Proceedings of the IEEE*, 109(3), 247-278.
8. Чалий С. Ф., Лещинський В. О. Темпорально-каузальні методи побудови пояснень в системах штучного інтелекту. *АСУ та прилади автоматики*. Вип. 181. 2024. С. 91-99. DOI: 10.30837/0135-1710.2024.181.091.
9. Чалий С. Ф., Лещинський В. О. Побудова пояснень на локальному рівні представлення процесу формування рішень для внутрішніх користувачів інтелектуальної системи. *Біоніка інтелекту*, 2024. Т. 1 (100). С. 58-64. DOI 10.30837/bi.2024.1(100).08.
10. Чалий, С., Лещинський, В., & Лещинська, І. (2022). Каузальна модель процесу побудови пояснень в інформаційній системі. *Системи управління, навігації та зв'язку*, 3(69), 99-103. <https://doi.org/10.26906/SUNZ.2022.3.099>.
11. Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267, 1-38.
12. Johnson-Laird, P. N. (1983). *Mental models: Towards a cognitive science of language, inference, and consciousness*. Harvard University Press.
13. Johnson-Laird, P. N. (2006). *How we reason*. Oxford University Press/
14. Чалий, С. Ф., & Лещинська, І. О. (2024). Принципи побудови ментальних моделей рішення для зовнішнього користувача в задачі формування пояснень в інтелектуальній системі. *АСУ і прилади автоматики*, (181), 17-25.
15. Norman, D. A. (2014). Some Observations on Mental Models. In D. Gentner & A. L. Stevens (Eds.), *Mental Models* (pp. 7-14). Psychology Press.
16. Чалий, С., & Лещинська, І. (2023). Концептуальна ментальна модель пояснення в системі штучного інтелекту. *Вісник Національного технічного університету «ХПІ»*. Серія: Системний аналіз, управління та інформаційні технології, (1 (9)), 70-75. <https://doi.org/10.20998/2079-0023.2023.01.11>
17. Gentner, D., & Stevens, A. L. (2014). *Mental models*. Psychology Press.
18. Pearl, J., & Mackenzie, D. (2018). *The book of why: the new science of cause and effect*. Basic Books.
19. Bansal, G., Nushi, B., Kamar, E., Lasecki, W. S., Weld, D. S., & Horvitz, E. (2019). Beyond accuracy: The role of mental models in human-AI team performance. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, 7(1), 2-11.
20. Lipton, Z. C. (2018). The mythos of model interpretability. *Queue*, 16(3), 31-57.

Надійшла до редколегії 25.09.2024 р.

Чалий Сергій Федорович, доктор технічних наук, професор, професор кафедри ІУС ХНУРЕ, м. Харків, Україна, e-mail: serhii.chalyi@nure.ua; ORCID: 0000-0002-9982-9091
Лещинська Ірина Олександрівна, кандидат технічних наук, доцент, доцент кафедри програмної інженерії ХНУРЕ, м. Харків, Україна, e-mail: iryna.leshchynska@nure.ua, ORCID: <https://orcid.org/0000-0002-8737-4595>

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ВИЯВЛЕННЯ ТЕРМІНІВ ТА АРТЕФАКТІВ ПРОЄКТУ У ВИМОГАХ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ

Проаналізовано теоретичні і прикладні можливості застосування методів обробки природної мови для автоматизації елементів процесу «Визначення вимог до системи». Встановлено формальну основу побудови інформаційної технології виявлення термінів та артефактів проєкту у вимогах до інформаційної системи. Розроблено опис архітектури і технологічний стек, розглянуто особливості реалізації основних елементів цієї інформаційної технології. Проведено успішну експериментальну перевірку отриманих результатів.

1. Вступ

Управління ІТ-проєктами в сучасному динамічному технологічному середовищі є складним завданням, яке вимагає системного та структурованого підходу для досягнення успішних результатів. Для ефективного керування такими проєктами важливо мати чітке розуміння процесів, що складають життєві цикли проєкту та ІТ-продукту, який розглядається як результат цього проєкту. Одним із цих процесів є процес «Визначення вимог до системи», який відіграє ключову роль у розробці успішних ІТ-продуктів. Цей процес спрямований на систематичний збір та аналіз вимог до системи з метою точного визначення функціональних і нефункціональних характеристик, які система повинна втілювати [1]. Під системою тут і надалі будемо розуміти ІТ-продукти, описи яких повинні базуватися на системному підході.

Центральну роль у процесі «Визначення вимог до системи» відіграє діяльність «Визначення системних вимог». Під час цієї діяльності проводиться аналіз потреб користувачів та зацікавлених сторін щодо функціональності, продуктивності, безпеки та інших аспектів системи. Результатом цієї діяльності є визначення та обґрунтування системних вимог, яке охоплює визначення вимог до системи відповідно до вимог зацікавлених сторін, функціональних меж, функцій, обмежень, цільових показників вартості, визначених інтерфейсів і критичних характеристик якості [1].

Але виконання діяльності «Визначення системних вимог» ускладнюється тим, що вимоги зацікавлених сторін, які є початковою інформацією для цієї діяльності, є такими, що постійно змінюються під час планування та виконання ІТ-проєкту. Потрібно відзначити, що практика виконання ІТ-проєктів підтверджує, що внесення змін до ІТ-продуктів на ранніх етапах їхнього життєвого циклу є найефективнішим та економічно найдоцільнішим підходом. Це означає, що вирішення проблем та удосконалення системи на початковому етапі її розвитку вимагає менших витрат часу, грошей та ресурсів порівняно з такими ж діями на пізніших етапах проєкту [2]. Такий підхід свідчить про важливість ретельного аналізу визначення вимог до системи і важливість врахування потреб користувачів, технічних обмежень, а також вимог щодо забезпечення безпеки, надійності та інших якостей системи.

Для успішного вирішення проблеми внесення змін до вимог до системи на ранніх етапах її життєвого циклу необхідне розуміння цих вимог та особливостей описів архітектури і проєкту системи, засноване на попередньому досвіді. Таке розуміння дозволяє заздалегідь передбачити можливі складнощі та проблеми, які можуть виникнути під час реалізації актуальних вимог до системи або внесення змін до цих вимог.

Одним із способів використання попереднього досвіду у вирішенні цієї проблеми є

аналіз архітектурних особливостей існуючої системи, її архітектурних сутностей та їхніх зв'язків з іншими компонентами системи під час визначення вимог до системи. Таким способом можна ідентифікувати області системи, які можуть бути найвразливішими до змін, а також визначити потенційні ризики та виклики, пов'язані з цими змінами. Попередній досвід також дозволяє оцінити вплив запропонованих змін на вже існуючі компоненти системи та їхні функціональні можливості. Це допомагає уникнути неочікуваних негативних наслідків та забезпечити сумісність нових функцій з вже існуючими.

Але, хоча попередній досвід є важливим джерелом знань, він може також викликати упередженість, що може негативно впливати на процес прийняття рішень щодо створюваної або вдосконалюваної системи. У [3] показано, що більшість випадків негативного впливу локальних рішень на загальний дизайн і якість великої програмно-технічної системи усувалося шляхом раннього поділу системи на окремі елементи. Однак зворотні випадки виникали, головним чином, внаслідок неправильного тлумачення вимог або упередженості особистого досвіду [3]. Це дозволяє зробити висновок про доцільність використання для ідентифікації конфігурації великих систем та, зокрема, для визначення вимог до системи на основі вимог зацікавлених сторін людино-машинних або машинних методів, у яких суб'єктивний вплив окремого аналітика зведено до мінімуму. Такий підхід допомагає уникнути упередженості особистого досвіду та забезпечити об'єктивний та систематичний аналіз складнощів та потенційних проблем в процесі розробки та управління великими системами.

2. Аналіз сучасного стану теоретичних та прикладних робіт з аналізу системних вимог

2.1. Аналіз особливостей застосування методів штучного інтелекту в сучасних інструментальних засобах аналізу та управління вимогами

Одним з найперспективніших напрямів досліджень у галузі створення людино-машинних або машинних методів інженерії вимог є дослідження методів обробки природної мови (Natural Language Processing, NLP). Такі методи використовуються для аналізу, розуміння та генерації природної мови комп'ютерними системами, що може допомогти у виявленні та аналізі вимог користувачів, коментарів, відгуків та інших текстових джерел, які стосуються розробки та управління великими системами. Зокрема, ці методи можуть використовуватися для автоматизації процесу аналізу тексту з метою виявлення ключових сутностей, патернів або проблем, які можуть впливати на розвиток системи. Крім того, методи NLP можуть допомогти у створенні об'єктивних критеріїв та метрик для оцінки якості та ефективності системи, що дозволяє уникнути суб'єктивності та упередженості в процесі оцінки системних вимог та описів архітектури і дизайну системи. Як зазначається у [4], в середньому 60 % помилок у ІТ-проектах спричинені виникненням помилок на етапах визначення та аналізу вимог через неоднозначності або плутанину в критеріях прийняття цих вимог.

Як інструментальні засоби, що використовуються для аналізу та управління вимогами та засновані на методах NLP, у [4] особливо відзначають такі ІТ-продукти та інформаційні технології (ІТ):

- а) Visual Narrator;
- б) QuARS (Quality Analyzer for Requirements Specification);
- в) Qualicen Requirements Scout (QRS)
- г) Samantha;
- д) ReqSuite;
- е) IBM Engineering Requirements Quality Assistant;

ж) генеративний штучний інтелект (ШІ), такий як GPT-4 від OpenAI.

Visual Narrator є автоматизованим рішенням, що виводить концептуальну модель з вимог користувачів, використовуючи такі методи як POS-tagging для ідентифікації лінгвістичних патернів речень. Результатом є «доменна онтологія», яка формує модель, кількісно описуючи об'єкти та відносини між ними таким чином, що програмне забезпечення може їх інтерпретувати [5].

QuARS є методологією, призначеною для оцінки якості специфікацій вимог, та інструментом на основі цієї методології. Основна мета QuARS полягає у виявленні аспектів, де вимоги можуть бути неясними, непослідовними або неповними, що сприяє покращенню загальної якості вимог та зниженню ризику непорозуміння або помилок у процесі розробки. Серед особливостей QuARS можна виділити контрольні списки або керівництва для оцінки якості вимог, інструменти для автоматизованого аналізу документів вимог з метою ідентифікації потенційних проблем, а також панелі управління або звіти, які надають уявлення про якість вимог і виокремлюють області для покращення [6].

QRS є програмним інструментом, який надає рішення для інженерії та управління вимогами від моменту їх створення до остаточного впровадження системи. Він пропонує централізоване сховище для вимог, які можна організувати, відстежувати та контролювати протягом виконання IT-проєкту, забезпечуючи можливість відстеження вимог від їх початкового створення до остаточного впровадження. Таке відстеження гарантує, що вимогами належним чином управляють та що будь-які зміни контролюють та відстежують. Крім того, QRS включає функції аналізу вимог, такі як аналіз впливу, аналіз простежуваності та аналіз якості, які допомагають організаціям забезпечити повноту, послідовність та високу якість своїх вимог [7].

Інструментальний засіб Semantha призначений для порівняння документів на семантичному рівні. Це означає, що Semantha може розуміти ідеї та значення, що перевищують буквальний рівень окремих слів і фраз, дозволяючи виявляти спільні концепції в різних документах, порівнювати їх та виділяти відмінності. Ця здатність робить інструмент надзвичайно корисним для класифікації вимог, ідентифікації пов'язаних ризиків та документування порівняння на семантичному рівні [8].

Особливістю ReqSuite є функція допомоги, яка дозволяє автоматично виявляти кілька ключових проблем у вимогах: концептуальну неповноту (коли важливі вимоги повністю забуваються зацікавленими сторонами або коли вимоги передбачають додаткові вимоги, які відсутні), неточність вимог (коли не зрозуміло, чи є вимога обов'язковою чи опціональною) та суперечності між вимогами. Цей інструмент допомагає забезпечити повноту, точність та відсутність внутрішніх суперечностей для усієї множини вимог, що значно знижує ризик виникнення помилок у процесі розробки та впровадження проєктів. Використання ReqSuite також може спростити процес управління вимогами, забезпечуючи краще розуміння та узгодженість вимог усіма учасниками проєкту, зменшуючи час та витрати на перегляд і корекцію вимог [9].

IBM ERQA використовує передові можливості обробки природної мови для автоматизованого аналізу вимог. Це дозволяє виявляти потенційні двозначності та генерувати реальні оцінки в реальному часі для оцінки якості вимог. Система оцінювання базується на кількох концепціях, таких як складні вимоги, неточні дієслова, неповнота, відсутність обмежень, відсутні одиниці виміру, негативні твердження, незрозумілі займенники тощо. Інструмент дозволяє швидко ідентифікувати можливі проблеми з якістю та забезпечити чіткість та однозначність вимог перед початком розробки. Використання IBM ERQA може значно знизити ризики, пов'язані з неправильним тлумаченням вимог, та покращити якість кінцевого продукту, забезпечуючи ефективнішу співпрацю між усіма

учасниками проекту [10].

Генеративний ШІ, такий як GPT-4 від OpenAI, в контексті інженерії вимог може використовуватися для уточнення, ідентифікації потенційних конфліктів або проблем та надання рекомендацій для поліпшення якості вимог, а також генерування тестових випадків і підтримки управління та документації вимог [7].

На основі аналізу цих інструментів можна зробити висновок, що багато з них вирішують завдання аналізу та класифікації вимог, виявлення неоднозначностей та навіть деякою мірою валідації вимог щодо їхньої повноти та консистентності. Проте ці інструменти часто обмежуються рамками текстового аналізу та не здатні прямо аналізувати та інтегрувати виявлені сутності з існуючими системами або візуалізувати зв'язки між сутностями у формі графів. Можливість такої візуалізації може відіграти важливу роль у виявленні системних вимог [11], а її відсутність значно обмежує застосування розглянутих інструментів та ІТ для комплексного розуміння структур даних системи.

2.2. Аналіз особливостей методів обробки природної мови, які застосовуються для розпізнавання та класифікації термінів із текстових даних

Методи NLP охоплюють широкий спектр від базового синтаксичного аналізу до складних методів глибокого навчання, які здатні вловлювати семантичні зв'язки у тексті. Зокрема, методи NLP дозволяють автоматизувати процес розпізнавання та класифікації термінів із текстових даних, що може значно підвищити ефективність аналізу та управління вимогами.

Серед доступних методів класифікації термінів як ключових сутностей і використання їх для подальшого аналізу пропонується приділити особливу увагу методам стемінгу та лематизації. Стемінг є процесом видалення афіксів зі слів, щоб повернутися до кореня або базової форми слова [12]. Цей метод може бути корисним для спрощення текстових даних, але він може призвести до втрати значущої інформації через недоліки у врахуванні словоформ. Лематизація, у свою чергу, є складнішим процесом, що залучає морфологічний аналіз для перетворення слова на його базову форму або лему. Це процес групування різних словоформ одного слова, так щоб їх можна було аналізувати як єдиний елемент, який можна ідентифікувати за лемою слова або його словниковою формою [12].

Результати SWOT-аналізу методу стемінгу наведено у табл. 1.

Таблиця 1

Результати SWOT-аналізу методу стемінгу

	Корисні фактори	Шкідливі фактори
Внутрішні фактори	Сильні сторони	Слабкі сторони
	Швидкість обробки	Неточність Мовна залежність
Зовнішні фактори	Можливості	Загрози
	Розвиток алгоритмів Інтеграція з іншими методами NLP	Зміни в обробці природної мови Зростання вимог до якості обробки тексту Обмеження в обробці неструктурованих даних

Результати SWOT-аналізу методу лематизації наведено у табл. 2.

Результати SWOT-аналізу методу лематизації

	Корисні фактори	Шкідливі фактори
	Сильні сторони	Слабкі сторони
Внутрішні фактори	Точність Контекстна обробка Універсальність Підтримка алгоритмами різноманітних мов	Ресурсоємність Складність реалізації Залежність від мови
Зовнішні фактори	Можливості	Загрози
	Інтеграція з іншими NLP-інструментами; Покращення машинного навчання; Розширення застосування.	Зміни в обробці природної мови Обмеження в обробці неструктурованих даних

На основі проведеного SWOT-аналізу як метод виявлення сутностей у тексті пропонується обрати метод лематизації через його здатність забезпечувати високу точність аналізу. Цей метод враховує контекст слова, що дозволяє точно ідентифікувати базові форми слів, ця характеристика робить лематизацію особливо цінною для задач, де важлива точність розпізнавання сутностей. Хоча лематизація може вимагати більше обчислювальних ресурсів порівняно зі стемінгом та висувати вищі вимоги до реалізації, багато сучасних алгоритмів підтримують українську мову, що свідчить про її застосовність у широкому спектрі лінгвістичних середовищ.

2.3. Формулювання проблеми дослідження

На основі проведеного аналізу особливостей застосування методів штучного інтелекту в сучасних інструментальних засобах аналізу та управління вимогами та особливостей методів обробки природної мови, які застосовуються для розпізнавання та класифікації термінів із текстових даних, можна зробити такі висновки.

