



А. І. Пукач, В. М. Теслюк

Національний Університет «Львівська Політехніка», Львів, Україна

ORCID: <https://orcid.org/0009-0001-8563-3311> – А. І. Пукач

<https://orcid.org/0000-0002-5974-9310> – В. М. Теслюк



andriipukach@gmail.com

МОДЕЛЬ ДЕКОМПОЗИЦІЇ КОМПЛЕКСНОЇ ПІДТРИМКИ ПРОГРАМНИХ ПРОДУКТІВ НА ОСНОВІ ВИМІРІВ СПРИЙНЯТТЯ

Основною проблемою процесу програмного забезпечення є необхідність розроблення моделі декомпозиції комплексної підтримки програмних продуктів, зокрема на основі вимірів сприйняття, сформованих релевантними факторами впливу суб'єктивізації сприйняття досліджуваних об'єктів підтримки (як самих підтримуваних програмних комплексів, так і процесів їх комплексної підтримки) відповідними суб'єктами взаємодії. Метою роботи є розроблення відповідної моделі декомпозиції комплексної підтримки програмних продуктів на основі вимірів сприйняття. Адже комплексна підтримка програмних продуктів є надзвичайно багатогранним і складно структурованим процесом, що включає в себе, фактично, будь-які активності, спрямовані на забезпечення життєдіяльності та функціонування будь-якого підтримуваного програмного продукту. Отож, саме тому, комплексна підтримка програмних продуктів потребує ефективних та дієвих механізмів та інструментів декомпозиції, одним з яких є, власне, запропонована модель декомпозиції на основі вимірів сприйняття. Таким чином, розроблена модель дає змогу здійснити декомпозиції комплексної підтримки програмних продуктів, забезпечуючи тим самим можливість більш ефективного дослідження процесів комплексної підтримки на кожному із складових рівнів декомпозиції. При цьому, в якості основних методів дослідження використовуються базові методи класифікації, а також методи декомпозиції. Основні результати дослідження полягають в отриманні моделей декомпозиції комплексної підтримки досліджуваних програмних продуктів на основі вимірів сприйняття. Також, в рамках проведеного дослідження, розв'язана прикладна практична задача ідентифікації домінуючого виміру сприйняття досліджуваного середовища комплексної підтримки програмного продукту. Задекларовані основні перспективи подальшого застосування розробленої моделі декомпозиції на основі вимірів сприйняття в контексті науково-прикладної проблематики автоматизації й інтелектуалізації комплексної підтримки програмних продуктів.

Ключові слова: модель, програмний продукт, комплексна підтримка, декомпозиція, сприйняття, виміри.

А. І. Пукач, В. М. Теслюк

Lviv Polytechnic National University, Lviv, Ukraine

SOFTWARE PRODUCTS' COMPREHENSIVE SUPPORT DECOMPOSITION MODEL BASED ON PERCEPTION DIMENSIONS

Annotation. The main problem is the need to develop a comprehensive support model for software products, based on the dimensions of perception formed by the relevant impact factors influencing the subjectivisation of the researched support objects (i.e. the supported software complexes themselves and the processes of their comprehensive support) by the relevant interaction subjects. The purpose of the research is to develop an appropriate decomposition model for the comprehensive support of software products based on perception dimensions. Comprehensive support of software products is an extremely multifaceted and complex process that encompasses all activities aimed at ensuring the existence and operation of supported software products. That is why comprehensive support of software products requires effective and efficient decomposition mechanisms and tools, one of which is, in fact, the proposed decomposition model that is based on perception dimensions. Thus, the developed model enables the decomposition of the complex support of software products, providing the opportunity for more effective research into the comprehensive support processes at each component level. At the same time, basic classification and decomposition methods are used as the primary research tools. The main results of the research consists in obtaining a decomposition models (of comprehensive support of the investigated software products) based on perception dimensions. The practical applied task of identifying the dominant

perception's dimension (of the researched environment of investigated software product's comprehensive support) has been solved in scope of current research. The main prospects for further application of the developed decomposition model (based on perception dimensions) in the context of scientific and applied problematics of automation and intellectualization of software products' comprehensive support are declared.

Keywords: model, software product, comprehensive support, decomposition, perception, dimensions.

Вступ / Introduction

Кількість та складність розроблюваного програмного забезпечення продовжує природно зростати в зв'язку із складністю задач, для вирішення яких вони покликані. На ринку інформаційних технологій (IT) щороку з'являється все більше тих продуктів, які переходять з фази розробки у фазу використання, при цьому як перші, так і останні, потребують комплексної підтримки, що включає в себе надзвичайно велику кількість найрізноманітніших активностей щодо забезпечення та підтримування життєдіяльності та функціонування цих програмних продуктів та середовищ (інфраструктури) їх існування. Найпопулярнішими складовими комплексної підтримки є клієнтська підтримка [1,2], тестування програмного забезпечення [3,4], методології Agile [5,6] та DevOps [7,8], та багато інших.

Враховуючи надзвичайну складність комплексної підтримки, виникає природна необхідність її декомпозиції. Зокрема, автори дослідження [9] розглядають деякі ключові методи та техніки моделей декомпозиції в контексті проблематики дизайну складних програмних комплексів та систем, що включають, в тому числі, мульти-модальні підходи, котрі передбачають такі складові як: моделювання точок зору; моделювання аспектів; моделювання ролей; та моделювання суб'єктів. Додатково, дослідження [10] репрезентує напрацювання авторів, основою якого є розроблений напівавтоматичний підхід, який розкладає моноліт на безсерверні функції шляхом аналізу бізнес-логіки, притаманної інтерфейсу програми, використовуючи двоетапну стратегію рефакторингу, де спочатку виконується грубозерниста декомпозиція, а потім дрібнозерниста, завдяки чому процес декомпозиції спрощено на більш дрібні етапи та адаптовано для створення рішення на рівні мікросервісу або функції, забезпечивши, при цьому, можливість цілісного інтегрування розробленого підходу в DevOps. Крім того, в роботі [11] авторський колектив розглядає декомпозицію програмного забезпечення та пов'язані з ним аспекти модульності, необхідні для розробки сучасних архітектур програмного забезпечення, досліджуючи, також, проблематику так званого «корисного погіршення» (англ. «useful deterioration») для вдосконалення дизайну розробки програмних продуктів. Автори праці [12] реалізували комплексне дослідження проблематики декомпозиції монолітних програмних продуктів на

мікросервісній архітектурі, а також розробили відповідну структуру декомпозиції з моноліту на мікросервіси (M2MDF), яка визначає основні фази та ключові елементи декомпозиції.

