

ВДОСКОНАЛЕННЯ МЕТОДУ ВИРІШЕННЯ ЗАДАЧІ РАННЬОЇ ІДЕНТИФІКАЦІЇ КОНФІГУРАЦІЙНИХ ЕЛЕМЕНТІВ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Анотація. Формальне вирішення задачі ранньої ідентифікації конфігураційних елементів (Configuration Items, CI) інформаційної системи (ІС) дозволить значно зменшити витрати часу та зусиль на створення описів архітектури, планування та управління ІТ-проектом створення ІС. Науково-прикладною проблемою дослідження є визначення можливості вирішення задачі ранньої ідентифікації CI ІС з використанням методів та алгоритмів кластеризації, які не сильно залежать від припущень про форму кластерів. Метою дослідження є розробка нового або вдосконалення існуючого методу вирішення задачі ранньої ідентифікації CI ІС. Запропоновано вдосконалення методу вирішення цієї задачі, яке полягає у використанні алгоритму кластеризації DBSCAN для виявлення архітектурних елементів як кластерів окремих функцій ІС. Використання вдосконаленого методу дозволяє автоматизувати процедуру синтезу опису архітектури ІС. Застосування цієї автоматизованої процедури сприяє підвищенню якості розробки ІС за рахунок виділення для проектування множини архітектурних сутностей цієї системи, яка є значно меншою, ніж множина елементарних функцій ІС.

Ключові слова: інформаційна система, конфігураційний елемент, задача ранньої ідентифікації, функція, архітектурна сутність, DBSCAN, опис архітектури.

Постановка проблеми. Процес управління конфігурацією є одним з найважливіших процесів життєвого циклу інформаційних систем (ІС) управління підприємствами та організаціями. Призначенням цього процесу є систематичний контроль за змінами конфігурації, підтримка цілісності та відстеження конфігурації протягом усього життєвого циклу продукту [1]. Незалежно від підходів до структуризації та опису окремих робіт і задач цього процесу, у ньому виділяється задача ідентифікації конфігурації, пов'язана із визначенням системних елементів конфігурації («Configuration item», CI) [1, 2]. Рішення цієї задачі поступово встановлюють та підтримують поточну основу для контролю та обліку стану ІС та її CI протягом усіх їхніх життєвих циклів [2].

Теоретичні та прикладні основи систем управління конфігурацією ІТ-продуктів визнано в цілому сформованими [2, 3] і запропоновано представляти як сукупність таких груп [3]:

- теоретичні основи (зокрема, трикомпонентна модель);
- кращі практики (зокрема, ISO 20000 та ITIL, а також їх аналоги);

– інновації (зокрема, інтеграція DevOps, інфраструктура як код, технології контейнеризації).

Теоретичні основи формують фундаментальне розуміння управління конфігурацією. Найкращі практики забезпечують досить ефективне вирішення задачі ідентифікації, контролю та обліку стану СІ. Розглянуті інновації удосконалюють традиційні методи управління конфігурацією ІТ-продукту, забезпечуючи масштабованість, автоматизацію та адаптованість прикладних рішень [3].

Головним недоліком такого представлення є його концептуальність – незважаючи на єдність поглядів на визначення задачі ідентифікації конфігурації, її розв’язання й досі розглядаються на концептуальному рівні [2]. Теоретичні основи і кращі практики управління конфігурацією вимагають значних зусиль з адаптації до особливостей конкретних ІС. Така адаптація виконується, зазвичай, із використанням інноваційних підходів і методів і залишає невирішеною цілу низку проблем. До цих проблем належать: проблеми безпеки; питання дотримання вимог до ІТ-продукту; складності, що виникають під час реалізації різних варіантів співробітництва, взаємодії ІТ-продукту з навколишнім світом [3].

Головною причиною відсутності формальних засобів розв’язання слід вважати усталене представлення СІ як сукупності апаратних засобів, програмного забезпечення або того й іншого разом. Таке розмите представлення викликає суперечливі вимоги до задачі ідентифікації конфігурації. Наслідком цього є виникнення в ході вирішення задачі ідентифікації конфігурації складної проблеми виділення оптимальної кількості СІ. Під оптимальною тут слід розуміти таку кількість СІ, яка вимагатиме мінімальних витрат часу та ресурсів на виконання подальших робіт процесу управління конфігурацією ІТ-продукту [4]. Ця проблема додатково ускладнена необхідністю простеження (трасування) вимог, що висунуто до ІС, до описів окремих СІ, з яких складається опис архітектури створюваної ІС. Під час подолання цього ускладнення виникає необхідність об’єднання окремих функцій як СІ створюваної ІС в окремі архітектурні сутності, з яких складається загальний опис архітектури створюваної ІС. Використання цих архітектурних сутностей, чия кількість значно менша кількості окремих функцій ІС, суттєво спрощує перехід від процесів формування та аналізу вимог до ІС до процесів проектування цієї ж ІС. Вирішення цієї задачі дозволить значно зменшити витрати часу та зусиль на створення описів архітектури, планування та управління ІТ-проектом створення ІС. Тому проведення досліджень в галузі розробки нових і вдосконалення існуючих методів вирішення задачі ідентифікації СІ ІС залишається одним з актуальних напрямів розвитку комп’ютерних наук.