По-перше, існує велика потреба в інтеграції досвіду з попередніх проєктів та використання передових практик при визначенні системних вимог. Недостатня увага до досвіду минулих проєктів та відсутність систематичного підходу до збору та аналізу цього досвіду обмежує можливість врахування вже існуючих рішень та помилок. В сучасному контексті розробки та модифікації систем суб'єктивний вплив, заснований на попередньому досвіді, може ініціювати упередженість, що, у свою чергу, негативно позначається на процесі ухвалення рішень та прогресу систем.

По-друге, існуючі методи та підходи до визначення системних вимог часто стикаються з проблемою неоднозначності та невизначеності. Це призводить до труднощів у визначенні вимог до системи, що може спричинити помилки на ранніх етапах проєктування та збільшити витрати на внесення змін у майбутньому.

По-третє, на сьогодні існує дефіцит повноцінних теоретичних та практичних рішень, призначених для автоматизованої обробки вимог до систем. Особливо це стосується забезпечення глибокої інтеграції виявлених сутностей з існуючими системами та візуалізації їх для детального аналізу. Така ситуація ускладнює розробку та адаптацію ІТ-продуктів та, зокрема, інформаційних систем (ІС) управління підприємствами та організаціями, підвищуючи ризик неврахування важливих аспектів на ранніх етапах проєктування.

Дані висновки дозволяють припустити, що забезпечення успішного виконання процесу визначення системних вимог критично потребує розвитку і впровадження автоматизованих методів аналізу текстових даних для виявлення та розуміння системних вимог, особливо в контексті управління ІТ-проєктами. Цей процес також має включати

ефективне використання уроків, винесених з минулого досвіду, і водночас застосування сучасних аналітичних інструментів для уникнення упередженості.

Виділені проблеми є наслідком недостатніх теоретичних та практичних досліджень у сфері формування та аналізу вимог до ІС. Основні публікації зарубіжних дослідників [13], [14] переважно спрямовані на осмислення практичного досвіду у цій області. Проте в цих та інших роботах майже не розглянуто можливості використання автоматизованих інструментів для визначення та аналізу системних вимог.

Тому проблему даного дослідження пропонується визначити як проблему створення спеціалізованих ІТ визначення, аналізу та управління системними вимогами, які застосовують методи NLP для виявлення окремих термінів предметної області та створюваної ІС і аналізу можливості опису цих термінів через артефакти попередніх ІТ-проектів.

3. Мета і задачі дослідження

Метою даного дослідження є розробка ІТ виявлення термінів та артефактів проекту ІС. Досягнення цієї мети дозволить автоматизувати діяльність «Визначення системних вимог» і, зокрема, дозволить визначити можливість повторного використання артефактів системи в результаті встановлення їхньої відповідності виявленим під час визначення системних вимог термінам предметної галузі та створюваної системи.

Для досягнення цієї мети треба вирішити такі задачі:

- розробити опис архітектури та визначити технологічний стек для розробки ІТ виявлення термінів та артефактів проекту ІС;
- розробити елементи ІТ для виявлення та класифікації термінів та артефактів проекту ІС в текстових документах, з акцентом на групування відмінюваних форм;
- провести експериментальну перевірку отриманих результатів.

4. Матеріали і методи дослідження

Об'єктом дослідження є діяльність «Визначення системних вимог» процесу «Визначення вимог до системи» [1]. Предметом дослідження є ІТ виявлення термінів та артефактів проекту ІС.

Як основну гіпотезу даного дослідження запропоновано розглядати можливість формувати та зберігати у спеціалізованому репозиторії описи окремих елементів ІС як сукупностей структур даних із встановленими назвами кожної окремої структури. Застосування цієї гіпотези дозволяє представити задачу зіставлення виявлених у текстах системних вимог термінів предметної області та створюваної системи з існуючими артефактами ІС як задачу порівняння формальних представлень цих сутностей з назвами існуючих артефактів. Під артефактом в даному дослідженні будемо розуміти [15]:

- а) опис окремого елемента ІС на різних етапах його життєвого циклу;
- б) опис ІС в цілому як елементу більшої системи.

Для утворення формального опису термінів предметної області та створюваної системи, які присутні у текстових публікаціях вимог до системи, пропонується задіяти методи лематизації. Базою цих методів є [12]:

- статистичні моделі;
- правила на основі морфології;
- комбінація статистики та правил.

Статистична модель у контексті NLP є моделлю машинного навчання, яка навчається на великих корпусах тексту з метою вивчення зв'язків між словами та їхніми лемами. Ці моделі аналізують частоту вживання слів у різних контекстах та інші мовні ознаки, щоб визначити ймовірність того, що дане слово в певному контексті буде мати певну лему. Позначимо множину усіх слів у тексті як S , множину контекстів як C , множину усіх

можливих лем у мові як L . Тоді статистичні моделі визначають умовну ймовірність P того, що дане слово в певному контексті буде мати певну лему, як

$$P = f(l/w, c), \quad (1)$$

де $l \in L$ – лема слова $w \in S$ у контексті $c \in C$.

Спочатку статистична модель аналізує контекст кожного слова у текстовій публікації системної вимоги, порівнюючи цей контекст із статистичними зв'язками між словами та їхніми лемами. На основі цього аналізу модель визначає умовну ймовірність для кожної можливої базової форми слова у даному контексті. За допомогою цих умовних ймовірностей модель вибирає найімовірнішу базову форму для кожного слова у текстовій публікації системної вимоги.

Іншим способом визначення умовної ймовірності того, що дане слово в певному контексті буде мати певну лему, є застосування правил на основі морфології. Цей спосіб використовує правила лематизації, які базуються на морфологічних особливостях мови. Для цього способу вираз (1) матиме вигляд:

$$P = f(w, c, \{r_i\}), \quad (2)$$

де $\{r_i\}$ – множина морфологічних правил, які застосовуються для визначення лем (якщо $\{r_i\} = \emptyset$, то правила ігноруються).

Для розроблюваної ІТ виявлення термінів та артефактів проєкту ІС найкращим варіантом лематизації запропоновано визнати комбінацію статистики та правил. Ця комбінація поєднує в процесі вибору лем l для слова w у контексті c результати статистичного моделювання та застосування правил на основі морфології. Це означає, що умовна ймовірність вибору лем розраховується на основі статистичних даних. Правила на основі морфології застосовуються для вибору лем в тому випадку, якщо ці лем не враховуються статистичною моделлю, що використовується. Така комбінація формально може бути представлена як функція мети

$$l^* = \arg \max_{l \in L} f(l/w, c, \{r_i\}). \quad (3)$$

Таким чином, тепер можливо отримати у вигляді лем формальні представлення термінів, присутніх у текстових публікаціях системних вимог, для подальшого формального співставлення їх із термінами, присутніми в описах артефактів ІС.

Аналогічно можливо отримати у вигляді лем формальні представлення термінів, присутніх в описах артефактів ІС. Відмінність полягає в тому, що функцію (3) в цьому випадку було запропоновано застосовувати для аналізу описів структур даних, використаних в описах досліджуваних артефактів ІС.

Тоді задачу виявлення термінів та артефактів проєкту ІС пропонується розглядати як задачу пошуку такого відображення множини лем термінів текстової публікації системної вимоги $L_R^* = (l_j^*(R_i))$ у множину лем термінів, присутніх в описах артефактів ІС, призначених для повторного використання з метою реалізації системної вимоги $L_{Ar}^* = (l_j^*(Ar_x))$, яке в оптимальному випадку може бути зведене до повної еквівалентності, тобто до ситуації

$$(l_j^*(R_i)) = (l_j^*(Ar_x)), \quad (4)$$

де R_i – текстова публікація системної вимоги, $i=1,2,\dots,n$; n – кількість системних вимог,

які обробляються ІТ за один раз; $l_j^*(R_i)$ – j -та лема, визначена на текстовій публікації системної вимоги R_i , $j=1,2,\dots,m$; m – кількість лем, виявлених у текстовій публікації системної вимоги R_i ; Ar_x – опис x -го артефакта ІС, який пропонується для повторного використання у реалізації системної вимоги R_i , $x=1,2,\dots,z$; z – кількість артефактів, які пропонуються для повторного використання у реалізації системної вимоги R_i ; $l_j^*(Ar_x)$ – j -та лема, визначена на описі артефакта ІС Ar_x .

У випадку отримання раціонального рішення задачі виявлення термінів та артефактів проєкту ІС відображення (4) матиме вигляд

$$(l_{j=1,\dots,m}^*(R_i)) \geq (l_{j=1,\dots,k \leq m}^*(Ar_x)), \quad (5)$$

де k – кількість лем, виявлених у описі артефакта ІС Ar_x .

5. Результати дослідження

5.1. Розробка опису архітектури та визначення технологічного стека інформаційної технології

Розроблювану ІТ виявлення термінів та артефактів проєкту ІС було запропоновано представити як послідовність таких етапів і кроків.

Етап 1. Перетворення функціональних вимог, які висуваються до ІС, до початкової форми.

Крок 1.1 Формування та зберігання текстової публікації функціональної вимоги до ІС.

Крок 1.2. Токенізація текстової публікації функціональної вимоги до ІС.

Крок 1.3. Формування початкової форми текстової публікації функціональної вимоги до ІС шляхом лематизації множини токенів, отриманої в результаті виконання Кроку 1.2.

Етап 2. Перетворення описів артефактів ІС, придатних для повторного використання, до початкової форми.

Крок 2.1. Токенізація опису артефакту ІС, придатного для повторного використання.

Крок 2.2. Формування початкової форми опису артефакту ІС, придатного для повторного використання, шляхом лематизації множини токенів, отриманої в результаті виконання Кроку 2.1.

Крок 2.3. Оновлення множини початкових форм описів артефактів ІС, придатних для повторного використання.

Етап 3. Визначення відповідностей між початковими формами текстових публікацій функціональних вимог та описів артефактів ІС, придатних для повторного використання.

Крок 3.1. Визначення відповідності лем з початкової форми текстової публікації функціональної вимоги до ІС та лем з початкової форми опису артефакту ІС, придатного для повторного використання.

Крок 3.2. Відбір початкових форм описів артефактів ІС, придатних для повторного використання, відповідної функціональної вимоги за результатами виконання Кроку 3.1.

Крок 3.3. Візуалізація описів артефактів ІС, придатних для повторного використання, початкові форми яких було відібрано в результаті виконання Кроку 3.2. Завершення застосування ІТ.

Опис архітектури ІТ виявлення термінів та артефактів проєкту ІС формувався під впливом двох конкуруючих вимог:

а) необхідно інтегрувати всі необхідні компоненти обробки, аналізу та візуалізації даних у єдиний цілісний сервіс, що вимагає застосування монолітної архітектури;

б) з огляду на використання новітніх технологій, опис архітектури ІТ повинен бути

розроблений таким чином, щоб у майбутньому можна було б легко адаптувати його, наприклад, до мікросервісної моделі з метою підвищення масштабованості та ефективності обробки публікацій вимог, що надходять з великої кількості територіально розподілених джерел.

Додатковою вимогою, яку треба було враховувати під час розробки опису архітектури ІТ, є вимога можливості налаштування сервісу, який реалізує ІТ, для обробки публікацій вимог незалежно від мови користувача. У випадку, якщо вимоги опубліковані українською мовою, таке налаштування включатиме підключення до бази даних, де описи артефактів визначені українською мовою, а для обробки вимог завантажувється статична модель лематизації, відповідна до мови користувача.

Виходячи з цих та ряду інших вимог, в процесі створення опису архітектури ІТ виявлення термінів та артефактів проєкту ІС було запропоновано виділяти такі архітектурні сутності (компоненти) цієї ІТ:

а) компонент перетворення вимог до початкової форми, який забезпечує виконання токенизації текстової публікації вимоги, розбиваючи її на окремі слова, які, в свою чергу, служать вхідними даними для процесу лематизації;

б) компонент декорування схеми бази даних ІС, який забезпечує виконання токенизації та лематизації назв таблиць і полів, з яких складаються описи артефактів, придатних для повторного використання;

в) компонент створення діаграми зв'язків, який забезпечує встановлення відповідностей між лемами вимог користувача, отриманими з компонента перетворення вимог до початкової форми, та лемами опису артефактів, отриманими з компонента декорування схеми бази даних ІС, а також генерацію представлення встановлених відповідностей у вигляді візуальної діаграми для подальшого аналізу вимоги.

Оскільки основною забезпечуючою системою [1] для ІТ виявлення термінів та артефактів проєкту ІС є програмна система, подальша деталізація опису архітектури системи повинна враховувати результати формування технологічного стека цієї ІТ. Під технологічним стеком тут і далі будемо розуміти набір технологій, які використовуються разом для розробки та підтримки програмного забезпечення [16]. Зазвичай технологічний стек для програмної забезпечуючої системи складається з таких компонентів [16]:

– фронтенд (клієнтська частина): технології, що використовуються для створення інтерфейсу користувача;

– бекенд (серверна частина): технології для роботи з бізнес-логікою, базами даних та серверними процесами;

– база даних: системи для зберігання та управління даними;

– система управління версіями: інструменти для відстеження змін у коді;

– інструменти для розгортання та підтримки: технології для автоматизації розгортання та підтримки проєкту.

Технологічний стек ІТ виявлення термінів та артефактів проєкту ІС має такі складові:

а) для розробки загальних елементів сервісу, що повинен реалізувати ІТ, запропоновано використати мову програмування C# та кросплатформенну технологію .NET Core [17] від компанії Microsoft;

б) для розробки компонента перетворення вимог до початкової форми запропоновано використати:

– мову програмування Python та, зокрема, бібліотеку SpaCy [18], яка є ключовим інструментом надання розширених можливостей для лематизації та морфологічного аналізу, включаючи підтримку багатьох мов, включаючи українську мову;

– офіційну бібліотеку SpacyDotNet [19], яка дозволяє використовувати можливості SpaCy в середовищі .NET, зокрема в .NET Core, і базується на Python.NET для забезпечення

взаємодії між Python та .NET;

в) для розробки компонента декорування схеми бази даних ІС запропоновано використати:

– СУБД MS SQL Server;

– бібліотеку Microsoft.SqlServer.Management.Smo, можливості якої забезпечують вилучення та аналіз метаданих як елементів описів артефактів ІС, необхідних для кореляції із лематизованими токенами, отриманими з текстових публікацій системних вимог;

г) для розробки компонента створення діаграми зв'язків запропоновано використати інструмент PlantUML [20], який призначений для створення діаграм програмного забезпечення і дозволяє описувати структуру програми за допомогою текстового опису та автоматично генерувати візуальні діаграми на основі цього опису, використовуючи простий синтаксис, що базується на UML;

д) для розробки частини сервісу, яка забезпечить взаємодію користувача з цим сервісом та можливість подання текстових публікацій вимог до обробки, запропоновано використати фреймворк React [21], який був обраний за рахунок свого розширеного набору готових до використання стилізованих компонентів і, зокрема, бібліотеки компонентів BlueprintJs [22] для створення інтуїтивного інтерфейсу.

Цей технологічний стек був обраний з огляду на здатність до швидкої розробки, необхідності підтримки різних людських мов, а також легкості інтеграції різних компонентів у єдиний цілісний сервіс.

З врахуванням обраного технологічного стека, опис архітектури сервісу, який реалізує ІТ виявлення термінів та артефактів проекту ІС, було приведено до діаграми компонентів, наведеної на рис. 1.

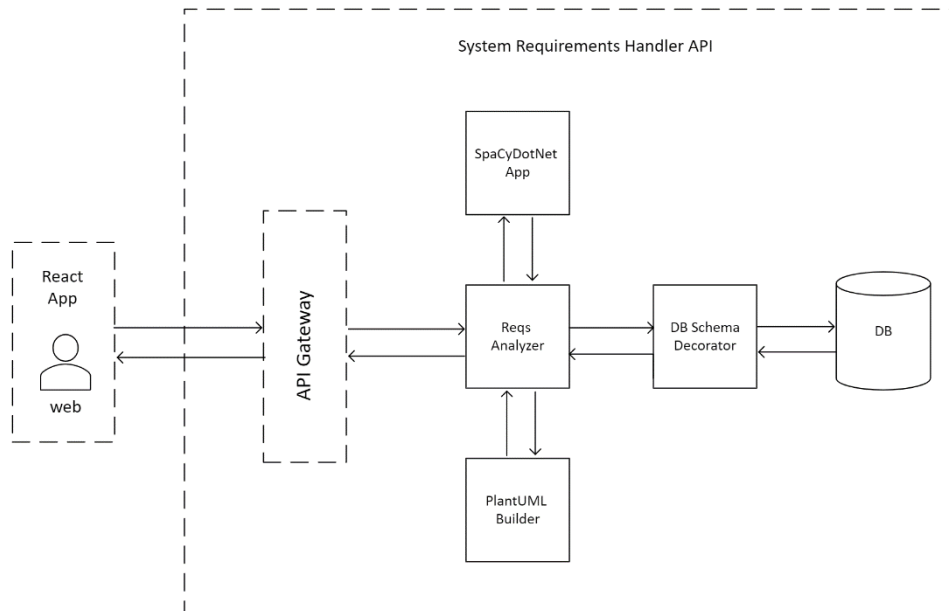


Рис. 1. Діаграма компонентів, яка описує архітектуру сервісу

На рис. 1 прийняті такі позначення:

– Reqs Analyzer – модуль, який забезпечує реалізацію компонента перетворення вимог до початкової форми та елементів компонента створення діаграми зв'язків, які відповідають за встановлення відповідностей між лемами вимог користувача та лемами опису артефактів;

- SpaCyDotNetApp – модуль, який забезпечує інтеграцію бібліотеки SpaCy у модуль Reqs Analyzer із використанням можливостей бібліотеки SpacyDotNet;
- DB Schema Decorator – модуль, який забезпечує реалізацію компонента декорування схеми бази даних IC;
- DB – сховище даних, яке забезпечує реалізацію репозиторія артефактів IC, придатних для повторного використання;
- PlantUML Builder – модуль, який забезпечує реалізацію елементів компонента створення діаграми зв'язків, які відповідають за генерацію представлення встановлених відповідностей у вигляді візуальної діаграми для подальшого аналізу вимоги;
- API Gateway – модуль, який реалізує шлюз для інтеграції серверної та клієнтської частин сервісу;
- React App – модуль, який реалізує частину сервісу на основі фреймворку React, яка забезпечує взаємодію користувача з цим сервісом та можливість подання текстових публікацій вимог до обробки.