Іншою складовою дослідження є персоніфікація (або «кастомізація») складових комплексної підтримки програмних продуктів. Зокрема, в роботі [13] автори досліджують проблематику кастомізації систем штучного інтелекту (як складових кластера інформаційних систем – IC), що, на відміну від існуючих IC, потребують відмінної концептуалізації, запропонованої авторами, яка складається з таких чотирьох рівнів: даних, моделей, алгоритмів, інфраструктур. Дослідження [14] зосереджене на розробці дійсної та надійної моделі налаштування програмного забезпечення для якості (англ. «Software as a Service» – SaaS), яка складається із загальних типів налаштування програмного забезпечення та списку загальних практик для кожного типу налаштування в багатокористувачькому контексті SaaS, а також ключових атрибутів якості програм SaaS, пов'язаних із налаштуванням. Разом з тим, автори праці [15], спираючись на попередньо розроблену модель налаштування програмного забезпечення для якості SaaS, реалізували дві основні цілі цього дослідження, котрі полягали в тому, щоб визначити, чи різні підходи до налаштування програмного забезпечення мають прямий вплив на якість SaaS, а також оцінити надійність та валідність конструкції моделі. Автори праці [16] досліджують проблематику кастомізації ERP-систем (англ. Enterprise Resource Planning), зосереджуючись саме на аспектах технічних параметрів кастомізації цих систем (програмних продуктів), в тому числі в контексті переходу від модифікації коду ERP у локальних системах до налаштування інтерфейсу та інтеграції в хмарних системах ERP, а також нових технологічних можливостей як способу для клієнтів і ключових користувачів здійснювати необхідну їм кастомізацію/персоніфікацію системи.

Отже, на сьогодні, більшість напрямків комплексної підтримки програмних продуктів проходять свій індивідуальний шлях розвитку, автоматизації та інтелектуалізації. Водночас, представлення комплексної підтримки програмних продуктів, як одного цілого та неподільного, дає змогу згодом застосувати отримані результати для всіх її складових. При цьому, важливо віднайти спільні фундаментальні складові, присутні в кожному з елементів комплексної підтримки. Таким чином, виміри сприйняття є однією із таких

фундаментальних складових. Водночас, розглянуті існуючі підходи, рішення та засоби декомпозиції складових комплексної підтримки програмних продуктів, представлені у попередніх двох абзацах цього ж розділу, хоча і, безумовно, забезпечують необхідний рівень цієї декомпозиції в контексті своїх релевантних сфер дослідження та впливу, проте вони, на жаль, не забезпечують можливості врахування фактору(-ів) суб'єктивізації сприйняття підтримуваних програмних продуктів тими суб'єктами (розробники, тестувальники, фахівці DevOps, системні інженери, бізнес-аналітики тощо), котрі, власне, безпосередньо забезпечують та реалізують відповідні складові цієї комплексної підтримки.

Розроблення моделі декомпозиції на основі вимірів сприйняття забезпечує можливість отримання максимально деталізованого та достовірного представлення (а, отже, і необхідну формалізацію) процесів суб'єктивізації сприйняття досліджуваних об'єктів або процесів комплексної підтримки програмних продуктів з подальшим використанням отриманих напрацювань в контексті вирішення глобальних науково-прикладних проблем автоматизації й інтелектуалізації комплексної підтримки програмних продуктів.

Методи дослідження / Methods

В якості основних методів дослідження використано базові методи класифікації [17, 18] та методи декомпозиції [19]. Проте, використання лише цих базових існуючих методів не дає змогу відобразити всю суть досліджуваної проблеми (декомпозиції комплексної підтримки програмних продуктів на основі вимірів сприйняття) та її вирішення. Саме тому, в якості додаткової обов'язкової компоненти вирішення поставленої проблеми використано метод формування мультифакторних портретів суб'єктів підтримки програмних комплексів [20].

Розглянемо основний вклад кожного кластера використаних методів для розв'язання поставленої задачі.

Зокрема, базові методи класифікації необхідні для чіткого ранжування елементів (в даному конкретному випадку: існуючих вимірів сприйняття, а також релевантних факторів впливу, в контексті комплексної підтримки програмних продуктів) за їх класовою належністю до відповідних рівнів та відгалужень декомпозиції.

Водночас, базові методи декомпозиції дають змогу реалізувати фундаментальні принципи ранжування рівнів та відгалужень ієрархії структури досліджуваних об'єктів чи процесів (в даному конкретному випадку – комплексної підтримки програмних продуктів).

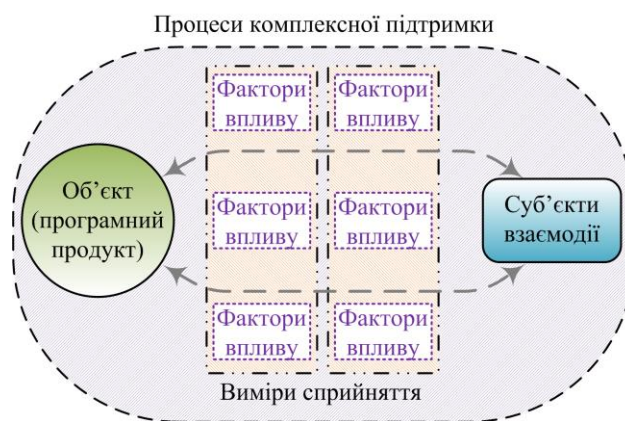
В той час метод формування мультифакторних портретів суб'єктів підтримки програмних комп-

лексів дає змогу отримати симбіоз всіх складових досліджуваної проблеми, а саме:

- об'єкта (безпосередньо найбільш підтримуваного програмного продукту);
- процесів (комплексної підтримки програмного продукту);
- суб'єктів (що безпосередньо забезпечують та реалізують комплексну підтримку програмного продукту);
- факторів впливу (що існують в рамках відповідних вимірів сприйняття та безпосередньо впливають на суб'єктивізацію сприйняття об'єкта суб'єктами взаємодії з ним);
- вимірів сприйняття (що поєднують в собі відповідні релевантні фактори впливу, та відповідають за стратегію впливу цих факторів на суб'єктивізацію сприйняття суб'єктами об'єкта, з яким вони взаємодіють).