Аналіз сучасних досліджень і публікацій. Як показано у [3], існуючі основи систем управління конфігурацією ІТ-продуктів є надто узагальненими і не надають формальних описів вирішення задачі ідентифікації СІ ІС. До того ж ці положення та моделі є надто узагальненими і вимагають значних зусиль з адаптації до особливостей конкретних ІС.

Одним із підходів до подолання цього недоліку є пропозиція відмовитися від єдиного опису СІ протягом усього життєвого циклу ІС. Така відмова дає змогу розглядати задачу ідентифікації конфігурації ІС як сукупність таких задач [4]:

– задача ранньої ідентифікації СІ (результатом вирішення якої є множина СІ як архітектурних сутностей ІС, виділених за результатами аналізу добре сформульованих вимог до цієї системи);

– задача проєктної ідентифікації СІ (результатом вирішення якої є множина СІ як елементів дизайну ІС, виділених за результатами аналізу архітектурних сутностей цієї системи);

– задача ідентифікації СІ під час реалізації (development) (результатом вирішення якої є множина СІ як елементів програмного та/або апаратного забезпечення ІС, виділених за результатами аналізу СІ як елементів дизайну цієї системи).

Даний підхід спрощує роботи з формування описів СІ на різних стадіях життєвого циклу ІС. Але він залишає невирішеною проблему виділення оптимальної кількості СІ [4].

У [3] також було показано, що вирішення задачі ранньої ідентифікації СІ значною мірою залежить від вирішення проблеми дотримання вимог до ІТ-продукту. В загальному випадку цю залежність можна визначити таким твердженням: задача ранньої ідентифікації СІ як архітектурних сутностей ІС повинна бути вирішеною для описів добре сформульованих вимог до цієї ІС. При цьому під терміном «архітектурна сутність» слід розуміти групи функціональних вимог (функціональні підсистеми і задачі, функціональні модулі, контури управління ресурсами, ІТ-сервіси тощо), які співвідносять функції ІС із бізнес-потребами користувачів цієї ІС [4].

Загальне вирішення задачі ранньої ідентифікації СІ для ІС може здійснюватися на основі таких підходів [4]:

а) функціональна декомпозиція загального опису архітектури ІС на окремі архітектурні сутності та системні елементи, з яких складаються ці сутності;

б) синтез описів архітектурних сутностей та загального опису архітектури з описів окремих системних елементів або вимог до цих елементів.

Приклад використання підходу а) описано у [5]. При цьому поставлена задача ідентифікації конфігурації вирішується у два етапи:

– на першому етапі виконується функціональна декомпозиція опису архітектури на окремі елементи з урахуванням необхідних еволюційних змін;

– на другому етапі здійснюється вибір тих варіантів декомпозиції, які задовольняють обмеження на вартість необхідних змін.

Однак розглянутий у [5] підхід орієнтований переважно на відстеження змін, що вносяться до опису архітектури. При цьому залишаються нерозглянутими питання первинної ідентифікації окремих функцій як СІ у результаті аналізу точок зору на архітектуру цієї системи, а також формування та розвитку архітектурних сутностей ІС як множин СІ, що були виділені під час вирішення задачі.

На відміну від підходу а), сучасні дослідження методів та засобів, які базуються на підході б), з самого початку використовують методи та засоби штучного інтелекту.

Це викликано тим, що, як показано у [6], проблеми раннього поділу системи на окремі елементи виникали, переважно, в результаті неправильного тлумачення вимог або упрежденості особистого досвіду. Так, у [7] наведено приклади застосування для моделювання та налаштування функцій системи таких інструментів штучного інтелекту, як рекомендаційні системи та методи машинного навчання. Але хоча розглянуті приклади враховують чинник мінливості описів окремих функцій, питання зв'язності цих функцій та утворення з них архітектурних сутностей залишаються невирішеними.

У [8] для вирішення проблеми вибору конфігурації системи запропоновано використовувати семплер на основі глибокого навчання з підкріпленням. Результатом застосування цього семплеру є ідентифікована мінімальна підмножина конфігурацій, яка охоплює всі варіанти взаємодії функцій при мінімізації надмірності. Але таке рішення для ІС, призначених для управління стабільними процесами підприємств та організацій, є надмірно складним.

Часто для вирішення задач ідентифікації СІ в межах підходу б) дослідники застосовують методи кластеризації. Як приклади подібного застосування методів та алгоритмів кластеризації можна вказати дослідження з вирішення таких задач:

- виділення мікросервісів у ході рефакторингу вихідного коду монолітного програмного застосування [9];
- рефакторингу складних багаторівневих монолітних програмних систем у ході їх декомпозиції на мікросервіси [10];
- ранньої ідентифікації беклогів команд виконавців як СІ системи [11].

Але в розглянутих дослідженнях не враховуються зв'язки між окремими функціями ІС, які обумовлюють групування функцій у архітектурні сутності ІС. Взагалі питання про фактори, які визначають функцію мети, обмеження та спосіб розрахунку відстані між окремими функціями ІС, залишається й досі відкритим.