5.2. Особливості реалізації елементів інформаційної технології автоматизованого виявлення та аналізу сутностей вимог до IC

Однією з основних вимог до IT виявлення термінів та артефактів проєкту IC є її здатність обробляти вимоги користувачів різними мовами, зокрема англійською та українською. Для задоволення цієї вимоги сервіс, який реалізує зазначену IT, було спроектовано з використанням гнучкої конфігурації API, що дозволяє легко інтегрувати підтримку додаткових мов.

Конфігураційний файл сервісу містить окремі секції для кожної мови, де визначаються параметри для обробки тексту, взаємодії зі сховищем даних, ідентифікації колонок та регулярних виразів для обробки тексту, що дозволяє адаптувати сервіс під конкретні мовні особливості. Моделі для обробки тексту використовують можливості бібліотеки SpaCy, що дозволяє користувачам обрати будь-яку мову, що підтримується цією

```

"DataSet": {
  "ua": {
    "LangModel": "uk_core_news_sm",
    "ConnectionDb": "Data Source=localhost;Initial Catalog=ФутбольнаЛіга;User ID=nure;",
    "ColumnIdentifierSuffix": "Код",
    "ColumnSplitRegex": "[А-ЯЁІіЄГ][^А-ЯЁІіЄГ]*"
  },
  "en": {
    "LangModel": "en_core_web_sm",
    "ConnectionDb": "Data Source=localhost;Initial Catalog=FootballLeague;User ID=nure;",
    "ColumnIdentifierSuffix": "Id",
    "ColumnSplitRegex": "[A-Z][^A-Z]*"
  }
}

```

Рис. 2. Фрагмент програмного коду, який описує конфігурацію мовних моделей інформаційної технології

бібліотекою. Для апробації налаштовано використання української мови за допомогою попередньо навченої моделі uk_core_news_sm та англійської мови за допомогою попередньо навченої моделі en_core_web_sm. Фрагмент програмного коду, який описує конфігурацію мовних моделей, наведено на рис. 2. Такі параметри, як

LangModel, ConnectionDb, ColumnIdentifierSuffix та ColumnSplitRegex, забезпечують не тільки адекватне розпізнавання та обробку мовних структур, але й належну взаємодію зі сховищем даних та адаптацію до специфічних мовних ідентифікаторів у даних.

В рамках реалізації модулів, зазначених на рис. 1, а саме ReqsAnalyzer для визначення лем з тексту вимог та DB Schema Decorator для визначення і декорації лемами описів артефактів IC, використовується метод nlp з бібліотеки SpaCy. Цей метод є частиною стандартного інтерфейсу обробки тексту в SpaCy.

```

C# Lang.cs x
13 public class Lang : IXmlSerializable
14
15     [1 usage] [luchytskyi]
16     public Doc GetDocument(string text)
17     {
18         using (Py.GIL())
19         {
20             var pyString = new PyString(text);
21             var doc:dynamic = PyLang.__call__(pyString);
22             return new Doc(doc, text);
23         }
24     }
25
26
27
28
29
30
31
32
33

```

Рис. 3. Фрагмент програмного коду, який описує клас Lang містить структуровану інформацію про оброблений текст і його властивості, такі як Text, Lemma, POS, Tag тощо.

Фрагмент програмного коду, який описує клас DbEntitiesLemmatizer, що декорує описи артефактів ІС лемами (зроблені на основі таблиць, їхніх назв і полів), наведений на рис. 4. DecorateEntitiesWithLemma – основний публічний метод цього класу, призначений для обробки колекцій таблиць. Він приймає колекцію об'єктів типу Microsoft.SqlServer.Management.Smo.TableCollection і повертає колекцію об'єктів типу ReqsTable, кожен з яких містить леми для відповідних елементів таблиці.

```

using System.Text.RegularExpressions;
using Microsoft.IdentityModel.Tokens;
using Microsoft.SqlServer.Management.Smo;
using ReqsHandler.Core.Configuration;
using ReqsHandler.Core.Services.Models;

namespace ReqsHandler.Core.Services;

public class DbEntitiesLemmatizer(ISpacyInstance spacyInstance, ICurrentContext context)
    : IDbEntitiesLemmatizer
{
    private Regex ColumnSplitRegex => new Regex(context.DataSet.ColumnSplitRegex);
    public IEnumerable<ReqsTable> DecorateEntitiesWithLemma(TableCollection collection)
    {
        var tables = new List<ReqsTable>();
        foreach (Table table in collection)
        {
            var isSplitName = SplitAndLemmatize(table.Name, out var splitList, out var lemmas);
            tables.Add(new ReqsTable
            {
                Name = table.Name,
                Lemmas = lemmas,
                IsSplitName = isSplitName,
                SplitNames = splitList,
                BaseEntity = table,
                Columns = MapColumnLemma(table.Columns)
            });
        }
        return tables;
    }
}

```

Рис. 4. Фрагмент програмного коду, який описує клас DbEntitiesLemmatizer

Оскільки в технологічному стеку було запропоновано використовувати бібліотеку SpacyDotNet для інтеграції функцій бібліотеки SpaCy у додатки, написані на C#, в сервісі було реалізовано клас Lang і його метод GetDocumnet, який повертає об'єкт Doc (рис. 3). Цей об'єкт

```

private bool SplitAndLemmatize(string name, out IList<string> splitList, out IEnumerable<string> lemmas)
{
    splitList = Array.Empty<string>();
    lemmas = Array.Empty<string>();
    if (name.IsNullOrEmpty())
    {
        return false;
    }

    var isSplitName = SplitIfNeed(name, out splitList);
    var doc = spacyInstance.GetDocument(string.Join(" ", splitList));
    lemmas = doc.Tokens.Where(t => !t.IsPunct).Select(t => t.Lemma.ToLower());

    return isSplitName;
}

private bool SplitIfNeed(string entityName, out IList<string> result)
{
    var isSplit = false;
    result = Array.Empty<string>();
    var parts = ColumnSplitRegex.Matches(CleanUpName(entityName));
    if (parts.Count > 1)
    {
        isSplit = true;
    }

    result = new List<string>();
    foreach (Match part in parts)
    {
        result.Add(part.Value);
    }
    return isSplit;
}

private IEnumerable<ReqsColumn> MapColumnLemma(ColumnCollection tableColumns)
{
    var columns = new List<ReqsColumn>();
    foreach (Column column in tableColumns)
    {
        var isSplitName = SplitAndLemmatize(column.Name, out var splitList, out var lemmas);
        columns.Add(new ReqsColumn
        {
            Name = column.Name,
            Lemmas = lemmas,
            BaseEntity = column,
            IsSplitName = isSplitName,
            SplitNames = splitList
        });
    }

    return columns;
}

private string CleanUpName(string name)
{
    return name.Replace(context.DataSet.ColumnIdentifierSuffix, "");
}
}

```

Рис. 4, аркуш 2

Метод `SplitAndLemmatize` призначений для визначення лем. В цьому методі реалізовано аналіз назв об'єктів описів артефактів ІС із застосуванням регулярних виразів, які використовуються в конфігурації (див. рис. 1) для визначення необхідності розбиття складних назв на окремі слова. За допомогою методу `SplitAndLemmatize` перш за все перевіряється, чи потрібно розділити назву на менші фрагменти, що значно покращує точність подальшої обробки вимог.

Треба також зазначити, що перед розбиттям або аналізом текст обробляється методом `CleanUpName`, який за допомогою зазначеної в конфігурації властивості `ColumnIdentifierSuffix` видаляє суфікс, яким зазвичай помічають в структурованих базах даних поле з ідентифікатором (наприклад, суфікс «Код» для української локалізації та «Id», відповідно, для англійської локалізації артефактів ІС).

Клас `PlantUmlBuilder` відіграє ключову роль у створенні коду для візуалізації діаграми за

```
C# PlantUmlBuilder.cs ×
8 public class PlantUmlBuilder(List<TableDto> tables)
    {
        [1] usage  ▾ Viacheslav Luchytskyi +1
28 private string BuildUml(IList<string> inputLemmas, HashSet<string> includedTables)
29     {
30         var diagram = new StringBuilder();
31         diagram.AppendLine("@startuml");
32         diagram.AppendLine(DefinedFunctions);
33         diagram.AppendLine();
34
35         // Definition of tables
36         var result:string = BuildTableEntities(inputLemmas, includedTables);
37         if (result.IsNullOrEmpty())
38         {
39             return string.Empty;
40         }
41
42         diagram.Append(result);
43
44         // Definition of relationships between tables
45         diagram.Append(BuildDependentTables(inputLemmas, includedTables));
46
47         diagram.AppendLine("@enduml");
48         return diagram.ToString();
49     }
    }
```

Рис. 5. Фрагмент програмного коду, який описує метод `BuildUml` класу `PlantUmlBuilder`

допомогою `PlantUML`. Фрагмент програмного коду з описом основного методу цього класу `BuildUml`, який створює текстове представлення діаграми, організовуючи таблиці та їхні взаємозв'язки, наведено на рис. 5. Особливість `PlantUmlBuilder` полягає в його здатності деталізувати кожен сутність за допомогою лем, отриманих з тексту вимог, що дозволяє відобразити можливі структурні і логічні зв'язки між елементами

досліджуваної ІС. Такий підхід робить зрозумілішими та кориснішими діаграми, які використовуються для аналізу і планування системних вимог.

Як зазначено у підрозд. 5.1, архітектура ІТ передбачає впровадження різних додаткових модулів. Для таких модулів доступ до функцій сервісу, який реалізує розроблену ІТ, здійснюється через REST API, реалізовані на платформі ASP.NET Core, що забезпечує високу продуктивність і масштабованість.

Для оптимізації процесів налагодження та документації кожного модуля було вирішено забезпечити доступ до API та впровадити використання Swagger [23], як це представлено на рис. 6. Використання цього інструментального засобу значно спрощує інтеграцію та перевірку взаємодії між модулями сервісу.

5.3. Опис експериментальної перевірки отриманих результатів

Для проведення експериментальної перевірки отриманих результатів було використано ІС «Футбол». Цю ІС було спроектовано спеціально для проведення експериментального дослідження, вона містить набір основних функцій, що дозволяють автоматизувати вирішення задач планування та обліку проведення футбольних матчів. Для перевірки можливостей обробки вимог, опублікованих різними мовами, було реалізовано українську та англійську версію цієї ІС. Як описи артефактів цієї ІС у експериментальних

дослідженнях розглядаються описи таблиць бази даних цієї ІС, полів цих таблиць та зв'язків між цими таблицями.

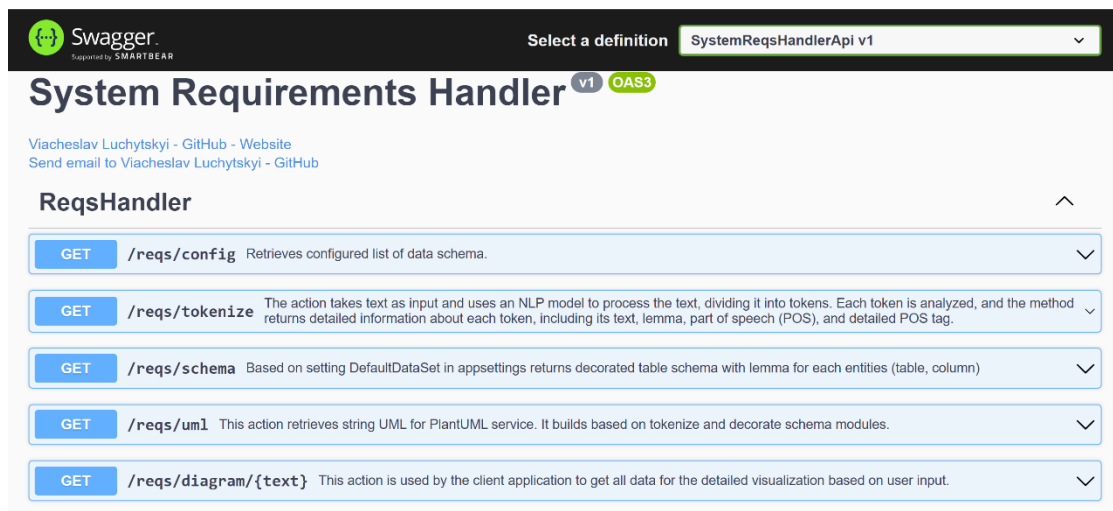


Рис. 6. Інтерфейс для тестування API модулів сервісу

План проведення експериментальної перевірки результатів розробки ІТ виявлення термінів та артефактів проекту ІС передбачав послідовне виконання таких етапів:

- а) етап 1: перевірка працездатності сервісу, який реалізує ІТ, на прикладі системної вимоги, опублікованої англійською мовою;
- б) етап 2: перевірка працездатності сервісу, який реалізує ІТ, на прикладі системної вимоги, опублікованої українською мовою;
- в) етап 3: перевірка релевантності результатів обробки системної вимоги, опублікованої українською та англійською мовами.

Під час виконання першого етапу експериментальної перевірки було використано системну вимогу, яка висувається до англійської локалізації ІС «Футбол». Текстова публікація цієї вимоги виглядає так: «Give the coach the opportunity to train several teams». Результати токенізації цієї вимоги з використанням можливостей бібліотеки SpaCy наведено на рис. 7. Фрагмент результату декорування лемами описів відповідних артефактів ІС, що були реалізовані у попередніх версіях ІС «Футбол» та можуть бути повторно використані для проектування та реалізації вказаної вище вимоги, наведено на рис. 8. Результат виділення сукупності артефактів ІС, яка може бути повторно використана для проектування та реалізації зазначеної вище системної вимоги, наведено на рис. 9. Візуальна діаграма, яку бачить користувач сервісу як результат застосування ІТ, наведена на рис. 10.

Під час виконання другого етапу експериментальної перевірки було використано системну вимогу, яка висувається до української локалізації ІС «Футбол». Текстова публікація цієї вимоги виглядає так: «Додати можливість для гравців виступати за декілька команд». Візуальну діаграму, яку бачить користувач сервісу як результат застосування ІТ для обробки цієї системної вимоги, наведено на рис. 11.