Таким чином, застосування виключно лиш існуючих базових методів класифікації та декомпозиції, на жаль, не дає змоги розв'язати поставлену задачу, оскільки потребує наявності ще однієї складової компоненти, яка б забезпечила можливість їх комутації між собою. І в якості такої складової компоненти виступає метод формування мультифакторних портретів суб'єктів підтримки програмних комплексів.

Нижче, на рисунку 1, подано відповідну графічну інтерпретацію реалізованої (на базі відповідної трискладової компоненти) структури моделі декомпозиції комплексної підтримки програмних продуктів на основі вимірів сприйняття.



Рисунк 1 – Структура моделі декомпозиції комплексної підтримки програмних продуктів на основі вимірів сприйняття

Крім того, розроблена розгорнута класифікація та декомпозиція вимірів сприйняття комплексної підтримки програмних продуктів, представлена нижче на рисунку 2, відповідно з якою введені такі виміри комплексної підтримки програмних продуктів, а саме:

- фізичний та метафізичний;
- логічний;
- раціональний та ірраціональний;

А також введено такі відгалуження цих вимірів, що передбачають, зокрема, такі елементи класифікації як:

- структурний логічний;
- архітектурний логічний;
- системний логічний;
- технічний логічний;
- інформаційний логічний;
- операційний логічний;
- інженерний логічний;
- комунікаційний логічний;
- координаційний логічний;
- функціональний логічний;
- технологічний логічний;
- сутнісний логічний;
- ергономічний логічний;
- математичний логічний;
- ресурсний логічний;
- часовий логічний;
- бізнес логічний;
- стратегічний логічний;
- абстрактний раціональний;
- теоретичний раціональний;
- емпіричний раціональний;
- аналітичний раціональний;
- світоглядний ірраціональний;
- ментальний ірраціональний;
- національний ірраціональний;
- етнічний ірраціональний;
- культурний ірраціональний;
- субкультурний ірраціональний;
- традиційний ірраціональний;
- соціальний ірраціональний;
- освітній ірраціональний;
- фізіологічний ірраціональний;
- емоційний ірраціональний;
- чуттєвий ірраціональний;
- інтуїтивний ірраціональний;
- суб'єктивний ірраціональний.

Фізичний			Метафізичний	
Логічні	Структурний	Функціональний	Абстрактний	Раціональні
	Архітектурний	Технологічний	Теоретичний	
	Системний	Сутнісний	Емпіричний	
	Технічний	Ергономічний	Аналітичний	
	Інформаційний	Математичний	Ірраціональні	
	Операційний	Ресурсний	Світоглядний	Соціальний
	Інженерний	Часовий	Ментальний	Освітній
	Комунікаційний	Бізнес	Національний	Фізіологічний
	Координаційний	Стратегічний	Етнічний	Емоційний
Розгорнута класифікація та декомпозиція вимірів сприйняття комплексної підтримки програмних продуктів			Культурний	Чуттєвий
			Субкультурний	Інтуїтивний
			Традиційний	Суб'єктивний

Рисунок 2 – Розгорнута класифікація та декомпозиція вимірів сприйняття комплексної підтримки програмних продуктів

В свою чергу, кожен з введених вимірів визначається своїм набором факторів впливу, що відповідним чином впливають на суб'єктивізацію сприйняття об'єкта підтримки суб'єктом взаємодії, зміщуючи це сприйняття в напрямку того чи іншого виміру залежності і від персонального мультифакторного портрета кожного окремо взятого конкретного суб'єкта, і від узагальненого полісуб'єктного мультифакторного портрета середовища підтримки, сформованого всіма цими суб'єктами.

Результати дослідження / Results

Розроблена модель декомпозиції комплексної підтримки програмних продуктів на основі вимірів сприйняття, структура та стратегія функціонування

якої представлена у попередньому розділі цієї статті, отримала апробацію при розв'язанні релевантних прикладних практичних задач, однією з яких є, зокрема представлена в рамках цього дослідження, прикладна практична задача ідентифікації домінуючого виміру сприйняття досліджуваного середовища комплексної підтримки програмного продукту.

Означимо основні категоріальні поняття, а також вхідні дані розв'язуваної прикладної практичної задачі. Зокрема, в якості суб'єктів взаємодії (досліджуваного середовища комплексної підтримки програмного продукту) розглядаються фахівці команд технічної підтримки відділу клієнтської підтримки досліджуваного програмного

продукту. В якості прикладу факторів впливу розглянуто такі фактори (із зазначенням їх належності до відповідних вимірів сприйняття):

- Ф1 – поточна позиція на ринку ІТ (Вимір3 – інженерний логічний; Вимір7 – технічний логічний; Вимір10 – комунікаційний логічний);
- Ф2 – рівень персональної конфліктності (Вимір5 – емоційний ірраціональний);
- Ф3 – рівень взаємодії всередині команди (Вимір2 – структурний логічний; Вимір8 – координаційний логічний);
- Ф4 – рівень персонального добово-часового впливу (нічні зміни, чергування 24/7, тощо) (Вимір6 – фізіологічний ірраціональний);
- Ф5 – рівень персональної мотивації (Вимір4 – стратегічний логічний);
- Ф6 – рівень персонального емоційного самопочуття (Вимір5 – емоційний ірраціональний; Вимір8 – координаційний логічний);
- Ф7 – рівень взаємодії з іншими командами (Вимір2 – структурний логічний);
- Ф8 – рівень сертифікації запитів баз даних (Вимір7 – технічний логічний);
- Ф9 – рівень сертифікації системної інженерії (Вимір3 – інженерний логічний);
- Ф10 – рівень фінансової задоволеності (Вимір1 – бізнес логічний; Вимір9 – ресурсний логічний);
- Ф11 – рівень персонального фізичного самопочуття (Вимір6 – фізіологічний ірраціональний);
- Ф12 – рівень сертифікації володіння мовою (клієнта) (Вимір4 – стратегічний логічний; Вимір10 – комунікаційний логічний).