Під час виконання третього етапу експериментальної перевірки текстову публікацію системної вимоги, яка була використана на другому етапі, було перекладено англійською мовою. Результатом перекладу став текст «Add ability for players to participate for several team». Візуальну діаграму, яку бачить користувач сервісу як результат застосування ІТ для обробки цього англійського перекладу системної вимоги, наведено на рис. 12.


```

</> Give the coach the opportunity to train several teams
UML TOKENS DATA
|Text |Lemma|Pos |Tag
|give |give |VERB |VB
|the |the |DET |DT
|coach|coach|NOUN |NN
|the |the |DET |DT
|opportunity|opportunity|NOUN |NN
|to |to |PART |TO
|train|train|VERB |VB
|several|several|ADD |JJ
|teams|team |NOUN |NNS

```

Рис. 7. Результати токенизації опублікованої системної вимоги англійською мовою

```

</> Give the coach the opportunity to train several teams
UML TOKENS DATA
}
{
  "name": "Matches",
  "lemmas": [
    "match"
  ],
  "foreignKeyDto": [
    {
      "name": "FK_Match_Home_Team",
      "referencedTable": "Teams",
      "referencedTableSchema": "dbo",
      "columns": [
        {
          "name": "HomeTeamId",
          "referencedColumn": "Teams"
        }
      ]
    }
  ]
}

```

Рис. 8. Результати декорування лемами описів артефактів англійської локалізації інформаційної системи

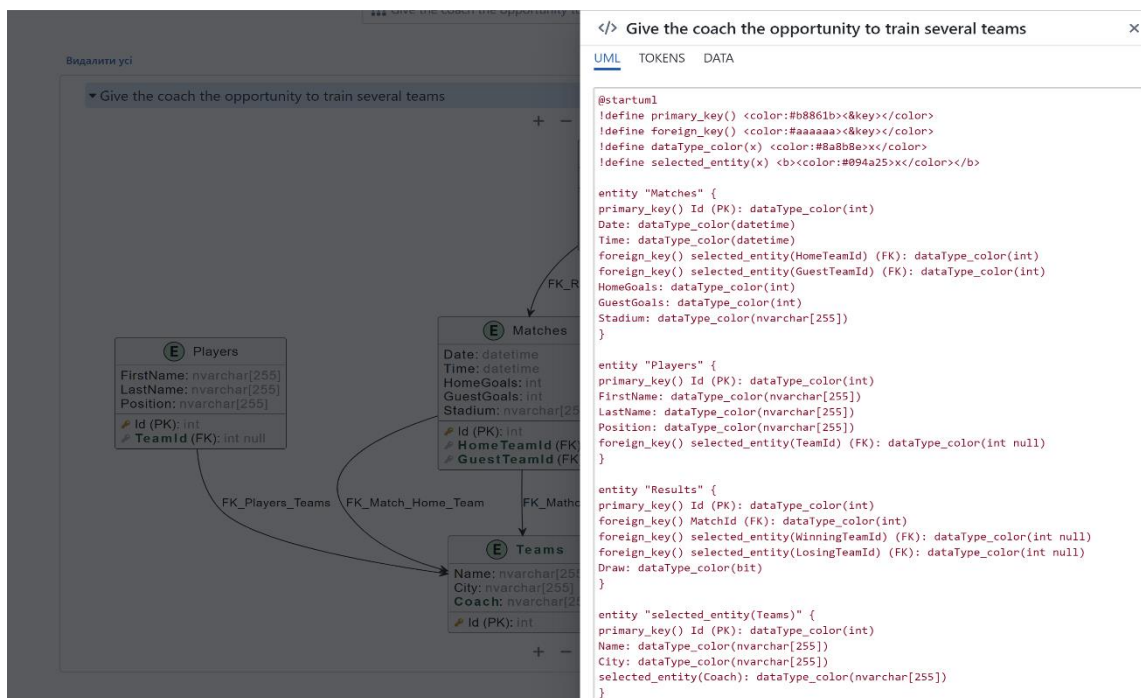


Рис. 9. Результат виділення сукупності артефактів інформаційної системи, яка може бути повторно використана для проєктування та реалізації опублікованої системної вимоги

Порівняння результатів обробки, наведених на рис. 11 та рис. 12, підтверджує релевантність отриманих результатів.

6. Обговорення результатів дослідження

За результатами проведення експериментальної перевірки розробленої ІТ виявлення термінів та артефактів проєкту ІС слід зробити такі висновки:

- а) сервіс, який реалізує розроблену ІТ, дозволяє визначати терміни системних вимог до ІС «Футбол», тексти яких опубліковано англійською та українською мовами;
- б) сервіс, який реалізує розроблену ІТ, дозволяє визначати терміни, що використовуються в описах артефактів ІС, які можуть бути повторно застосовані для реалізації опублікованих системних вимог у англійській та українській локалізаціях ІС «Футбол»;

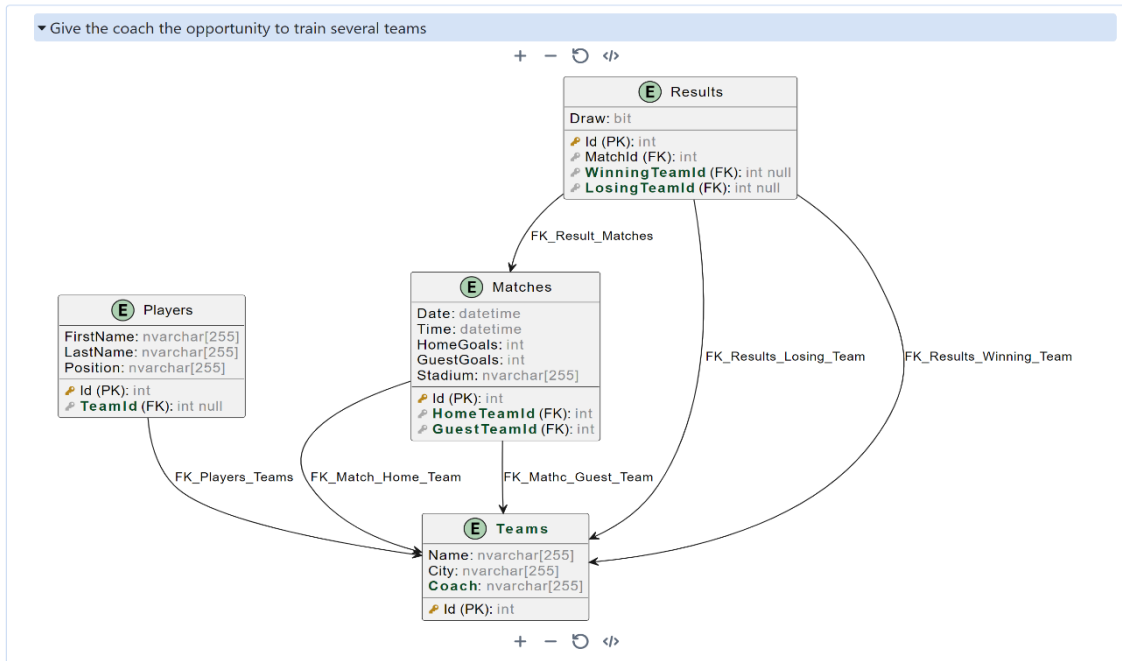


Рис. 10. Візуальна діаграма, яку бачить користувач сервісу як результат обробки опублікованої системної вимоги

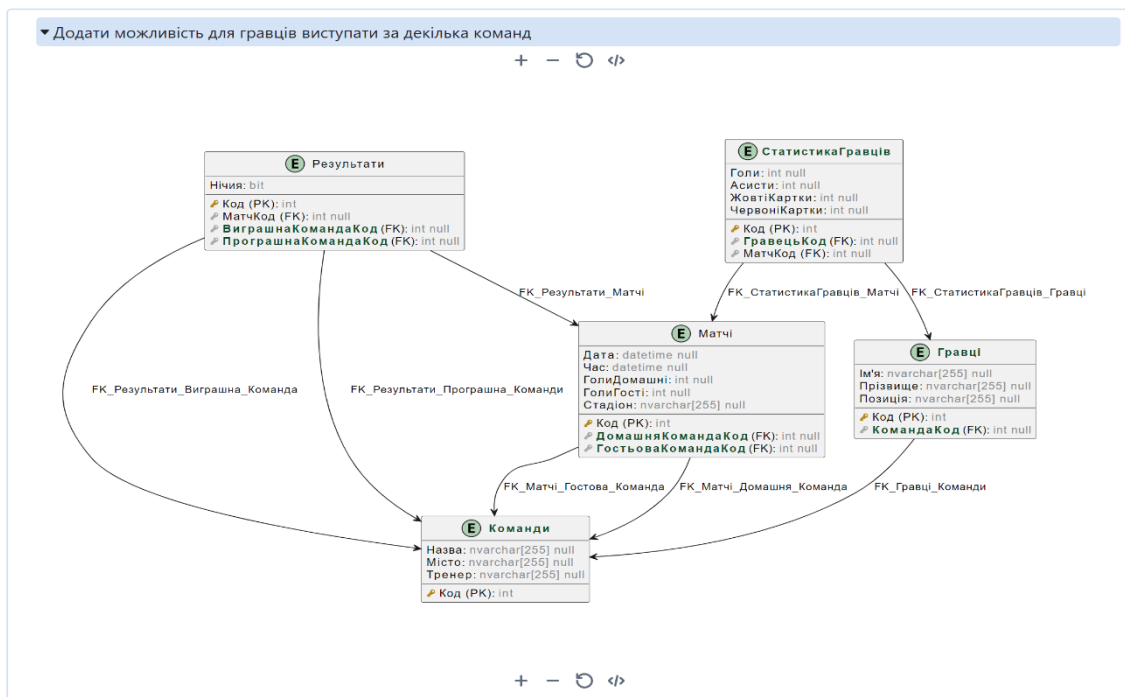


Рис. 11. Візуальна діаграма, яку бачить користувач сервісу як результат обробки системної вимоги, опублікованої українською мовою

в) сервіс, який реалізує розроблену ІТ, дозволяє візуалізувати підмножину артефактів ІС, терміни яких співпадають із термінами опублікованої системної вимоги, у вигляді візуальної діаграми;

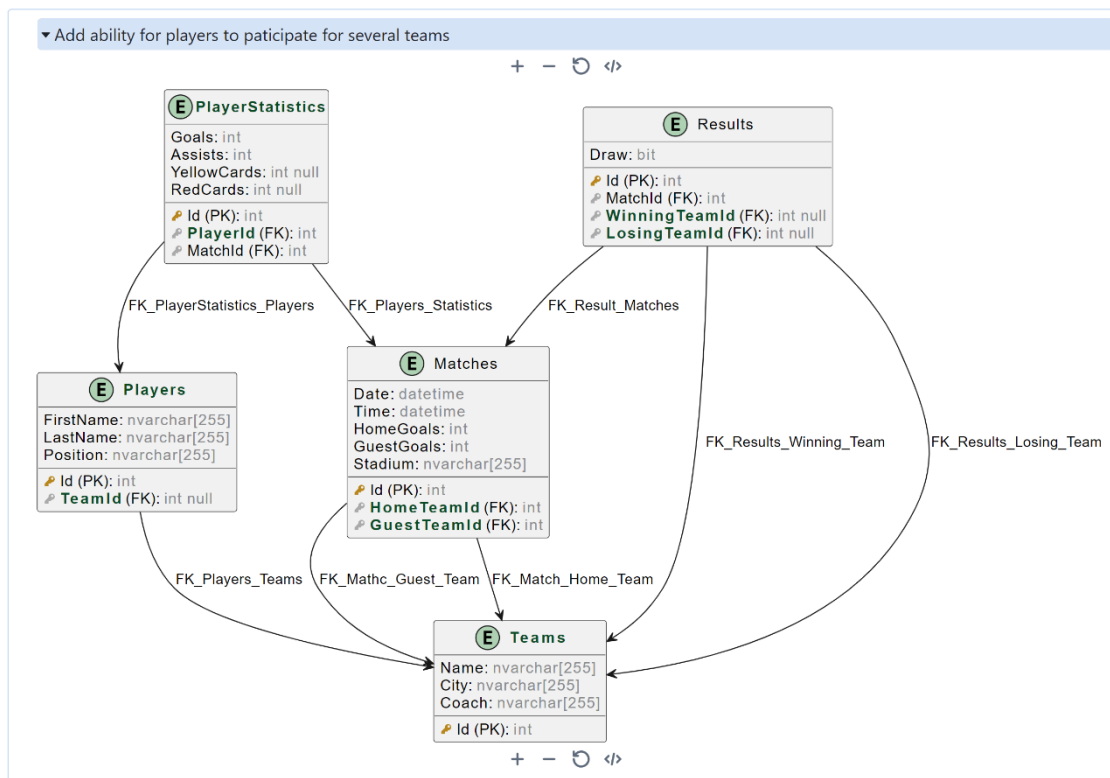


Рис. 12. Візуальна діаграма, яку бачить користувач сервісу як результат обробки системної вимоги, опублікованої українською мовою і перекладеної англійською мовою

г) сервіс, який реалізує розроблену ІТ, забезпечує релевантність результатів виявлення термінів та артефактів проекту ІС у випадку перекладу тієї самої публікації системної вимоги різними мовами (за умови релевантного перекладу).

У попередніх дослідженнях, які проводилися, в тому числі, одним з авторів цього дослідження, для аналізу текстових публікацій найменувань функцій розроблюваної ІС було використано метод стемінгу [24], [25]. На відміну від цих досліджень, розроблена ІТ виявлення термінів та артефактів проекту ІС використовує для аналізу текстових публікацій системних вимог методи лематизації, що дозволяє точніше визначати лєми в контексті використання слова, забезпечуючи цим підвищення точності ідентифікації термінів, які мають значення для подальшого аналізу вимог.

Але розроблена ІТ виявлення термінів та артефактів проекту ІС має і недоліки, які визначають декі обмеження її застосування. Серед цих недоліків слід особливо визначити такі:

а) проблеми визначення коректного контексту для артефактів, знайдених у вимогах (наприклад, для вимоги «Increase the number of characters for the team name» атрибут «name» буде визначено системою для кожної сутності, а не тільки для сутності «team», яка зазначена у вхідному тексті вимоги);

б) визначення правильних типів зв'язків (наприклад, «один до одного», «один до багатьох» тощо) потребує, можливо, додаткової розробки та впровадження в ІТ відповідних запитів до сховища, де зберігаються описи артефактів ІС.

Ці недоліки визначають такі основні напрями подальших досліджень з вдосконалення та подальшого розвитку розробленої ІТ виявлення термінів та артефактів проекту ІС:

а) розвиток та апробація зазначеної ІТ для випадків, коли описами артефактів ІС є

описи патернів програмного забезпечення, фрагментів програмного коду, окремі сервіси та мікросервіси, інші представлення програмних та програмно-апаратних рішень елементів ІС;

б) модифікація зазначеної ІТ для врахування особливостей найпоширеніших підходів, методів та інструментальних засобів формування публікацій вимог зацікавлених сторін та системних вимог до створюваної або вдосконалюваної ІС;

в) дослідження можливості адаптації зазначеної ІТ для аналізу аудіо- та відеофайлів, які містять результати інтерв'ювання представників замовників ІС.

7. Висновки

В процесі виконання даного дослідження було поставлено та вирішено задачу розробки ІТ виявлення термінів та артефактів проєкту ІС. Ця ІТ дозволяє проводити дослідження текстових публікацій системних вимог, які висуваються до створюваної або вдосконалюваної ІС, і, зокрема, визначати артефакти ІТ, які можуть бути повторно використані для реалізації окремих термінів, присутніх в текстовій публікації системної вимоги. В основу цієї ІТ було запропоновано покласти рішення про доцільність застосування для аналізу текстових публікацій системних вимог методів NLP і, зокрема, методів лематизації.

Виходячи з особливостей методів лематизації та основних вимог, які були висунуті до ІТ в процесі її розробки, було визначено основні архітектурні компоненти ІТ та модулі сервісу, які відповідають за реалізацію цих компонентів. Розроблено опис архітектури ІТ у вигляді діаграми компонентів UML, яка описує основні взаємодії між виділеними модулями сервісу, а також між сервісом та сховищем, де зберігаються описи артефактів ІС, та між сервісом і можливими користувацькими інтерфейсами.

Виходячи з розробленого опису архітектури ІТ виявлення термінів та артефактів проєкту ІС, було визначено технологічний стек цієї ІТ. Він дозволяє забезпечити швидко розробку сервісу, який реалізує ІТ, підтримку можливості обробки публікацій системних вимог різними людськими мовами, а також легкість інтеграції окремих модулів у єдиний цілісний сервіс.

Під час розробки елементів ІТ виявлення термінів та артефактів проєкту ІС було використано мови програмування C# та Python. Продемонстровано результати розробки таких елементів програмного забезпечення сервісу, що реалізує ІТ, як опис конфігурації мовних моделей, клас Lang і його метод GetDocument, клас DbEntitiesLemmatizer, який декорує описи артефактів ІС лемами, а також клас PlantUmlBuilder, що відіграє ключову роль у створенні коду для візуалізації діаграми.

Для проведення експериментальної перевірки отриманих результатів було застосовано ІС «Футбол», яка включає до себе набір основних функцій, що дозволяють автоматизувати вирішення задач планування та обліку проведення футбольних матчів. Розроблений сервіс, який реалізує ІТ виявлення термінів та артефактів проєкту ІС, дозволив визначати терміни системних вимог до ІС «Футбол» та артефактів ІС «Футбол», тексти яких опубліковано англійською та українською мовами, візуалізувати підмножину артефактів ІС, терміни яких співпадають із термінами опублікованої системної вимоги, у вигляді візуальної діаграми, а також забезпечити релевантність результатів виявлення термінів та артефактів проєкту ІС у випадку перекладу тієї самої публікації системної вимоги різними мовами (за умови релевантного перекладу).

Отримані результати дозволяють в подальшому вирішити задачу розробки складнішої ІТ, головна задача якої полягає у автоматизації синтезу описів архітектури ІС, яка створюється або вдосконалюється на основі обмеженої множини текстових публікацій вимог зацікавлених сторін або системних вимог до цієї ІС.

Перелік посилань:

1. ДСТУ ISO/IEC/IEEE 15288:2016 Інженерія систем і програмного забезпечення. Процеси життєвого циклу систем. [Електронний ресурс]. Сайт «Друковані видання в цифровому вигляді». URL: <https://www.yumpu.com/xx/document/read/67240647/-iso-iec-ieee-15288-2016-/7> (дата звернення: 02.04.2024).
2. Настанова до Зводу знань з управління проєктами. Настанова РМВОК. Сьоме видання. Стандарт з управління проєктами. Project Management Institute, Inc., 14 Campus Boulevard Newtown Square, Pennsylvania 19073-3299 USA, 2021. 370 с.
3. Cadavid H., Andrikopoulos V., Avgeriou P., Broekema P. Ch. System and software architecting harmonization practices in ultra-large-scale systems of systems: A confirmatory case study. *Information and Software Technology*. 2022. Vol.150. № 106984. DOI: <https://doi.org/10.1016/j.infof.2022.106984>
4. Revolutionizing Requirements Engineering: Unleashing the Power of NLP and Generative AI [Електронний ресурс]. Сайт «LinkedIn». URL: <https://www.linkedin.com/pulse/revolutionising-requirements-engineering-unleashing-power-shaun-koon>, вільний (дата звернення: 03.04.2024)
5. VisualNarrator Tool [Електронний ресурс]. Сайт «GitHub». URL: <https://github.com/MarcelRobeer/VisualNarrator> (дата звернення: 03.04.2024).
6. QuARS: A Tool for Analyzing Requirements [Електронний ресурс]. Сайт «QUARS». URL: <https://www.quars.it/> (дата звернення: 03.04.2024).
7. Henning Femmer. Requirements Quality Defect Detection with the Qualicen Requirements Scout. Technical University Munich and Qualicen, Munich, Germany [Електронний ресурс]. URL: https://ceur-ws.org/Vol-2075/NLP4RE_paper2.pdf (дата звернення: 04.04.2024).
8. Semantha [Електронний ресурс]. Сайт «Semantha». URL: <https://www.semantha.de/semantha-requirements/> (дата звернення: 04.04.2024).
9. ReqSuite [Електронний ресурс]. Сайт «ReqSuite». URL: <https://www.osseno.com/en/requirements-management-tool/> (дата звернення: 04.04.2024).
10. IBM Engineering Requirements Quality Assistant [Електронний ресурс]. Сайт «IBM». URL: <https://www.ibm.com/docs/en/erqa?topic=assistant-overview>, вільний (дата звернення: 05.04.2024).
11. Defining Terms Used to Describe Requirements [Електронний ресурс]. Сайт «Learn Microsoft». URL: <https://learn.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2015/modeling/model-user-requirements?view=vs-2015&redirectedfrom=MSDN#RequirementsClasses>.
12. Lee R. S. T. Natural Language Processing. A Textbook with Python Implementation. Singapore: Springer Nature Singapore Pte Ltd., 2024. XXXII, 437 p. DOI: <https://doi.org/10.1007/978-981-99-1999-4>
13. Candase Hokanson, Karl Wieggers. Software Requirements Essentials: Core Practices for Successful Business Analysis. Addison-Wesley Professional. 2023. 208 p.
14. Phillip A. Laplante, Mohamad Kassab. Requirements Engineering for Software and Systems (Applied Software Engineering Series. Taylor & Francis. 2022. 404 p.
15. Levykin V., Yevlanov M., Neumyvakina O., Petrichenko O. Concept of Artifact-Event Description of Information System. Fourth International Scientific and Technical Conference «COMPUTER AND INFORMATION SYSTEMS AND TECHNOLOGIES». Kharkiv: DISA PLUS LLC, 2020. P. 57-58. DOI: <https://doi.org/10.30837/IVcsitic2020201438>
16. Як вибрати правильний технологічний стек для вашого проєкту [Електронний ресурс]. Сайт «REDSTONE». URL: <https://redstone.agency/blog/yak-vybraty-pravylnyi-tekhnologichnyi-stek-dlia-vashoho-proektu/> (дата звернення: 30.10.2024).
17. Introduction to .NET [Електронний ресурс]. Сайт «Learn Microsoft». URL: <https://dotnet.microsoft.com/en-us/platform/support/policy/dotnet-core>
18. Industrial-Strength Natural Language Processing [Електронний ресурс]. Сайт «SpaCy». URL: <https://spacy.io/>, вільний (дата звернення: 17.04.2024)
19. SpacyDotNet .NET wrapper [Електронний ресурс]. Сайт «GitHub». URL: <https://github.com/AMARostegui/SpacyDotNet>
20. Drawing UML with PlantUML [Електронний ресурс]. Сайт «PlantUML». URL: <https://plantuml.com/guide>
21. The library for web and native user interfaces [Електронний ресурс]. Сайт «React». URL: <https://react.dev/learn>
22. Blueprint React-based UI toolkit for the web [Електронний ресурс]. Сайт «BlueprintJs». URL: <https://blueprintjs.com/docs/>, вільний (дата звернення: 23.04.2024).
23. «OpenAPI Guide» [Електронний ресурс]. Сайт «Swagger». URL: <https://swagger.io/docs/specification/about/> (дата звернення: 26.04.2024).
24. Ievlanov M., Vasilcova N., Panforova I. Development of methods for the analysis of functional requirements to an information system for consistency and illogicality. *Eastern-European Journal of Enterprise Technologies*. 2018. T. 1. Vol. 2(91). P. 4–11. DOI: <https://doi.org/10.15587/1729-4061.2018.121849>
25. Vasilcova N., Panforova I., Neumyvakina O. Improving a method to analyze the requirements for an

Євланов Максим Вікторович, доктор технічних наук, професор, професор кафедри ІУС ХНУРЕ, м. Харків, Україна, e-mail: maksym.ievlanov@nure.ua, ORCID: <https://orcid.org/0000-0002-6703-5166> (науковий керівник здобувача вищої освіти Лучицького В.В.).

Мороз Борис Іванович, доктор технічних наук, професор, професор кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна, e-mail: moroz.b.i@nmu.one, ORCID: <https://orcid.org/0000-0002-5625-0864>;

Мороз Дмитро Максимович, доктор філософії, доцент, доцент кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна, e-mail: moroz.d.m@nmu.one, ORCID: <https://orcid.org/0000-0003-2577-3352>

Лучицький В'ячеслав Володимирович, здобувач вищої освіти, група УПГІТм-22-3, факультет комп'ютерних наук, ХНУРЕ, м. Харків, Україна, e-mail: viacheslav.luchytskyi@nure.ua.

УДК 004.8:004.9

DOI: 10.30837/0135-1710.2024.182.093

Є.В. БОДЯНСЬКИЙ, О.С. ЧАЛА

ШВИДКА КЛАСИФІКАЦІЯ В ONLINE- ТА NEARLINE-РЕЖИМАХ В УМОВАХ КЛАСІВ, ЩО ПЕРЕТИНАЮТЬСЯ

Предметом дослідження є процес швидкої класифікації в умовах класів, що перетинаються. Метою є розробка підходу до швидкої класифікації, який поєднує online- та nearline-режими для підвищення точності класифікації в умовах класів, що перетинаються. Розроблено багатопарову нейронну мережу з ядерними дзвонуватими функціями активації для роботи в online-режимі. Запропоновано адаптивну нейро-фаззі систему для класифікації даних у матричній формі, яка використовує гібридне комбіноване навчання для роботи в nearline-режимі. Підхід орієнтований на вирішення задач уточнення границь між класами, адаптації до змін у розподілі вхідних даних, усунення дисбалансу класів та видалення шумових точок.

1. Вступ

Класифікація, яка є однією з ключових задач машинного навчання, полягає у віднесенні вхідних даних до одного з попередньо визначених класів. Ця задача має широкий спектр практичних застосувань, включаючи розпізнавання об'єктів на фотографіях, виявлення комп'ютерних вірусів, аналіз відеоданих із систем відеоспостереження тощо [1].

Швидка класифікація передбачає обробку даних у режимі реального часу (real time) або близькому до нього (near real time). Вирішення задачі швидкої класифікації потребує розробки архітектур нейронних мереж, здатних забезпечити високу точність класифікації при мінімальних обчислювальних витратах. Такі системи мають практичне застосування у задачах, де швидкість обробки є ключовим фактором, наприклад, у системах підтримки керування транспортними засобами або безпекового моніторингу [2].

Швидка класифікація поєднує online- та nearline-класифікацію. Online-класифікація орієнтована на обробку даних у режимі реального часу, що передбачає задоволення часових обмежень та, як наслідок, обмежень щодо обчислювальної складності алгоритмів обробки даних. Nearline-класифікація передбачає затримку в декілька хвилин на обробку даних, що дає можливість використовувати складніші моделі та отримати результати з вищою точністю. Як online-, так і nearline-класифікація передбачають узгодження швидкості та точності класифікації [3].

Швидка класифікація зазвичай виконується в умовах класів, що перетинаються (overlapping classes). Проблема класів, що перетинаються, виникає в тому випадку, коли об'єкти з різних класів мають схожі характеристики, тобто відповідні дані розміщуються рядом у просторі ознак. Як результат, відсутньою є чітка границя між класами, що суттєво ускладнює класифікацію [4]. Причинами перетину класів можуть бути низька роздільна здатність ознак, наявність шуму у вхідних даних, природна варіабельність даних в рамках класів тощо [5].

Реалізація швидкої класифікації в умовах класів, що перетинаються, пов'язана із вирішенням задач усунення дисбалансу класів, видалення точок шуму у просторі ознак, уточнення границі перетину класів та адаптації з урахування нестаціонарності вхідних даних (concept drift). Дисбаланс класів виникає, коли класи представлені суттєво різною кількістю зразків даних, що може впливати на точність класифікації. Точки шуму (noise points) є такими елементами даних, які не відповідають представленим класами закономірностям. Вказані точки можуть розглядатись як аномалії або викиди (outliers). Причинами точок шуму можуть бути помилки вимірювання, неправильне маркування вхідних даних для класифікації, непередбачувані впливи в процесі збору даних [6]. Видалення точок шуму у просторі ознак дає можливість підвищити точність класифікації з урахуванням випадкових флуктуацій у вхідних даних. Уточнення границі перетину класів є важливою умовою для підвищення точності класифікації у випадках, коли частина даних з різних класів має схожі характеристики. Врахування concept drift дозволяє виконати адаптацію моделі до змін у розподілі вхідних даних, які відбуваються з часом, що є актуальним для систем, які функціонують у динамічних середовищах [7].

Поєднання підходів online- та nearline-класифікації дає можливість створити умови для вирішення розглянутих задач в умовах класів, що перетинаються. Online-класифікація може бути використана для оперативного уточнення границь між класами та створення умов для адаптації до змін у розподілі вхідних даних. В подальшому в рамках nearline-класифікації можуть бути доповнені попередні результати online-класифікації за рахунок усунення дисбалансу класів та видалення шумових точок. Зазначене свідчить про актуальність інтеграції online- та nearline-підходів при реалізації швидкої класифікації в умовах класів, що перетинаються.

2. Аналіз літературних даних і постановка проблеми дослідження

Методи online-класифікації орієнтовані на постійну адаптацію до нових даних у режимі реального часу. Дана адаптація реалізується з використанням інкрементного методу опорних векторів [8]. Підхід до online-класифікації з використанням еволюційних нейро-фаззі систем забезпечує високу швидкість навчання та точність класифікації [9]. Застосування нейронних мереж з нечіткою логікою для обробки потокових даних забезпечує високу точність класифікації при обмеженнях щодо витрат обчислювальних ресурсів [10]. Для вирішення задач усунення дисбалансу класів у online-режимі використовуються стратегії генерації додаткових даних у вибірці для менш представленого класу та зменшення кількості даних більш представленого класу [11]. Проте імплементація даних стратегій може привести до часткової втрати вхідної інформації.

Методи nearline-класифікації використовують розподілену обробку для зменшення часу класифікації [12]. В [13] запропоновано адаптивний підхід для обробки даних в умовах concept drift. Задача усунення дисбалансу класів вирішується з використанням методу SMOTE, який збільшує кількість зразків менш представленого класу [14]. Видалення шумових точок здійснюється з використанням методу Edited Nearest Neighbors (ENN) для випадків, коли більшість найближчих сусідів належить до іншого класу [15].

Вдосконалену багатовимірну нео-нечітку систему для задач класифікації відео, що має можливість адаптації до змін у вхідних даних, запропоновано в [16].

Проблема класів, що перетинаються, вирішується з використанням стратегій відкидання зон перетину або їх об'єднання у мета-клас [17]. Використовується також підхід, який полягає у зменшенні кількості даних домінуючого класу у зоні перетину [18]. Використання нейро-фаззі систем для класифікації в умовах класів, що перетинаються, які здатні адаптуватися до змін у структурі даних з використанням комбінованого навчання, запропоновано в [9], [19], [20]. Такі нейро-фаззі системи та нео-нечіткі системи [16] мають можливість комплексного вирішення задач швидкої класифікації за рахунок використання ковзного вікна, адаптації границі між класами, а також використання фаззі-підходу.

Таким чином, сучасні підходи до швидкої класифікації вирішують окремо задачу усунення дисбалансу класів, видалення шумових точок у просторі ознак, уточнення границь перетину класів та адаптації до змін у розподілі вхідних даних (concept drift). Однак ці задачі в умовах класів, що перетинаються, є взаємопов'язаними і потребують комплексного вирішення. Тому при класифікації в нестационарному середовищі з дисбалансом класів, шумом у вхідних даних, перетином класів актуальною є задача побудови єдиного підходу до швидкої класифікації з використанням можливостей нейро-фаззі системи та нео-нечітких систем.

3. Мета і задачі дослідження

Метою даного дослідження є розробка підходу до швидкої класифікації, який поєднує online- та nearline-режими, для підвищення точності класифікації в умовах класів, що перетинаються.

Для досягнення поставленої мети у роботі вирішуються такі задачі: розробка багатосарової нейронної мережі з ядерними дзвонуватими функціями активації, що орієнтована на роботу в online-режимі; розробка адаптивної нейро-фаззі системи для вирішення задачі класифікації даних, представлених у матричній формі, яка орієнтована на роботу як у online-, так і у nearline-режимі.

Реалізація даного підходу полягає у реалізації швидкої класифікації у online-режимі з тим, щоб забезпечити оперативну адаптацію до змін у вхідних даних та подальше уточнення класифікації у nearline-режимі за рахунок усунення дисбалансу класів та видалення шумових точок.

4. Багатосарова нейронна мережа з ядерними дзвонуватими функціями активації

Розроблена багатосарова нейронна мережа з ядерними дзвонуватими функціями активації, яка є подальшим розвитком роботи [21], призначена для вирішення задач класифікації в умовах класів, що перетинаються, і відрізняється тим, що не потребує великих обсягів навчальних вибірок і не використовує зворотне поширення помилок.

Мережа містить три шари.

Вхідний шар отримує дані у вигляді матриць, що спрощує обробку зображень.

Прихований шар містить нейрони з ядерними дзвонуватими функціями активації, які моделюють складні нелінійні границі між класами. Рівень активації $\psi_n(x(t))$ розраховується на основі відстані між вхідними даними та центром нейрона:

$$\psi_n(x(t)) = \exp(-Sp(x(t) - z_n)^T \Sigma_n^{-1}(x(t) - z_n)), \quad (1)$$

де $x(t)$ – вхідні дані; z_n – центр n -го нейрона; Σ_n^{-1} матриця, що визначає форму та орієнтацію рецептивного поля, тобто області вхідних даних, на яку реагує конкретний

нейрон у прихованому шарі.

Такий підхід дає можливість мережі ефективно розрізняти класи навіть у випадках, коли їхні границі є складними та нелінійними.

Вихідний шар формує результат класифікації на основі виходів прихованого шару. Він використовує лінійну комбінацію виходів $\tilde{\psi}(x(t))$ нейронів прихованого шару з вектором v вагових коефіцієнтів :

$$\hat{y}(t) = v^T \tilde{\psi}(x(t)). \quad (2)$$

Дана система використовує рекурентний алгоритм навчання, який може оновлювати параметри мережі в реальному або близькому до реального часі. Такий підхід дозволяє мережі адаптуватися до нових даних без необхідності повторного навчання на всьому наборі даних. Адаптивне налаштування параметрів функцій активації виконується за допомогою градієнтного методу. Тобто якщо поточний вихід мережі відрізняється від бажаного, ваги коригуються у напрямку, що зменшує цю різницю. Параметр швидкості навчання $\omega_v(t+1)$ визначає, наскільки великою буде така корекція. Даний алгоритм не використовує зворотне поширення помилки, що дозволяє уникнути проблеми «зникаючого градієнта».

Таким чином, використання ядерних функцій активації забезпечує адаптацію системи до складних границь між класами при класифікації в умовах класів, що перетинаються. Онлайн-уточнення границь класів створює умови для адаптації до змін у розподілі вхідних даних.

5. Адаптивна нейро-фаззі система для вирішення задачі класифікації даних, представлених у матричній формі

Адаптивна нейро-фаззі система призначена для вирішення задачі класифікації при представленні даних як у матричній, так і в векторній формі в умовах класів, що перетинаються, та є подальшим розвитком роботи [22].

Дана система складається з чотирьох шарів.

Перший шар містить нейрони, які працюють на основі активаційних функцій, що залежать від відстані між вхідними даними та певним центром у просторі ознак. Вказані центри визначаються під час навчання і розташовуються у просторі ознак так, щоб мінімізувати похибку класифікації. Кількість нейронів визначається кількістю спостережень N . Активаційні функції цього шару позначаються як $\psi(x, z_{t_n}, \sigma_{t_n}^2)$, де z_{t_n} – центр активаційної функції; $\sigma_{t_n}^2$ – параметр, що визначає «ширину» дзвонуватої активаційної функції.