Одразу ж, важливо зауважити, що вибір саме такої конфігурації факторів впливу та вимірів сприйняття зумовлений, насамперед, вимогами безпосередньо самих дослідників, та/або вимогами самих задач, проте, в жодному разі не обмежується цим вибором, тобто може бути переглянута, доповнена, модифікована тощо, залежно від потреб (дослідників та/або задач). Водночас, основний акцент дослідження полягає не у наборі та дефініції номенклатури факторів впливу, а натомість – саме у забезпеченні зв'язок типу «фактор-вимір» з метою забезпечення можливості переходу між релевантними рівнями декомпозиції, в тому числі, зокрема, підняття з рівня факторів впливу – на більш глобальний рівень вимірів сприйняття. Разом з тим, єдиним фундаментальним та обов'язковим критерієм є попереднє узгодження та декларування дослідниками відповідності кожного окремого

фактора впливу – релевантного(-им) виміру(-ам) сприйняття. При цьому, модель не диктує жодних жорстких аксіоматичних вимог щодо певних константних зв'язок типу «фактор-вимір», натомість – однією з переваг розробленої моделі декомпозиції є забезпечення можливостей для дослідників самостійно як декларувати критерії відбору необхідних факторів впливу та вимірів сприйняття, так і формувати релевантні зв'язки між ними, залежно від поставлених задач, наявних умов, ситуацій чи обставин, а також на основі свого власного експертного та/або емпіричного досвіду.

Основним джерелом вхідних даних (тобто, в даному випадку – персональних мультифакторних портретів суб'єктів підтримки) – є відповідний спеціалізований метод, поданий у вищезгаданій праці [20], де він максимально детально описаний. В свою чергу, важливо зауважити, що розроблена модель декомпозиції, представлена в поточному дослідженні, не відповідає за формування цих мультифакторних портретів суб'єктів досліджуваних середовищ, а лише використовує їх в якості одних із можливих вхідних даних.

Розв'язання задекларованої експериментальної задачі полягає у поетапному виконанні певних кроків, а саме:

- формування персональних мультифакторних портретів суб'єктів підтримки досліджуваного об'єкта (підтримуваного програмного продукту);
- формування узагальненого портрета всього наявного середовища підтримки (на основі отриманих на попередньому кроці персональних портретів суб'єктів, з яких сформоване це середовище);
- розрахунок частки впливу кожного з вимірів сприйняття (в межах досліджуваного полісуб'єктного мультифакторного середовища підтримки програмного продукту) на основі прямих зв'язків наявних факторів впливу із відповідними вимірами сприйняття;
- ідентифікація домінуючого виміру сприйняття досліджуваного середовища комплексної підтримки програмного продукту.

Нижче, на рисунку 3, представлено відповідну графічну візуалізацію мультифакторних портретів суб'єктів досліджуваного середовища підтримки (отриманих шляхом застосування відповідного методу формування мультифакторних портретів суб'єктів підтримки програмних комплексів), що позначає частку присутності кожного із факторів впливу по кожному із суб'єктів досліджуваного середовища підтримки.

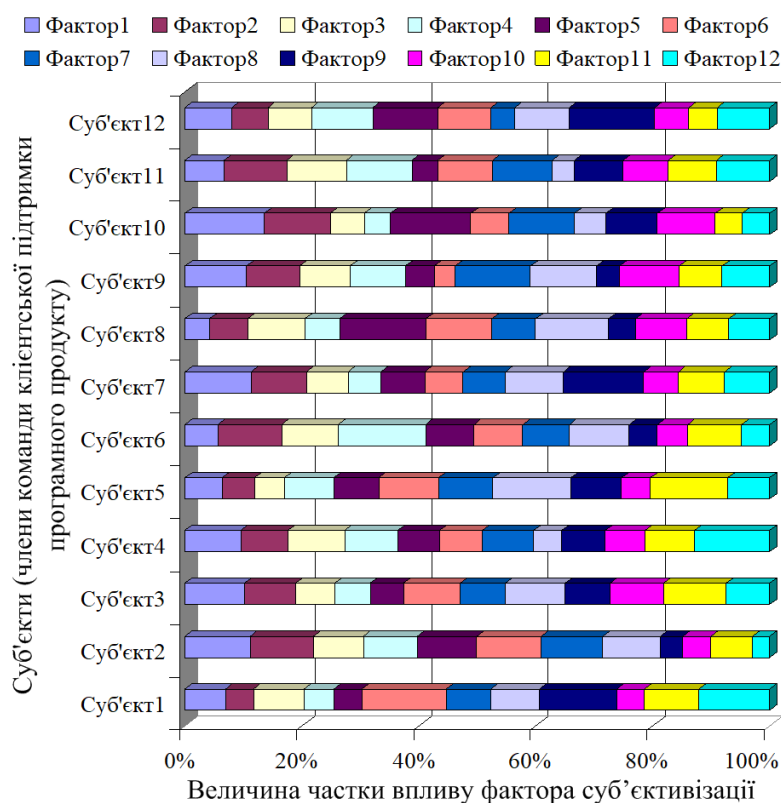


Рисунок 3 – Графічна візуалізація мультифакторних портретів суб'єктів досліджуваного середовища підтримки

Наступним етапом є формування узагальненого портрета всього наявного середовища підтримки (на основі отриманих на попередньому кроці персональних портретів суб'єктів, з яких сформоване це середовище).

Отож на рисунку 4, подано відповідну графічну візуалізацію досліджуваного полісуб'єктного мультифакторного середовища підтримки, отриманого на основі персональних мультифакторних портретів його суб'єктів.

Після цього здійснюється розрахунок частки впливу кожного з вимірів сприйняття (в межах досліджуваного полісуб'єктного мультифакторного середовища підтримки програмного продукту) на основі прямих зв'язків наявних факторів впливу із відповідними вимірами сприйняття.

Зокрема на рисунку 5 подано інтерпретацію зв'язку між наявними факторами впливу та відповідними вимірами сприйняття, у форматі коефіцієнтів належності кожного із факторів впливу – до відповідних вимірів сприйняття.

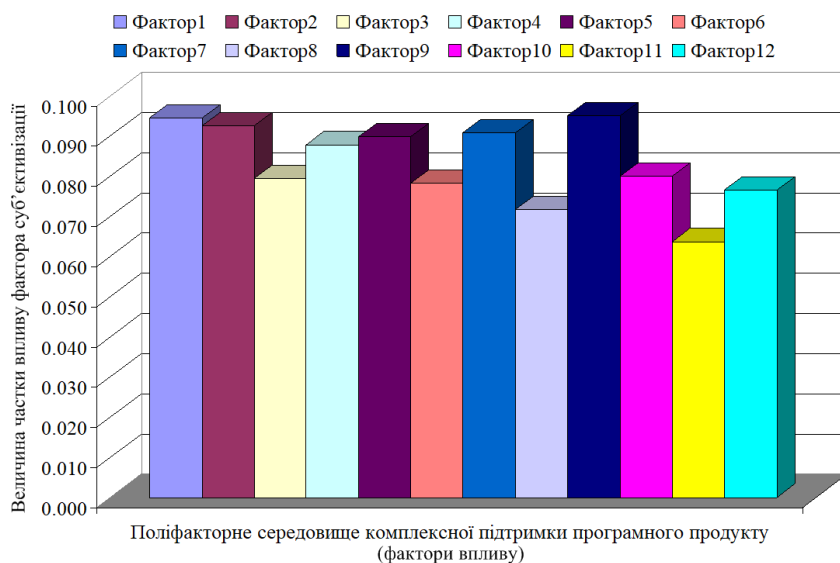


Рисунок 4 – Графічна візуалізація досліджуваного полісуб'єктного мультифакторного середовища комплексної підтримки програмного продукту

* Вимір1 - бізнес логічний; Вимір2 - структурний логічний; Вимір3 - інженерний логічний; Вимір4 - стратегічний логічний; Вимір5 - емоційний ірраціональний; Вимір6 - фізіологічний ірраціональний; Вимір7 - технічний логічний; Вимір8 - координаційний логічний; Вимір9 - ресурсний логічний; Вимір10 - комунікаційний логічний.

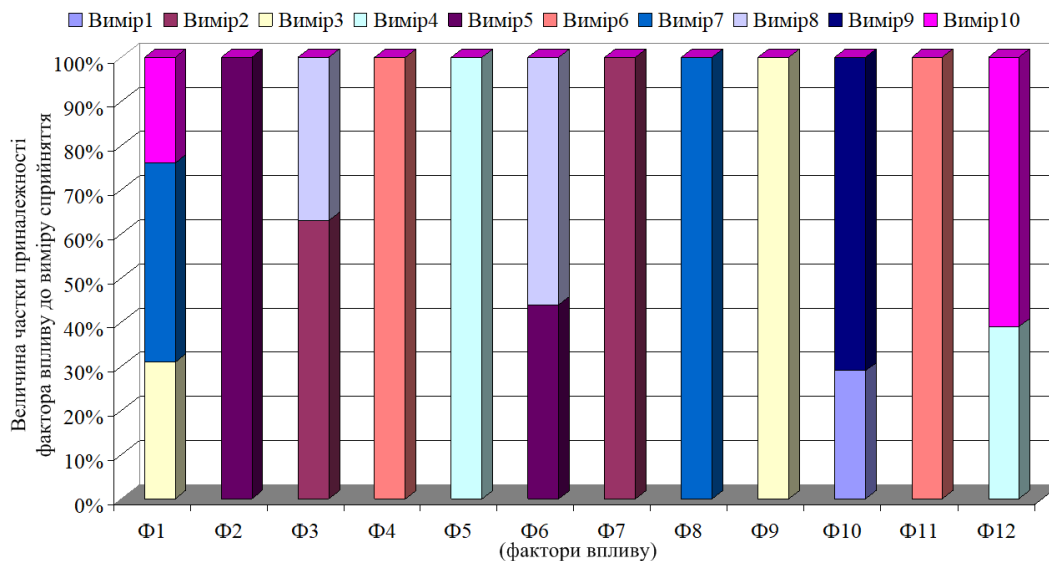


Рисунок 5 – Коефіцієнти належності кожного із факторів впливу до відповідних вимірів сприйняття

При цьому важливо зауважити, що не завжди конкретний фактор впливу можна однозначно віднести до лиш одного єдиного виміру сприйняття, тож іноді (а на практиці – доволі часто) один і той самий фактор впливу може бути пов'язаний із кількома вимірами сприйняття (з обов'язковим зазначенням величини частки його належності до кожного з вимірів).

Таким чином, зв'язок між релевантними факторами впливу та відповідними вимірами сприйняття позначається класичним полігамним реляційним зв'язком типу «багато до багатьох».

Завершальним етапом є ідентифікація домінуючого виміру сприйняття досліджуваного середовища комплексної підтримки програмного продукту (на основі попередньо отриманого узагальненого мультифакторного портрета досліджуваного середовища, а також наявних коефіцієнтів належності факторів впливу до відповідних вимірів сприйняття).

Розрахунок домінуючого виміру сприйняття досліджуваного середовища комплексної підтримки програмного продукту здійснюється на основі підсумовування добутоків (по кожному із факторів впливу та вимірів сприйняття): «частки належності фактора впливу» (див. рис. 5) на «частку впливу фактора суб'єктивізації (в рамках поліфакторного середовища)» (див. рис. 4). В результаті такого розрахунку, отримано відповідні результати ідентифікації домінуючого виміру сприйняття досліджуваного середовища комплексної підтримки програмного продукту, представлені на рисунку 6.

Отже, в результаті розв'язання поставленої прикладної практичної задачі ідентифікації домінуючого виміру сприйняття досліджуваного середовища комплексної підтримки програмного продукту встановлено, що домінуючим виміром сприйняття є «Вимір6», частка присутності якого в інтерпретації досліджуваного полісуб'єктного мультифакторного середовища комплексної підтримки програмного продукту виявилась максимальною у порівнянні з частками усіх інших представлених та задекларованих вимірів сприйняття.

Оскільки «Вимір6» відповідає за фізіологічний ірраціональний вимір сприйняття, де, згідно із запропонованою (в рамках вирішення поставленої експериментальної прикладної практичної задачі) класифікацією, факторами впливу є Ф4 (рівень персонального добовочасового впливу: нічні зміни, чергування 24/7 тощо) та Ф11 (рівень персонального фізичного самопочуття), то отримаємо додаткову інтерпретацію розуміння того аспекту, що це середовище підтримки (представлене відповідною командою технічної підтримки відділу клієнтської підтримки досліджуваного програмного продукту) потребує додаткової уваги саме в контексті персонального фізичного самопочуття членів команди, зумовленого, в тому числі, наявністю нічних змін, чергувань 24/7 чи інших проявів фактора добовочасового впливу, з метою приведення релевантного виміру сприйняття із поточного домінуючого стану – до більш оптимальних показників.