Другий шар складається з елементарних суматорів для кожного класу. На виходах $o_n^{(2)}(x)$ формуються сигнали, які є оцінками Парзена щільності розподілу ймовірностей. Тобто розраховується сума активацій нейронів, що належать до n -го класу для вхідного образу x .

Третій шар коригує оцінки з урахуванням емпіричних апріорних ймовірностей та вартості помилок класифікації:

$$o_n^{(3)}(x) = o_n^{(2)}(x) \frac{N_n}{N} S_n, \quad (3)$$

де $\frac{N_n}{N}$ – емпірична апріорна ймовірність n -го класу; S_n – вартість помилки класифікації

для n -го класу.

Вихідний шар визначає рівні належності класифікованих образів до конкретних класів.

Налаштування параметрів активаційних функцій здійснюється згідно з концепцією «нейрони в точках даних», де центри активаційних функцій визначаються як

$$z_{t_n} = x(k) | x(k) \in Cl_n, \quad (4)$$

де z_{t_n} – центр активаційної функції; $x(k)$ – k -те спостереження у наборі даних, представлено вектором ознак; Cl_n – n -й клас.

Концепція «нейрони в точках даних» передбачає, що центри активаційних функцій нейронів встановлюються згідно зі значеннями вхідних даних. Тобто кожен нейрон у мережі відповідає конкретному спостереженню з навчального набору даних. Такий підхід забезпечує високу швидкість навчання, оскільки кожне нове спостереження стає центром відповідної активаційної функції.

Дана система використовує гібридне комбіноване навчання, яке включає «лінійне навчання» на основі концепції «нейрони в точках даних», самоорганізацію за принципом «переможець забирає все» для оновлення центроїдів класів, навчання векторного квантування для адаптивного коригування положення центроїдів, а також оцінку рівнів належності на основі нечіткої кластеризації. Навчання реалізується у послідовному online- або nearline-режимі, що дає можливість системі адаптуватись за допомогою алгоритмів рекурентного навчання. Це дозволяє системі динамічно змінювати свою архітектуру шляхом додавання нових нейронів, а також параметри у процесі надходження нових даних.

Дана система може бути використана в режимах online та nearline в залежності від обсягу даних, що обробляються. Nearline-режим дозволяє балансувати між швидкістю обробки та точністю моделі, забезпечуючи можливість більших об'ємів обчислень без значної затримки.

Дана система створює умови для усунення дисбалансу класів на основі налаштування вагових коефіцієнтів для різних класів. Видалення шумових точок може бути реалізовано на основі використання радіально-базисних функцій, що дають можливість зменшити вплив видалених точок. Крім того, nearline-навчання дає можливість уточнити класифікацію з урахуванням змін у розподілі вхідних даних.

Таким чином, комплексне використання багатошарової нейронної мережі з ядерними дзвонуватими функціями активації в online-режимі та адаптивної нейро-фаззі системи в nearline-режимі забезпечує вирішення задач уточнення границь між класами та адаптації класифікації до змін у розподілі вхідних даних, а також створює умови для вирішення задачі видалення шумових точок та усунення дисбалансу класів.

6. Експериментальна перевірка підходу до швидкої класифікації в умовах класів, що перетинаються, на основі комбінування online- та nearline-режимів

Експериментальна перевірка технології виконана для задачі розпізнавання зображень у системі відеоспостереження. Такі системи використовують два відеопотоки: основний (Main Stream) та додатковий (Sub Stream). Основний потік містить відео з максимальною роздільною здатністю, доступною для відеокамери. Додатковий потік характеризується нижчою якістю зображення, що дозволяє знизити об'єм переданих через мережу даних. Додатковий потік обробляється в online-режимі, а основний – в nearline-режимі.

При проведенні експерименту використано датасет CDnet (Change Detection Dataset), який використовується для оцінки алгоритмів відеоспостереження. Він містить

послідовності кадрів з низькою і високою роздільною здатністю [23].

Для оцінки впливу concept drift використано набір «winterDriveway» із вказаного датасету із низькою роздільною здатністю 320 на 240 пікселів. Набір містить зображення машин під снігом. Починаючи з кадру in000387, відбувається зміна розподілу: на зображенні крім машин з'являється водій (рис. 1). Оцінку зміни розподілу виконано з використанням показника точності (accuracy) [24] шляхом визначення змін точності показника через 10 кадрів з округленням до сотих і представлено в табл. 1. Із таблиці видно несуттєве, в межах 2% зниження точності класифікації в результаті зміни розподілу вхідних даних.



Рис. 1. Зміна розподілу вхідних даних

Оцінка балансу класів орієнтована на обробку зображень з вищою роздільною здатністю у nearline-режимі. Для оцінки балансу класів використано показник збалансованої точності (Balanced Accuracy). Даний показник обчислюється на основі значень чутливості та специфічності [25]. Збалансована точність, на відміну від традиційного показника точності, не залежить від розподілу класів у тестовому наборі даних.

Таблиця 1

Зміна точності класифікації в результаті зміни розподілу вхідних даних

Номер кадру	Точність,
in000387	0,89
in000397	0,88
in000407	0,87
in000417	0,87
in000427	0,87
in000437	0,87

У процесі експерименту використано 50 зображень 720 x 576 із незбалансованого набору PETS2006 із датасету CDnet. Отримане значення показника чутливості становить 0,89, показника специфічності 0,871. Відповідно, значення показника збалансованої точності становить 0,88, що свідчить про те, що модель добре справляється з класифікацією, навіть враховуючи незбалансованість класів.

7. Обговорення результатів

Результати виконаної експериментальної перевірки показали, що комбінування багатосарової нейронної мережі з ядерними дзвонуватими функціями активації та адаптивної нейро-фаззі системи для класифікації даних дає можливість комплексно вирішити задачі зміни розподілу вхідних даних та балансування класів. Тобто послідовне

використання моделей в online- та nearline-режимах створює умови для вирішення типових проблем класифікації в умовах класів, що перетинаються.

Подальший розвиток підходу пов'язаний із розробкою інформаційної технології, яка на основі комбінування online- та nearline-режимів забезпечувала б також додаткове вирішення задач видалення точок шуму та усунення дисбалансу класів для проведення класифікації, що створює умови для підвищення точності класифікації при наявності помилок вимірювання внаслідок неточності сенсорів або використання некоректних методів збору даних.

8. Висновки

У дослідженні розроблено підхід до швидкої класифікації в умовах класів, що перетинаються, на основі комбінування online- та nearline-режимів, який орієнтований на вирішення задач оперативної адаптації до змін у вхідних даних та уточнення результатів класифікації за рахунок усунення дисбалансу класів та видалення шумових точок. Даний підхід передбачає використання у online-режимі багатошарової нейронної мережі з ядерними дзвонуватими функціями активації та у nearline-режимі адаптивної нейро-фаззі системи, що призначена для вирішення задачі розпізнавання образів, які задані у матричній або векторній формі.

Запропоновано багатошарову нейронну мережу з ядерними дзвонуватими функціями активації. Мережа призначена для вирішення задачі класифікації в умовах класів, що перетинаються. Дана мережа відрізняється використанням рекурентного алгоритму навчання, що дає можливість відмовитись від процедури зворотного поширення помилок і, як наслідок, уникнути негативного ефекту «зникаючого градієнта». Багатошарова нейронна мережа не потребує великих навчальних вибірок і орієнтована на використання у online-режимі. У практичному плані мережа забезпечує адаптацію системи до складних границь між класами та створює умови для адаптації до змін у розподілі вхідних даних.

Запропоновано адаптивну нейро-фаззі (нейронечітку) систему для класифікації в умовах класів, що перетинаються. Система використовує дані, задані як у матричній, так і в векторній формі. Система, на відміну від існуючих, використовує гібридне комбіноване навчання для налаштування не лише синаптичних ваг, а й параметрів ядерних (дзвонуватих) функцій активації. Безпосередньо навчання реалізовано в послідовному online- або nearline-режимі при надходженні нових даних, що дає можливість уточнити результати класифікації з урахуванням змін у розподілі вхідних даних. У практичному плані мережа створює умови для усунення дисбалансу класів та шумових точок з урахуванням рівня належності для результатів класифікації.

Перелік посилань:

1. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.
2. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.
3. Wang, J., Ma, Y., Zhang, L., Gao, R. X., & Wu, D. (2018). Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems*, 48, 144-156.
4. Xiong, H., Wu, J., & Liu, L. (2010). Classification with class-overlapping: A systematic study and a new measure of class separability. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 40(4), 973-985.
5. Vuttipittayamongkol, P., Zhang, X., & Yang, H. (2020). Overlap-based undersampling for improving classification performance on imbalanced data with overlapping classes. *Knowledge-Based Systems*.
6. GeeksforGeeks. (2024). How to handle Noise in Machine learning? Retrieved from <https://www.geeksforgeeks.org/how-to-handle-noise-in-machine-learning/>
7. Krawczyk, B., Minku, L. L., Gama, J., Stefanowski, J., & Woźniak, M. (2017). Ensemble learning for data stream analysis: A survey. *Information Fusion*, 37, 132-156.
8. Bifet, A., Holmes, G., Kirkby, R., & Pfahringer, B. (2010). MOA: Massive Online Analysis. *Journal of Machine*

Learning Research, 11(May), 1601-1604.

9. Bodyanskiy, Y., Pliss, I., & Vynokurova, O. (2015). An adaptive learning algorithm for evolving neuro-fuzzy systems in the online mode. *Information Sciences*, 294, 189-204.
10. Bodyanskiy, Y., Pliss, I., Chala, O., & Deineko, A. (2016). Evolving fuzzy-probabilistic neural network and its online learning. In *10th International Conference on Advanced Computer Information Technologies*. Deggendorf, Germany. <https://doi.org/10.1109/ACIT49673.2020.9208904>
11. Wang S., Minku L.L., Yao X. (2016). Dealing with Multiple Classes in Online Class Imbalance Learning. *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*.
12. Ma C., Zhang Y., Li J. (2023). Multi-Scenario Nearline Retrieval for E-commerce Platforms. *Journal of Data and Information Science*.
13. Žliobaitė I., Pečenizkiy M., Gama J. (2016). An overview of concept drift applications in data mining. In *Big Data Analysis: New Algorithms for a New Society*
14. Chawla N.V., Bowyer K.W., Hall L.O., Kegelmeyer W.P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*.
15. Batista G.E.A.P.A., Prati R.C., Monard M.C. (2004). A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. *SIGKDD Explorations Newsletter*.
16. Bodyanskiy, Y., & Chala, O. (2024). Enhanced multidimensional neo-fuzzy classification system and its learning for the video classification task. *Management Information System and Devices*, 181, 42-50. Retrieved from https://www.ewdtest.com/asu/wp-content/uploads/2024/09/ASUtaPA_181_42_50.pdf
17. Xiong H., Wu J., Liu L. (2010). Classification with class-overlapping: A systematic study and a new measure of class separability. *IEEE Transactions on Systems*.
18. Das S., Dey S., Pal S.K. (2020). Handling Class Overlap and Imbalance in Classification Problems Using a Hybrid Approach. *IEEE Transactions on Neural Networks and Learning Systems*.
19. Bodyanskiy Y., Chala O.V. & others. (2017). Evolutionary neuro-fuzzy networks with cascade structure for intelligent data analysis tasks.
20. Bodyanskiy, Y., Deineko, A., Pliss, I., & Chala, O. (2020). Fast probabilistic neuro-fuzzy system for pattern classification task. *Information Technology and Management Science*, 23, 12-16.
21. Chala, O., Bodyanskiy, Y., Sachenko, A., & Dobrowolski, M. (2023). Matrix hyper-basis function neural network and its online learning. In *The 12th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, September 7-9, Dortmund, Germany (pp. 1-4).
22. Bodyanskiy, Y., Deineko, A., Pliss, I., & Chala, O. (2020). Matrix fuzzy-probabilistic neural network in image recognition task. In *Proceedings of the IEEE Third International Conference on Data Stream Mining & Processing (DSMP)*, Lviv, Ukraine, August 21-25 (pp. 33-36).
23. Wang, Y., Jodoin, P.-M., Porikli, F., Konrad, J., Benezeth, Y., & Ishwar, P. (2014). CDnet 2014: An expanded change detection benchmark dataset. In *Proceedings of the IEEE Workshop on Change Detection (CDW-2014) at CVPR-2014* (pp. 387-394). Retrieved from <http://jacarini.dinf.usherbrooke.ca/dataset2014>
24. Powers, D. M. W. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, 2(1), 37-63.

Надійшла до редколегії 25.09.2024 р.

Бодяньський Євген Володимирович, доктор технічних наук, професор, професор кафедри ШІ ХНУРЕ, м. Харків, Україна, e-mail: yevgeniy.bodyanskiy@nure.ua; ORCID: 0000-0001-5418-2143
Чала Ольга Сергіївна, старший викладач кафедри штучного інтелекту ХНУРЕ, м. Харків, Україна, e-mail: olha.chala@nure.ua; ORCID: 0000-0002-7603-1247

РЕФЕРАТИ / ABSTRACTS

УДК 004.91

Розробка методу вирішення задачі формування і класифікації запитів на зміни ІТ-продукту / М.В. Євланов, І.О. Юр'єв, Б.Є. Крук. АСУ та прилади автоматики. 2024. № 182. С. 5-25.

Об'єктом дослідження є робота «Визначте/уточніть потребу в змінах» підпроцесу «Формулювання змін» процесу «Управління змінами». Визначено, що існуючий фреймворк управління змінами має загальний та рекомендаційний характер і не визначає найбажаніші методи та інструменти управління змінами. Існуючі моделі, методи та технології формування і класифікації запитів на зміни розроблені згідно з особливостями конкретних ІТ-проектів та ІТ-продуктів. Проблема уніфікації формального опису моделей і методів робіт з формулювання та класифікації запитів на зміни ІТ-продукту залишається невирішеною, що обумовлює актуальність подібних досліджень.

Виділено та проаналізовано особливості існуючих інформаційних технологій формування та класифікації запитів на зміни, які виникають під час виконання ІТ-проектів. Внесено пропозицію розглядати ці технології як окремі випадки уніфікованої інформаційної технології. Виходячи з цієї пропозиції, було розроблено теоретико-множинну модель запиту на зміну ІТ-продукту, яка на формальному рівні дозволяє описати будь-які функціональні та технічні запити на зміни, що виникають під час виконання ІТ-проекту з розробки або модифікації ІТ-продукту.

На основі отриманої моделі було розроблено загальний метод вирішення задачі формування і класифікації RFC. Цей метод, на відміну від існуючих, дозволяє не тільки застосовувати апріорно визначену модель класифікації RFC, а й адаптувати існуючу модель до нових особливостей ІТ-проекту та ІТ-продукту та розробляти власну модель класифікації RFC, яка найкращим чином підходить до особливостей діяльності ІТ-компанії, її ІТ-проектів та ІТ-продуктів.

Ключові слова: управління змінами, ІТ-продукт, запит на зміну, класифікація, потреба у зміні, ключові слова.

Табл. 3. Іл. 8. Бібліогр.: 15 назв.

UDC 004.91

Development of a method for solving the task of forming and classifying requests for IT product changes / M.V. Yevlanov, I.O. Yur'ev, B.Ye. Kruk. Management Information System and Devices. 2024. № 182. P. 5-25.

The object of the study is the work "Identify/specify the need for changes" of the sub-process "Formulation of changes" of the process "Management of changes". It was determined that the existing framework of change management is general and recommendatory in nature and does not define the most desirable methods and tools of change management. Existing models, methods and information technologies for the formulation and classification of change requests are developed according to the specifics of specific IT projects and IT products. The problem of unification of the formal description of models and methods of work on the formulation and classification of IT product change requests remains unresolved, which determines the relevance of similar of research

Features of existing information technologies for formulating and classifying requests for changes that arise during the implementation of IT projects were highlighted and analyzed. It is proposed to consider these technologies as separate cases of unified information technology. Based on this proposal, a theoretical-multiple model of an IT product change request was developed, which at the formal level allows to describe any functional and technical requests for changes that arise during the implementation of an IT project for the development or modification of an IT product.

Based on the obtained model, a general method of solving the problem of RFC formation and classification was developed. This method, unlike the existing ones, allows not only to apply an a priori defined RFC classification model, but also to adapt the existing model to the new features of the IT project and IT product and to develop its own RFC classification model that best suits the specifics of the IT company's activities, its IT projects and IT products.

Keywords: change management, IT product, change request, classification, need for change, keywords.

УДК 004.8:004.9

Побудова процесно-орієнтованих пояснень в інтелектуальних інформаційних системах на основі можливісних каузальних залежностей / С.Ф. Чалий, В.О. Лещинський. АСУ та прилади автоматики. 2024. № 182. С. 25-33.