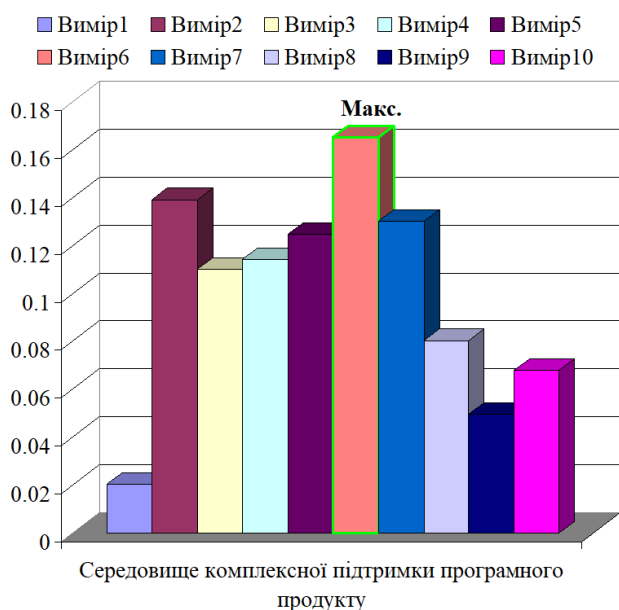


Рисунок 6 – Результати ідентифікації домінуючого виміру сприйняття досліджуваного середовища комплексної підтримки програмного продукту

Таким чином, за допомогою розробленої моделі декомпозиції комплексної підтримки програмних продуктів на основі вимірів сприйняття отримано розв'язок прикладної практичної задачі ідентифікації домінуючого виміру сприйняття досліджуваного середовища комплексної підтримки програмного продукту.

Обговорення результатів досліджень / Discussion

Робота [21] пропонує новітній спеціалізований композитний метод декомпозиції для широкомасштабної глобальної оптимізації (англ. Large-Scale Global Optimization), основною особливістю якого є інтеграція з існуючими методами композиційного групування роздільності (англ. Composite Separability Grouping) та класичних методів групування роздільності (англ. Separability Grouping), а також з методом виявлення мультиплікативно роздільної змінної (англ. Multiplicatively Separable Variable Detection) і методом групування нероздільних змінних (англ. Non-Separable Variable Grouping), використавши переваги кожного з них. Основний акцент запропонованого методу зроблено на оптимізацію використовуваних при декомпозиції обчислювальних ресурсів, водночас із збереженням точності декомпозиції та оптимізації результатів. Тобто, запропонований метод призначений для опрацювання саме високовимірних проблем (зокрема, мова йде про 1000-вимірні проблеми). Таким чином, запропонований метод безперечно є надзвичайно ефективним при вирішенні саме цього кластера задач. Водночас, в контексті визначеної в цьому дослідженні проблематики декомпозиції комплексної підтримки програмних

продуктів на основі вимірів сприйняття – доцільність застосування саме цього методу є неоправданою, оскільки вимірність поставленої проблеми є на кілька порядків меншою.

Інша праця [22] презентує задачо-декомпозиційний підхід для генеративного концептуального дизайну з використанням штучного інтелекту, що базується, в тому числі, на моделі декомпозиції «Функція-Поведінка-Структура» (англ. Function-Behavior-Structure), основною особливістю якої є те, що вона, фактично, розбиває задачу на ієрархічну систему функцій, поведінки та структур, таким чином її ієрархічна структура та акцент на декомпозиції завдань забезпечують гнучку та адаптовану структуру для керування процесом проектування та підтримки ефективного прийняття рішень. Ця модель безперечно є надзвичайно ефективною та результативною, що доведено десятиліттями її практичного застосування. Проте, в контексті досліджуваної проблеми декомпозиції комплексної підтримки програмних продуктів на основі вимірів сприйняття виникають додаткові виклики, пов'язані з необхідністю введення додаткової онтологічної категорії «сприйняття», що, фактично, повністю перевертає стандартне уявлення про класичну трикатегорійну онтологію, та, зрештою, призводить до виникнення додаткових нетривіальних науково-прикладних проблем.

Разом з тим розроблена та представлена в цій статті модель декомпозиції комплексної підтримки програмних продуктів на основі вимірів сприйняття в повній мірі відповідає потребам поставленої задачі, при цьому максимально гармонійно інтегруючись в існуючі напрацювання проблематики суб'єктивізації сприйняття об'єкта підтримки суб'єктами взаємодії, спричинених факторами впливу, які, крім того формують виміри сприйняття.

Апробація розробленої моделі на прикладі розв'язання практичної задачі ідентифікації домінуючого виміру сприйняття досліджуваного середовища комплексної підтримки програмного продукту додатково підтверджує життєздатність запропонованого рішення, а також перспективність його подальшого застосування для вирішення цього кластера науково-прикладних та практичних задач, як в області автоматизації й інтелектуалізації комплексної підтримки програмних продуктів, так і в інших областях міжсуб'єктної взаємодії та суб'єктивізації сприйняття.

Висновки / Conclusions

В статті розроблена модель декомпозиції комплексної підтримки програмних продуктів на основі вимірів сприйняття, що забезпечує можливість однойменної декомпозиції в контексті дослідження проблеми автоматизації й

інтелектуалізації комплексної підтримки програмних продуктів. В якості основних методів дослідження використано базові методи класифікації та методи декомпозиції, проте в якості додаткової обов'язкової компоненти вирішення поставленої проблеми використано метод формування мультифакторних портретів суб'єктів підтримки програмних комплексів.

Таким чином, розроблена модель дає змогу отримати симбіоз абсолютно всіх фундаментальних складових досліджуваної проблеми, а саме: об'єкта (безпосередньо самого підтримуваного програмного продукту); процесів (комплексної підтримки програмного продукту); суб'єктів (що безпосередньо забезпечують та реалізують комплексну підтримку програмного продукту); факторів впливу (що існують в рамках відповідних вимірів сприйняття, та безпосередньо впливають на суб'єктивізацію сприйняття об'єкта суб'єктами взаємодії з ним); а також вимірів сприйняття (що поєднують в собі відповідні релевантні фактори впливу, та відповідають за стратегію впливу цих факторів на суб'єктивізацію сприйняття суб'єктами об'єкта, з яким вони взаємодіють).

В контексті розробленої моделі декомпозиції, сформульовано та введено 38 вимірів сприйняття, що забезпечують можливість деталізованої декомпозиції процесів комплексної підтримки та комплексної класифікації релевантних факторів впливу, що призводять до суб'єктивізації сприйняття об'єкта комплексної підтримки тими суб'єктами, які безпосередньо забезпечують та реалізують цю підтримку.

В якості прикладу апробації розробленої моделі наведено деталізовані результати розв'язання прикладної практичної задачі ідентифікації домінуючого виміру сприйняття досліджуваного середовища комплексної підтримки програмного продукту, а також задекларовані основні перспективи подальшого застосування розробленої моделі декомпозиції на основі вимірів сприйняття в контексті науково-прикладної проблематики автоматизації й інтелектуалізації комплексної підтримки програмних продуктів.

Вдячність / Acknowledgments

Робота є ініціативною. Дослідження проводились в рамках наукової діяльності авторів поза робочим часом за основними посадами.

Список літератури:

1. Raju Cherukuri, B. (2022). Developing intelligent chatbots for real-time customer support in e-commerce. *International Journal of Science and Research (IJSR)*, 11(1), 1709–1719. doi.org/10.21275/sr220113103101
2. Promsiri, T., Wuittipappinyo, N. & Keerativutisest, V. (2025). Enhancing chatbot service

user satisfaction. *TEM Journal*, 14(1), 216–225. <https://doi.org/10.18421/tem141-20>

3. Madhumita, R., & Chandana, D. (2025). Advancing Software Testing: Integrating AI, Machine Learning, and Emerging Technologies. *International Journal of Research in Engineering, Science and Management*, 8(1), 93-97. journal.ijresm.com/index.php/ijresm/article/view/3198

4. Pareek, C.S. (2025). AI-Driven Software Testing: A Deep Learning Perspective. *International Journal of Innovative Research in Engineering & Multidisciplinary Physical Sciences*, 13(1), 1-10. <https://www.ijrmps.org/research-paper.php?id=231946>

5. Priyanka, M. (2025). Analyzing the impact of agile methodologies on software quality and delivery speed: A comparative study. *World Journal of Advanced Research and Reviews*, 25(1), 1207–1216. <https://doi.org/10.30574/wjarr.2025.25.1.0184>

6. Dugbartey, A.N. & Kehinde, O. (2025). Optimizing Project Delivery Through Agile Methodologies: Balancing speed, collaboration and stakeholder engagement. *World Journal of Advanced Research and Reviews*, 25(1), 1237–1257. <https://doi.org/10.30574/wjarr.2025.25.1.0193>

7. Kolawole, I. & Fakokunde, A. (2025). Machine learning algorithms in DevOps: Optimizing Software Development and deployment workflows with precision. *International Journal of Research Publication and Reviews*, 6(1), 247–264. <https://doi.org/10.55248/gengpi.6.0125.0209>

8. Kanakala, R.R. (2025). The evolution of DEVOPS and SRE: A technical perspective. *International Journal of Research in Computer Applications and Information Technology*, 8(1), 3511–3520. https://doi.org/10.34218/ijrcait_08_01_252

9. Ouali-Alami, C. et al. (2021). Approaches to design complex software systems. *Proceedings of the 2nd International Conference on Big Data, Modelling and Machine Learning*, 533–537. <https://doi.org/10.5220/0010740200003101>

10. Zhu, L., Tamburri, D.A. & Casale, G. (2023). RADF: Architecture decomposition for function as a service. *Software: Practice and Experience*, 54(4), 566–594. <https://doi.org/10.1002/spe.3276>

11. Gaba, A., & Srinivasan, G.N. (2020). Decomposition and Modularity in Software Systems. *International Research Journal of Engineering and Technology (IRJET)*, 7(6), 1349-1352. URL: <https://www.irjet.net/archives/V7/i6/IRJET-V7I6250.pdf>

12. Abgaz, Y. M., McCarren, A., Eklund, P., Solan, D., Lapuz, N., Bivol, M., Jackson, G., Yilmaz, M., Buckley, J., & Clarke, P. M. (2023). Decomposition of Monolith Applications Into

Microservices Architectures: A Systematic Review. *IEEE Transactions on Software Engineering*, 1–32. <https://doi.org/10.1109/tse.2023.3287297>

13. Diaferia, L., Blohm, I., De Rossi, L.M., & Salviotti, G. (2022). When Standard Is Not Enough: a Conceptualization of AI Systems' Customization and its Antecedents. In *Proceedings of Forty-Third International Conference on Information Systems*, 8, 1-18. URL: https://aisel.aisnet.org/icis2022/is_implement/is_implement/8

14. Ali, A. Q., Md Sultan, A. B., Abd Ghani, A. A., & Zulzalil, H. (2020). Development of a valid and reliable software customization model for SaaS quality through iterative method: perspectives from academia. *PeerJ Computer Science*, 6, e294, 1-34. <https://doi.org/10.7717/peerj-cs.294>

15. Qasem Ali, A., Md Sultan, A., Abd Ghani, A., & Zulzalil, H. (2021). An Empirical Investigation of Software Customization and Its Impact on the Quality of Software as a Service: Perspectives from Software Professionals. *Applied Sciences*, 11(4), 1677. <https://doi.org/10.3390/app11041677>

16. Abendroth, A., Bender, B., & Gronau, N. (2024). The Evolution of Original ERP Customization: A Systematic Literature Review of Technical Possibilities. In *Proceedings of the 26th International Conference on Enterprise Information Systems (ICEIS 2024)*, 1, 17-27. doi.org/10.5220/0012305500003690

17. Melody, Y.K. (2003). A comparative assessment of classification methods. *Decision Support Systems*, 35(4), 441-454. [https://doi.org/10.1016/S0167-9236\(02\)00110-0](https://doi.org/10.1016/S0167-9236(02)00110-0)

18. Aggarwal, C.C. (Ed.). (2014). *Data Classification: Algorithms and Applications* (1st ed.). Chapman and Hall/CRC, New York, 707 pages. <https://doi.org/10.1201/b17320>

19. Conejo, A. (2006). *Decomposition techniques in mathematical programming: Engineering and science applications*. Springer, 558 pages. <https://doi.org/10.5860/choice.44-2155>

20. Pukach, A. I. ., & Teslyuk, V. M. (2025). Method of forming multifactor portraits of the subjects supporting software complexes, using a multilayer perceptron. *Radio Electronics, Computer Science, Control*, (1), 130–141. <https://doi.org/10.15588/1607-3274-2025-1-12>

21. Tian, M. et al. (2024). A composite decomposition method for large-scale global optimization. *IEEE Transactions on Artificial Intelligence*, 5(12), 6304–6319. <https://doi.org/10.1109/tai.2024.3373391>