Предметом дослідження є процес побудови пояснень в інтелектуальних інформаційних системах, представлених у вигляді «сірої скриньки». Метою є розробка процесно-орієнтованого підходу до формування пояснень, що забезпечує підтримку та вдосконалення процесу прийняття рішень в інтелектуальних інформаційних системах. Завдання дослідження включають: розробку процесної моделі пояснення на основі можливісних каузальних залежностей; розробку методу побудови процесно-орієнтованого представлення пояснень. Розроблений підхід до побудови пояснення враховує темпоральні та причинно-наслідкові зв'язки між станами процесу формування рішення. Наукова новизна отриманих результатів полягає у розробці узагальненої процесної моделі пояснення, яка дозволяє деталізувати послідовність станів процесу прийняття рішень, визначити ключові дії та контекстні умови їх виконання. Модель створює умови для визначення дій процесу, які мають найбільше значення можливості і, відповідно, ключовий вплив на рішення інтелектуальної системи. Запропонований метод побудови процесно-орієнтованого пояснення в інтелектуальній інформаційній системі на основі можливісних каузальних залежностей включає етапи формування темпоральних і каузальних правил, розрахунок показників можливості та необхідності, а також представлення пояснень з урахуванням атрибутів станів процесу формування рішення. Метод орієнтований на підтримку внутрішніх користувачів інтелектуальної системи на основі побудови інтерпретованого представлення процесу формування рішення.

Ключові слова: пояснення, інтелектуальна система, пояснювальний штучний інтелект, темпоральна залежність, каузальна залежність, можливість, система штучного інтелекту, процесне моделювання.

Табл. 2. Іл. 0. Бібліогр.: 22 назв.

UDC 004.8:004.9

Development of process-oriented explanations in intelligent information systems based on possibilistic causal dependencies / S.F. Chaly, V.O. Leshchynskiy. Management Information System and Devices. 2024. № 182. P. 25-33.

The subject of the study is the process of constructing explanations in intelligent information systems represented as a "gray box." The aim is to develop a process-oriented approach to forming explanations that ensures support and improvement of decision-making processes in intelligent information systems. The research objectives include the development of a process model for explanations based on possibilistic causal dependencies and the creation of a method for constructing process-oriented representations of explanations. The proposed explanation approach considers temporal and causal relationships between states in the decision-making process. The scientific novelty lies in the development of a generalized process model for explanations, which enables detailing the sequence of states in decision-making processes, identifying key actions, and contextual conditions for their execution. The model creates conditions for determining process actions with the highest possibility values, thus having the most significant impact on intelligent system decisions. The proposed method for constructing process-oriented explanations in intelligent information systems based on possibilistic causal dependencies includes stages such as forming temporal and causal rules, calculating possibility and necessity metrics, and presenting explanations considering state attributes in the decision-making process. The method is designed to support internal users of intelligent systems by providing interpretable representations of decision-making processes.

Keywords: explanation, intelligent system, explanatory artificial intelligence, temporal dependency, causal dependency, possibility, artificial intelligence system, process modeling.

Tab. 2. Fig. 0. Ref.: 22 items.

УДК 004.8:004.9

Розробка темпоральної моделі представлення знань щодо бізнес-процесів першого рівня зрілості процесного управління / О.В. Чала, Є.О. Богатов. АСУ та прилади автоматики. 2024. № 182. С. 34-41.

Предметом дослідження є бізнес-процеси першого рівня зрілості процесного управління. Метою є розробка темпорального підходу до представлення знань з метою створення умов для побудови знання-орієнтованої моделі бізнес-процесів керованого рівня зрілості. Для досягнення мети вирішуються такі завдання: структуризація відмінностей між зрілими та незрілими бізнес-процесами з урахуванням темпоральних аспектів; розробка моделі представлення знань щодо бізнес-процесів першого рівня зрілості процесного управління.

Виконано порівняння зрілих та незрілих бізнес-процесів. Результати порівняння свідчать, що використання у зрілих бізнес-процесах знань для визначення альтернативних послідовностей дій, які дозволяють оперативно адаптувати такі бізнес-процеси до змін у вимогах користувачів та до зовнішніх впливів.

Запропоновано темпоральну модель представлення знань щодо бізнес-процесу на основі зважених темпоральних правил, яка дозволяє оцінювати альтернативні варіанти виконання процесів на основі порівняння ваг цих правил. Модель забезпечує швидку адаптацію бізнес-процесів шляхом вибору альтернативи з максимальною вагою за умови виконання обмежень поточного зовнішнього середовища.

Ключові слова: інформаційна система процесного управління, бізнес-процеси, рівні зрілості процесного управління, темпоральні знання, зважені темпоральні правила.

Табл. 2. Іл. 1. Бібліогр.: 24 назв.

UDC 004.8:004.9

Development of a temporal model for knowledge representation regarding business processes of the first level of process management maturity/ O.V. Chala, Y.O. Bogatov. Management Information System and Devices. 2024. № 182. P. 34-41.

The subject of the research is business processes at the first level of process management maturity. The aim is to develop a temporal approach to knowledge representation in order to create conditions for building a knowledge-oriented model of business processes at the managed maturity level. To achieve this goal, the following tasks are addressed: structuring the differences between mature and immature business processes considering temporal aspects; developing a model for knowledge representation regarding business processes at the first level of process management maturity.

A comparison between mature and immature business processes was conducted. The results indicate that the use of knowledge in mature business processes to determine alternative sequences of actions allows such processes to be quickly adapted to changes in user requirements and external influences.

A temporal model for knowledge representation regarding business processes based on weighted temporal rules has been proposed, which allows evaluating alternative process execution options based on comparing the weights of these rules. The model ensures quick adaptation of business processes by selecting the alternative with the highest weight, provided that current environmental constraints are met.

Keywords: process management information system, business processes, process management maturity levels, temporal knowledge, weighted temporal rules.

Tab. 2. Fig. 1. Ref.: 24 titles.

УДК 004.02; 004.51

Вибір підмножини методів розробки інтерфейсу мобільного застосунку для вивчення мов / Н.В. Васильцова, А.С. Бурковська. АСУ і прилади автоматики. 2024. № 182. С. 42-66.

Проведено аналіз відмінності процесу розробки інтерфейсу в мобільних застосунках для вивчення мов від процесу проектування інтерфейсу користувача для інших різновидів ІТ-продуктів. Аналіз показав, що мобільні застосунки, які використовуються для вивчення мов, мають специфічні характеристики та принципи розробки інтерфейсу користувача.

Основою відмінності є специфіка технічної побудови мобільних пристроїв та контекст їх використання. В рамках мобільних застосунків з вивчення мов виділені дві категорії таких

застосунків: спеціалізовані, які розробляються для курсів з вивчення мов; загальні, які розробляються для всіх, хто хоче вивчити мову самостійно.

Особливість виконання етапу проектування інтерфейсу в мобільних застосунках для вивчення мов та необхідність врахування на цьому етапі відмінностей інтерфейсу користувача обумовлюють необхідність аналізу існуючих методів розробки інтерфейсу для виявлення особливостей залучення користувачів у процес розробки мобільних застосунків.

Розглянуто основні існуючі методи розробки інтерфейсу, які використовуються під час проектування інтерфейсів мобільних застосунків для вивчення мов.

Встановлено, що жоден з цих методів не має значних переваг, які б дозволили рекомендувати цей метод як найбажаніший інструмент розробки користувацького інтерфейсу мобільного застосунку. Для вирішення задачі визначення та планування дій відносно необхідних методів розробки інтерфейсів мобільних застосунків для вивчення мов запропоновано адаптувати метод експертного оцінювання. Під час адаптації методу експертного оцінювання до особливостей вирішення задачі основна увага приділена вибору методу збору оцінок окремих експертів. Для цього запропоновано створення опитувальників для обох груп експертів (користувачів і команди розробки). Проведено апробацію адаптованого методу під час вирішення зазначеної задачі в IT-проекті розробки мобільного застосунку з вивчення мов.

Ключові слова: мобільний застосунок для вивчення мови, інтерфейс користувача, експертне оцінювання.

Табл. 16. Іл. 2. Бібліогр.: 21 назв.

UDC 004.02; 004.51

Selection of a subset of methods for developing the interface of a mobile application for language learning / N.V. Vasylytsova, A.S. Burkovska. Management Information System and Devices. 2024. № 182. P. 42-66.

An analysis of the difference between the process of interface development in mobile applications for learning languages and the process of designing a user interface for other types of IT products is carried out. The analysis showed that mobile applications used for language learning have specific characteristics and principles of user interface development.

The basis of the difference is the specifics of the technical design of mobile devices and the context of their use. Within the framework of mobile language learning applications, there are two categories of such applications: specialized, which are developed for language learning courses; general ones, which are developed for everyone who wants to learn the language on their own.

The peculiarity of the interface design stage in mobile applications for learning languages and the need to take into account the differences of the user interface at this stage necessitate the analysis of existing methods of interface development to identify the features of user involvement in the process of developing mobile applications.

The main existing methods of interface development, which are used in the design of interfaces of mobile applications for language learning, are considered.

It has been established that none of these methods has significant advantages that would allow recommending this method as the most desirable tool for developing the user interface of a mobile application. To solve the problem of defining and planning actions regarding the necessary methods of developing interfaces of mobile applications for learning languages, it is proposed to adapt the method of expert evaluation. When adapting the method of expert evaluation to the specifics of solving the problem, the main attention is paid to the choice of the method of collecting assessments of individual experts. For this purpose, it is proposed to create questionnaires for both groups of experts (users and the development team). Approbation of the adapted method in solving this problem in the IT project for the development of a mobile application for language learning was carried out.

Keywords: mobile language learning app, user interface, peer review.

Tab. 16. Fig. 2. Ref.: 21 items.

УДК 004.8:004.9

Уточнення ментальної моделі рішення на основі доповнення вхідних даних в задачі

формування пояснень в інтелектуальній системі / С.Ф. Чалий, І.О. Лещинська. АСУ та прилади автоматики. 2024. № 182. С. 66-72.

Предметом дослідження є процес побудови ментальної моделі рішення для користувача інтелектуальної системи. Метою є розробка підходу до уточнення ментальної моделі користувача на основі принципу доповнення вхідних даних для того, щоб врахувати як позитивні, так і негативні аспекти рішення. Задачі: структуризація принципу доповнення вхідних даних; розробка методу уточнення ментальної моделі користувача інтелектуальної системи на основі доповнення вхідних даних. Наукова новизна отриманих результатів полягає в такому. Удосконалено принципи побудови ментальних моделей з урахуванням особливостей формування пояснень в інтелектуальних системах. Запропоновано метод уточнення ментальної моделі на основі аналізу відгуків користувачів, що дозволяє доповнити модель негативними властивостями рішення. Метод містить етапи аналізу відгуків користувачів, оцінки важливості виявлених властивостей рішення, доповнення ментальної моделі з урахуванням негативних властивостей та балансування позитивних і негативних аспектів рішення. Використання даного методу при побудові пояснення дає можливість формувати повніші та збалансованіші пояснення, які враховують як переваги, так і обмеження рішення інтелектуальної системи. Експериментальна перевірка запропонованого методу показала можливість автоматизованого доповнення ментальної моделі рішення на основі аналізу відгуків користувачів інтелектуальної системи.

Ключові слова: пояснення, інтелектуальна система, самопояснювальний штучний інтелект, система штучного інтелекту, ментальна модель, користувачі інтелектуальної системи, доповнення вхідних даних.

Табл. 2. Іл. 0. Бібліогр.: 20 назв.

UDC 004.8:004.9

Refinement of the mental model of a solution based on input data augmentation in the task of generating explanations in an intelligent system / S.F. Chaly, I.O. Leshchynska. Management Information System and Devices. 2024. № 182. P. 66-72.

The subject of the research is the process of constructing a mental model of a solution for an intelligent system user. The aim is to develop an approach to refine the user's mental model based on the principle of input data augmentation to account for both positive and negative aspects of the solution. Objectives: structuring the principle of input data augmentation; development of a method for refining the mental model of an intelligent system user based on input data augmentation. The scientific novelty of the obtained results is as follows. The principles of constructing mental models have been improved, taking into account the peculiarities of generating explanations in intelligent systems. A method for refining the mental model based on the analysis of user feedback is proposed, which allows supplementing the model with negative characteristics of the solution. The method includes stages of analyzing user feedback, assessing the importance of identified solution characteristics, supplementing the mental model with negative characteristics, and balancing positive and negative aspects of the solution. The use of this method in constructing explanations enables the formation of more comprehensive and balanced explanations that consider both the advantages and limitations of the intelligent system's solution. Experimental verification of the proposed method demonstrated the possibility of automated augmentation of the mental model of the solution based on the analysis of intelligent system user feedback.

Keywords: explanation, intelligent system, self-explanatory artificial intelligence, artificial intelligence system, mental model, intelligent system users, input data augmentation.

Tab. 2. Fig. 0. Ref.: 20 items.

УДК 004.91; 004.8

Інформаційна технологія виявлення термінів та артефактів проєкту у вимогах до інформаційної системи / М.В. Євланов, Б.І. Мороз, Д.М. Мороз, В.В. Лучицький. АСУ та прилади автоматики. 2024. № 182. С. 73-93.

Об'єктом дослідження є діяльність «Визначення системних вимог» процесу «Визначення вимог до системи». Визначено, що одним з найкращих способів здійснення цієї діяльності є

застосування людино-машинних або машинних методів визначення та аналізу системних вимог. Але існуючі інструментальні засоби визначення та управління вимогами до ІТ-продуктів не дозволяють автоматизувати виконання відповідних робіт. Тому проведення досліджень із застосування методів штучного інтелекту і, зокрема, методів обробки природної мови для визначення та аналізу текстових публікацій системних вимог є актуальним.

Головною метою дослідження є розробка інформаційної технології виявлення термінів та артефактів проєкту у вимогах до інформаційної системи. Було визначено формальну основу цієї технології, яка базується на застосуванні методів лематизації. Розроблено опис архітектури інформаційної технології. Визначено технологічний стек для розробки сервісу, що реалізує цю технологію. Розглянуто особливості розробки найважливіших програмних модулів цього сервісу.

Проведено експериментальну перевірку отриманих результатів розробки інформаційної технології. Для цієї перевірки було застосовано українську та англійську локалізації інформаційної системи «Футбол». Ця система реалізує мінімально необхідний набір функцій які дозволяють автоматизувати вирішення задач планування та обліку проведення футбольних матчів.

Під час експериментальної перевірки було встановлено, що сервіс, який реалізує розроблену інформаційну технологію, успішно вирішує задачу виявлення термінів та артефактів проєкту у вимогах до інформаційної системи і забезпечує релевантні результати вирішення цієї задачі для перекладів однієї і тієї ж вимоги різними мовами.

Ключові слова: інформаційна система, системна вимога, NLP, лематизація, артефакт, технологічний стек, Python.

Табл. 2. Іл. 12. Бібліогр.: 25 назв.

UDC 004.91; 004.8

Information technology for identifying terms and project artifacts in the requirements for the information system / M.V. Yevlanov, B.I. Moroz, D.M. Moroz, V.V. Luchytskyi. Management Information System and Devices. 2024. № 182. P. 73-93.

The object of the study is the activity «Determining system requirements» of the «Determining system requirements» process. It has been determined that one of the best ways to carry out this activity is to use human-machine or machine-based methods of defining and analyzing system requirements. But the existing instrumental means of defining and managing the requirements for IT products do not allow automating the execution of the relevant work. Therefore, conducting research on the application of artificial intelligence methods and, in particular, natural language processing methods for the definition and analysis of textual publications of system requirements is relevant.

The main goal of the research is the development of information technology for identifying terms and artifacts of the project in the requirements for the information system. The formal basis of this technology was defined, which is based on the application of lemmatization methods. A description of the architecture of information technology has been developed. A technological stack for the development of a service that implements this technology has been defined. Features of the development of the most important software modules of this service are considered.

An experimental verification of the obtained results of information technology development was carried out. Ukrainian and English localizations of the «Football» information system were used for this check. This system implements the minimally necessary set of functions that allow automating the solution of the tasks of planning and accounting for football matches.

During the experimental verification, it was established that the service, which implements the developed information technology, successfully solves the problem of identifying project terms and artifacts in the requirements for the information system and provides relevant results for solving this problem for translations of the same requirement in different languages.

Keywords: information system, system requirement, NLP, lemmatization, artifact, technological stack, Python.

Tab. 2. Fig. 12. Ref.: 25 items.

УДК 004.8:004.9

Швидка класифікація в online- та nearline-режимах в умовах класів, що перетинаються /

Є.В. Бодянський, О.С. Чала. АСУ і прилади автоматики. 2024. № 182. С. 93-100.