22. Wang, B. et al. (2023). A task-decomposed AI-aided approach for Generative Conceptual Design. *35th International Conference on Design Theory and Methodology (DTM)*, 6, 1-11. <https://doi.org/10.1115/detc2023-109087>

References:

1. Raju Cherukuri, B. (2022). Developing intelligent chatbots for real-time customer support in e-commerce. *International Journal of Science and Research (IJSR)*, 11(1), 1709–1719. <https://doi.org/10.21275/sr220113103101>

2. Promsiri, T., Wuttiappinyo, N. & Keerativutisest, V. (2025). Enhancing chatbot service user satisfaction. *TEM Journal*, 14(1), 216–225. <https://doi.org/10.18421/tem141-20>

3. Madhumita, R., & Chandana, D. (2025). Advancing Software Testing: Integrating AI, Machine Learning, and Emerging Technologies. *International Journal of Research in Engineering, Science and Management*, 8(1), 93-97. journal.ijresm.com/index.php/ijresm/article/view/3198

4. Pareek, C.S. (2025). AI-Driven Software Testing: A Deep Learning Perspective. *International Journal of Innovative Research in Engineering & Multidisciplinary Physical Sciences*, 13(1), 1-10. <https://www.ijirms.org/research-paper.php?id=231946>

5. Priyanka, M. (2025). Analyzing the impact of agile methodologies on software quality and delivery speed: A comparative study. *World Journal of Advanced Research and Reviews*, 25(1), 1207–1216. <https://doi.org/10.30574/wjarr.2025.25.1.0184>

6. Dugbartey, A.N. & Kehinde, O. (2025). Optimizing Project Delivery Through Agile Methodologies: Balancing speed, collaboration and stakeholder engagement. *World Journal of Advanced Research and Reviews*, 25(1), 1237–1257. <https://doi.org/10.30574/wjarr.2025.25.1.0193>

7. Kolawole, I. & Fakokunde, A. (2025). Machine learning algorithms in DevOps: Optimizing Software Development and deployment workflows with precision. *International Journal of Research Publication and Reviews*, 6(1), 247–264. <https://doi.org/10.55248/gengpi.6.0125.0209>

8. Kanakala, R.R. (2025). The evolution of DEVOPS and SRE: A technical perspective. *International Journal of Research in Computer Applications and Information Technology*, 8(1), 3511–3520. https://doi.org/10.34218/ijrcait_08_01_252

9. Ouali-Alami, C. et al. (2021). Approaches to design complex software systems. *Proceedings of the 2nd International Conference on Big Data, Modelling and Machine Learning*, 533–537. <https://doi.org/10.5220/0010740200003101>

10. Zhu, L., Tamburri, D.A. & Casale, G. (2023). RADF: Architecture decomposition for function as a service. *Software: Practice and Experience*, 54(4), 566–594. <https://doi.org/10.1002/spe.3276>

11. Gaba, A., & Srinivasan, G.N. (2020). *Decomposition and Modularity in Software Systems*.

International Research Journal of Engineering and Technology (IRJET), 7(6), 1349-1352. URL: <https://www.irjet.net/archives/V7/i6/IRJET-V7I6250.pdf>

12. Abgaz, Y. M., McCarren, A., Eklund, P., Solan, D., Lapuz, N., Bivol, M., Jackson, G., Yilmaz, M., Buckley, J., & Clarke, P. M. (2023). Decomposition of Monolith Applications Into Microservices Architectures: A Systematic Review. *IEEE Transactions on Software Engineering*, 1–32. <https://doi.org/10.1109/tse.2023.3287297>

13. Diaferia, L., Blohm, I., De Rossi, L.M., & Salviotti, G. (2022). When Standard Is Not Enough: a Conceptualization of AI Systems' Customization and its Antecedents. In *Proceedings of Forty-Third International Conference on Information Systems*, 8, 1-18. URL: https://aisel.aisnet.org/icis2022/is_implement/is_implement/8

14. Ali, A. Q., Md Sultan, A. B., Abd Ghani, A. A., & Zulzalil, H. (2020). Development of a valid and reliable software customization model for SaaS quality through iterative method: perspectives from academia. *PeerJ Computer Science*, 6, e294, 1-34. <https://doi.org/10.7717/peerj-cs.294>

15. Qasem Ali, A., Md Sultan, A., Abd Ghani, A., & Zulzalil, H. (2021). An Empirical Investigation of Software Customization and Its Impact on the Quality of Software as a Service: Perspectives from Software Professionals. *Applied Sciences*, 11(4), 1677. <https://doi.org/10.3390/app11041677>

16. Abendroth, A., Bender, B., & Gronau, N. (2024). The Evolution of Original ERP

Customization: A Systematic Literature Review of Technical Possibilities. In *Proceedings of the 26th International Conference on Enterprise Information Systems (ICEIS 2024)*, 1, 17-27. <https://doi.org/10.5220/0012305500003690>

17. Melody, Y.K. (2003). A comparative assessment of classification methods. *Decision Support Systems*, 35(4), 441-454, [https://doi.org/10.1016/S0167-9236\(02\)00110-0](https://doi.org/10.1016/S0167-9236(02)00110-0)

18. Aggarwal, C.C. (Ed.). (2014). *Data Classification: Algorithms and Applications* (1st ed.). Chapman and Hall/CRC, New York, 707 pages. <https://doi.org/10.1201/b17320>

19. Conejo, A. (2006). *Decomposition techniques in mathematical programming: Engineering and science applications*. Springer, 558 pages. <https://doi.org/10.5860/choice.44-2155>

20. Pukach, A. I. ., & Teslyuk, V. M. (2025). Method of forming multifactor portraits of the subjects supporting software complexes, using a multilayer perceptron. *Radio Electronics, Computer Science, Control*, (1), 130–141. doi.org/10.15588/1607-3274-2025-1-12

21. Tian, M. et al. (2024). A composite decomposition method for large-scale global optimization. *IEEE Transactions on Artificial Intelligence*, 5(12), 6304–6319. <https://doi.org/10.1109/tai.2024.3373391>

22. Wang, B. et al. (2023). A task-decomposed AI-aided approach for Generative Conceptual Design. *35th International Conference on Design Theory and Methodology (DTM)*, 6, 1-11. <https://doi.org/10.1115/detc2023-109087>.

© А. І. Пукач, В. М. Теслюк, 2025.

Науково-методична стаття.

Надійшла до редакції 08.04.2025.

Прийнято до публікації 04.06.2025.