Предметом дослідження є процес швидкої класифікації в умовах класів, що перетинаються. Метою є розробка підходу до швидкої класифікації, який поєднує online- та nearline-режими для підвищення точності класифікації в умовах класів, що перетинаються. Задачі: розробка багатошарової нейронної мережі з ядрними дзвонуватими функціями активації для роботи в online-режимі; розробка адаптивної нейро-фаззі системи для роботи в nearline-режимі. Наукова новизна отриманих результатів полягає в такому. Вперше запропоновано підхід до швидкої класифікації, який поєднує online- та nearline-режими для вирішення задач уточнення границь між класами, адаптації до змін у розподілі вхідних даних, усунення дисбалансу класів та видалення шумових точок. Запропоновану багатошарову нейронну мережу з ядрними дзвонуватими функціями активації для вирішення задачі класифікації в умовах класів, що перетинаються, у online-режимі. Мережа використовує рекурентний алгоритм навчання, що дає можливість відмовитись від процедури зворотного поширення помилок та уникнути ефекту «зникаючого градієнта». У практичному плані мережа забезпечує адаптацію системи до складних границь між класами та створює умови для адаптації до змін у розподілі вхідних даних. Розроблено адаптивну нейро-фаззі систему для класифікації даних у матричній формі, яка використовує гібридне комбіноване навчання. Навчання реалізовано в послідовному online- або nearline-режимі при надходженні нових даних. Експериментальна перевірка запропонованого підходу показала можливість збереження точності класифікації в умовах класів, що перетинаються, та змін у розподілі вхідних даних.

Ключові слова: швидка класифікація, класи, що перетинаються, online-режим, nearline-режим, нейронна мережа, нейро-фаззі система, адаптивне навчання.

Табл. 2. Іл. 1. Бібліогр.: 24 назв.

UDC 004.8:004.9

Fast classification in online and nearline modes in conditions of overlapping classes / Ye. V. Bodyanskiy, O.S. Chala. Management Information System and Devices. 2024. № 182. P. 93-100.

The subject of the research is the process of fast classification under overlapping classes conditions. The aim is to develop an approach to fast classification that combines online and nearline modes to improve classification accuracy in overlapping classes conditions. Objectives: development of a multilayer neural network with kernel bell-shaped activation functions for online mode operation; development of an adaptive neuro-fuzzy system for nearline mode operation. The scientific novelty of the obtained results is as follows. For the first time, an approach to fast classification is proposed that combines online and nearline modes to solve problems of refining boundaries between classes, adapting to changes in input data distribution, eliminating class imbalance, and removing noise points. A multilayer neural network with kernel bell-shaped activation functions is proposed for solving the classification problem under overlapping classes conditions in online mode. The network uses a recurrent learning algorithm, which allows avoiding the backpropagation procedure and the «vanishing gradient» effect. In practical terms, the network provides system adaptation to complex boundaries between classes and creates conditions for adapting to changes in input data distribution. An adaptive neuro-fuzzy system for classifying data in matrix form has been developed, which uses hybrid combined learning. Learning is implemented in sequential online or nearline mode as new data arrives. Experimental verification of the proposed approach showed the possibility of maintaining classification accuracy under overlapping classes conditions and changes in input data distribution.

Keywords: fast classification, overlapping classes, online mode, nearline mode, neural network, neuro-fuzzy system, adaptive learning.

Tab. 2. Fig. 1. Ref.: 24 items.

**ПРАВИЛА ОФОРМЛЕННЯ СТАТЕЙ
У ВСЕУКРАЇНСЬКОМУ МІЖВІДОМЧОМУ НАУКОВО-ТЕХНІЧНОМУ
ЗБІРНИКУ
«АВТОМАТИЗОВАНІ СИСТЕМИ УПРАВЛІННЯ ТА ПРИЛАДИ АВТОМАТИКИ»**

1. Загальні вимоги

До розгляду приймаються раніше не опубліковані статті українською та англійською мовами. Статті англійською мовою подаються разом з українськомовним варіантом. Статті, перекладені англійською за допомогою комп'ютерного перекладача та не відредаговані належним чином, не розглядаються.

Наукова стаття, яка подається до розгляду, має бути структурована та містити всі основні частини, характерні для наукової статті:

- постановка проблеми у загальному вигляді та її зв'язок з важливими науковими та практичними задачами;
- аналіз останніх досліджень та публікацій, у яких розпочато вирішення даної проблеми та на які спирається автор, виділення невирішених раніше частин загальної проблеми;
- формулювання цілей статті (постановка задачі);
- подання основного матеріалу досліджень з повним обґрунтуванням отриманих результатів;
- висновки даного дослідження та перспективи подальших досліджень у даному напрямку;
- перелік посилань (References).

2. Вимоги до структури рукопису

Структурно матеріали статті поділяються на такі елементи:

- УДК;
- прізвища та ініціали авторів статті;
- заголовок статті;
- анотація до статті;
- основний текст статті;
- перелік посилань;
- дата надходження статті до редколегії збірника;
- відомості про авторів статті;
- реферати українською та англійською мовами.

Бажаний порядок та зміст розділів основного тексту статті:

а) розділ 1 «Вступ», в якому визначається проблема у загальному вигляді та її зв'язок з важливими науковими та практичними задачами;

б) розділ 2 «Аналіз літературних джерел та визначення проблеми дослідження», в якому наводяться результати аналізу останніх досліджень та публікацій, де розпочато вирішення даної проблеми та на які спирається автор, виділяються невирішені раніше частини загальної проблеми дослідження та конкретизується головна проблема дослідження у даній статті;

в) розділ 3 «Мета і задачі дослідження», в якому наводяться описи мети дослідження та задач дослідження, вирішення яких дозволяє досягти визначеної раніше мети дослідження;

г) розділ 4 «Матеріали і методи дослідження», в якому наводяться описи формального апарату та раніше проведених експериментальних досліджень, які будуть використані у подальшому тексті статті;

д) розділ 5 «Результати дослідження», в якому структуровано наводяться результати вирішення сформульованих у розділі 3 окремих задач дослідження (теоретичних та експериментальних);

е) розділ 6 «Обговорення результатів дослідження», в якому наводяться: опис особливостей отриманих результатів дослідження та їхньої відмінності від результатів попередніх досліджень у відповідній галузі; опис переваг отриманих результатів перед існуючими; опис недоліків і обмежень, які утруднюють використання отриманих результатів дослідження; опис подальших перспектив проведення досліджень за цим напрямом;

ж) розділ 7 «Висновки», в якому наводяться стислі описи отриманих результатів вирішення окремих задач дослідження та загальний висновок про досягнення поставленої у розділі 3 мети дослідження.

Заголовки окремих розділів основного тексту статті можуть змінюватися відповідно до змісту конкретної статті.

Розділи основного тексту статті, перелік посилань, дата надходження статті до редколегії збірника та відомості про авторів статті відокремлюються один від одного одним порожнім рядком.

3. Вимоги до оформлення рукопису

До розгляду приймаються матеріали статей обсягом не менше 5 повних сторінок (з урахуванням рисунків і таблиць).

Матеріали статті повинні бути набраними у редакторі MS Word. Припустимі формати файлу з матеріалами статті – .doc або .docx.

Формат сторінки – А4 (210x297 мм). Поля знизу, зверху, справа, зліва – 3 см.

Основний текст статті набирається шрифтом Times New Roman, кегль 11, інтервал – 1,1, абзацний відступ – 8 мм, інтервали перед і після – 0 мм, вирівнювання по ширині.

Для УДК – шрифт Times New Roman, кегль 11, інтервал – 1,1, абзацний відступ – 8 мм, інтервал перед – 0 мм, інтервал після – 6 мм, вирівнювання по ширині.

Для прізвищ та ініціалів авторів статті – шрифт Times New Roman, кегль 11, інтервал – 1,1, абзацний відступ – 8 мм, інтервали перед і після – 6 мм, вирівнювання по ширині.

Для заголовка статті – шрифт Times New Roman, кегль 11, напівжирний, інтервал – 1,1, абзацний відступ – 8 мм, інтервали перед і після – 6 мм, вирівнювання по ширині.

Для анотації – шрифт Times New Roman, кегль 10, інтервал – 1,1, відступ зліва – 0,8 см, абзацний відступ – 8 мм, інтервал перед – 6 мм, інтервал після – 0 мм, вирівнювання по ширині.

Для заголовків таблиць – шрифт Times New Roman, кегль 10, інтервал – 1,1, абзацного відступу немає, інтервали перед і після – 0 мм, слово «Таблиця» та її номер – з вирівнюванням вправо, назва таблиці (якщо вона є) – з вирівнюванням по центру.

Для підписових підписів – шрифт Times New Roman, кегль 10, інтервал – 1,1, абзацного відступу немає, інтервали перед і після – 0 мм, вирівнювання по центру.

Для переліку посилань та відомостей про авторів – шрифт Times New Roman, кегль 9, інтервал – 1,1, абзацний відступ – 8 мм, інтервали перед і після – 0 мм, вирівнювання по ширині.

Для рефератів – шрифт Times New Roman, кегль 10, інтервал – 1,1, абзацний відступ – 8 мм, інтервали перед і після – 0 мм, вирівнювання по ширині.

Формули набираються у редакторі формул Microsoft Equation або MathType, розташовуються у центрі робочого поля, нумерація – з правої сторони поля. Для цього необхідно весь рядок розташувати справа, а потім вирівняти формулу табуляціями так, щоб вона розташовувалася по центру. Відступ зверху і знизу – по 6 пунктів. Нумерація формул усередині кожної статті наскрізна.

Формули, а також їхні складові, присутні у тексті, набираються з такими параметрами (див. рис. 1).

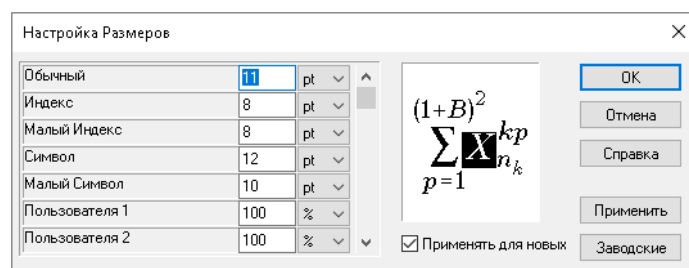


Рис. 1. Параметри настроювання розмірів редактора формул MathType

Кожна таблиця виконується та розташовується в тексті одразу після посилання на неї. Усі таблиці у статті обов'язково нумеруються, незважаючи на їх кількість. Таблиця відокремлюється від попереднього та наступного тексту (таблиці, рисунку тощо) одним порожнім рядком.

Дані всієї таблиці набираються шрифтом розміром 10 пунктів, розміщуються по центру; у випадках, коли необхідно показати розрядність, – вирівнювання за знаком. Товщина сітки таблиці – 1 пункт. Приклад оформлення таблиці наведено на рис. 2.

Таблиця 1

Множина описів сутностей функціональної задачі

ID	Найменування
1	Academic_load
2	Academic
3	Department
4	Individual_plan
5	Academic_section

Рис. 2. Приклад оформлення таблиці у тексті статті.

Бажано таблицю зі сторінки на сторінку не переносити. Якщо таблиця не може розміститися на сторінці, її поділяють на частини. У кожній частині таблиці повторюють її головку та боковик або замінюють їх відповідно номерами колонок або рядків, нумеруючи їх арабськими цифрами на першій частині таблиці. Слово «Таблиця» подається лише над першою її частиною. Над наступними її частинами праворуч друкується: «Продовження таблиці», а на останній – «Кінець таблиці», в усіх випадках вказується номер таблиці.

Кожен рисунок виконується та розташовується в тексті одразу після посилання на нього. Усі рисунки в статті обов'язково нумеруються, незважаючи на їх кількість. Необхідно вставляти рисунки у текст як графічні об'єкти (файли з розширенням .bmp, .jpg, .tiff чи .png, якість не менше 300 dpi), об'єкти MS Word або MS Visio.

Рисунок відокремлюється від попереднього та наступного тексту (таблиці, рисунку тощо) одним порожнім рядком.

Кожен рисунок повинен мати підписуноківий підпис, в якому вказується номер та, у випадку необхідності, назва рисунку. Якщо рисунок займає менше 50 % ширини робочого поля, то можна зробити обтікання рисунку текстом, розташувавши його ліворуч або праворуч від робочого поля. Приклад рисунку з підписом наведений на рис. 3.

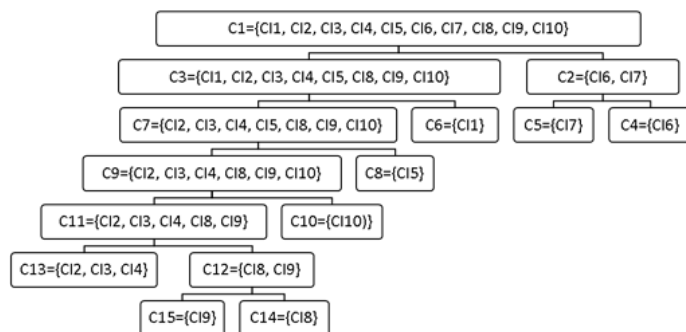


Рис. 3. Приклад виконання рисунку та підписуноківого підпису

Посилання на літературні та електронні джерела у тексті статті позначаються у квадратних дужках [1]. До переліку посилань включаються тільки ті роботи, на які посилається автор статті. Посилання на неопубліковані роботи не допускаються.

Для оформлення переліку посилань слід використовувати один з таких шаблонів:

а) шаблон IEEE (автоматичне оформлення за шаблоном [IEEE](https://www.citethisforme.com/ieee/source-type) <https://www.citethisforme.com/ieee/source-type>);

б) положення ДСТУ 8302:2015 «Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання» та ДСТУ 3582:2013 «Інформація та документація. Бібліографічний опис. Скорочення слів і словосполучень українською мовою. Загальні вимоги та правила».

Кожен з цих шаблонів слід використовувати для оформлення усіх елементів переліку посилань. Використання двох шаблонів для оформлення одного й того ж переліку посилань неприпустимо.

Кожне посилання у переліку посилань наводиться за порядком появи цих посилань у тексті статті.

У переліку посилань бажано використовувати посилання на сучасні публікації, вік яких не перевищує п'яти років у момент подачі статті до редакції. Крім того, під час формування переліку посилань статті необхідно дотримуватися такого розподілу: самоцитування – до 20 %, цитування зарубіжних публікацій – не менше 50%.

Відомості про авторів слід наводити українською та англійською мовами. У відомості про авторів слід включати: повні прізвище, ім'я та по-батькові; вчений ступінь (за наявності); вчене звання (за наявності); посаду; країну, місто; e-mail (вкрай бажано вказувати корпоративний e-mail, можна вказувати кілька e-mail, на які ви бажаєте отримувати повідомлення від редакції та читачів, які можуть зацікавитися вашою статтею); ORCID.

Реферат набирається українською та англійською мовами. Реферат повинен бути змістовним, дотримуватися логіки опису результатів у статті та давати можливість встановити її основний зміст. Реферат не повинен містити формул та рисунків. Необхідні символи в рефераті необхідно додавати через функцію вставки символів.

Реферат містить: УДК, назву статті (напівжирним шрифтом), ініціали та прізвища авторів (курсивом), текст (не менше 1800 друкованих знаків з пробілами та ключовими словами), ключові слова, кількість таблиць, рисунків та посилань у статті.

Ключові слова повинні містити до 10 слів, а не словосполучень, без використання аббревіатур, в іменному відмінку, розділятися крапкою з комою.

Реферати надаються до редколегії разом із статтею у вигляді окремого файлу.

4. Правила надсилання статей та подальшої взаємодії з редакційною колегією збірника

До редколегії збірника «АСУ та прилади автоматики» слід надсилати такі матеріали:

- файл у форматі .doc або .docx з текстом статті українською мовою;
- файл у форматі .doc або .docx з текстом статті англійською мовою (якщо автори бажають опублікувати статтю у збірнику англійською мовою);
- файл (у форматі .doc або .docx з текстами рефератів статті українською та англійською мовами;
- відскановану копію експертного висновку з дозволом опублікувати матеріали статті у відкритому друку. В разі потреби експертні висновки для авторів – співробітників (студентів, аспірантів тощо) ХНУРЕ можуть оформлюватися редколегією централізовано.

Матеріали статей надсилати електронною поштою – за адресою maksim.ievlanov@nure.ua.

Кожна надіслана в редакцію стаття після проходження рецензування і при позитивному рішенні редколегії буде надрукована в найближчому випуску збірника. Для цього авторам від імені редколегії надсилається ліцензійний договір, який закріплює право першої публікації статті у збірнику «АСУ та прилади автоматики». Автори статті повинні підписати цей ліцензійний договір та завірити свої підписи печаткою організації, в якій вони працюють. Підписаний ліцензійний договір автори статті надсилають на адресу редколегії збірника.

Відповідальний випусковий В.М. Левикін
Редактор О.Є. Неумивакіна
Комп'ютерна верстка М.В. Євланов, О.Є. Неумивакіна
Дизайн обкладинки номера за участю Є.Чех

Підп. до друку 04.12.2024. Формат 60x841/8. Умов. друк. арк.
Обл.-вид. арк. 13,4. Тираж 300 прим.
Зам. № 144. Ціна договірна.

Харківський національний університет радіоелектроніки (ХНУРЕ).
Україна, 61166, Харків, пр. Науки, 14

Оригінал-макет підготовлено у редакційно-видавничому відділі ХНУРЕ,
Україна, 61166, Харків, пр. Науки, 14

Збірник віддруковано в ТОВ «ДРУКАРНЯ МАДРИД»
61024, м. Харків, вул. Гуданова, 18
Тел.: +38(057)7565325
Свідоцтво суб'єкта видавничої справи
Серія ДК № 4399 від 27.08.2012 р.
www.madrid.in.ua e-mail:info@madrid.in.ua