Використання методів кластеризації для вирішення задачі ранньої ідентифікації вимагає також формалізації вимог, які висуваються до ІС. У [12] було проведено систематичний огляд сучасних наукових досліджень інструментів, методів, прийомів та процесів вилучення вимог із середовищ, що базуються на методології Model-Based System Engineering, та генерації моделей на основі вимог, сформульованих природною мовою. За результатами цього огляду було встановлено, зокрема, що генерація моделей на основі описів, зроблених природною мовою, залишається домінуючим способом формального опису вимоги до ІС. При цьому використання великих мовних моделей під час такої генерації вимагає більше даних, ніж можуть надати сучасні методи формування подібних текстових описів, і є недоцільним [12]. Але питання використання моделей вимог як основи для створення описів СІ ІС у [12] залишилося нерозглянутим.

Проведений аналіз розглянутих публікацій [3–12] дозволив зробити такі висновки:

- вирішення задачі ранньої ідентифікації СІ ІС бажано здійснювати з використанням підходу б), тому що створити загальний опис архітектури ІС для подальшої декомпозиції в багатьох випадках неможливо;

– для вирішення зазначеної задачі краще за все використовувати формальні методи штучного інтелекту і, зокрема, методи кластеризації;

– як похідні дані для вирішення зазначеної задачі слід використовувати моделі вимог, які висуваються до ІС, зокрема – формальні моделі функціональних вимог до ІС на основі їхніх онтологічних описів.

Мета дослідження. Метою дослідження є розробка нового або вдосконалення існуючого методу вирішення задачі ранньої ідентифікації СІ ІС. Це дозволить автоматизувати процедуру синтезу опису архітектури інформаційної системи. Застосування цієї автоматизованої процедури сприяє підвищенню якості розробки інформаційної системи за рахунок виділення для проектування множини архітектурних сутностей цієї системи, яка є значно меншою, ніж множина елементарних функцій інформаційної системи.

Основна частина. Об'єктом даного дослідження є процес управління конфігурацією ІТ-проєкту. Предметом дослідження є методи вирішення задачі ранньої ідентифікації СІ ІС.

В процесі дослідження було вирішено розглядати рішення задачі ранньої ідентифікації як результат послідовного вирішення таких підзадач:

а) підзадача синтезу описів раціональної архітектури створюваної ІС на основі описів добре сформульованих функціональних вимог до цієї ІС (Підзадача 1);

б) підзадача синтезу описів архітектурних сутностей для описів раціональної архітектури створюваної ІС, які є результатами вирішення Підзадачі 1 (Підзадача 2).

Підзадачу 1 у [13] запропоновано описати теоретико-ігровою моделлю вигляду

$$\Gamma_{IS} = \langle \{Pr, U\}, \{X^{Pr, U}\}, \{f^{Pr, U}\} \rangle, \quad (1)$$

де Γ_{IS} – гра Постачальника та Споживача ІТ-послуг (функцій ІС); $\{Pr, U\}$, – множина гравців, які беруть участь у грі Γ_{IS} (Постачальник та Споживач ІТ-послуг відповідно); $\{X^{Pr, U}\}$ – множина стратегій гри Γ_{IS} , яка формується на основі множини варіантів описів архітектури створюваної ІС Π_{acm} ; $\{f^{Pr, U}\}$ – множина функцій виграшу гри Γ_{IS} [13].

Гра (1) є скінченною безкоаліційною грою з нульовою сумою Постачальника та Споживача ІТ-послуг. Детальний опис елементів гри (1) та формальних моделей описів вимог до окремих функцій ІС для даної гри наведено у [13]. Для знаходження результату гри (1) у [13] запропоновано використовувати метод пошуку рівноваг Неша для біматричної гри в чистих стратегіях. Цей метод передбачав послідовність таких етапів.

Етап 1. Сформувані матриці виграшів Споживача U й Постачальника ІТ-послуг Pr .

Етап 2. Відзначити максимальні елементи в кожному стовпці матриці U . Якщо в деякому стовпці даної матриці кілька максимальних елементів, то необхідно відзначити всі такі елементи.

Етап 3. Відзначити максимальні елементи в кожному рядку матриці Pr . Якщо в деякому рядку даної матриці кілька максимальних елементів, то необхідно відзначити всі такі елементи.

Етап 4. Провести пошук перетинання результатів Етапу 2 і Етапу 3 (тобто максимальних елементів, які перебувають на тих самих позиціях у кожній з матриць).

Етап 5. Якщо результат виконання Етапу 4 не є порожньою множиною, зафіксувати відповідні варіанти описів архітектури створюваної ІС як описи раціональної архітектури даної системи $\{K_{ji}^{f_{js}}\}_{i=1,\dots,e}^{Pr,U}$, де e – кількість добре сформульованих функціональних вимог до ІС. У іншому випадку визнати, що опису раціональної архітектури ІС на чистих стратегіях не існує. Завершити застосування методу.

Результатом використання цього методу є опис архітектури створюваної ІС як множини окремих функцій цієї ІС. Це означає, що як окремих СІ ІС обрано опис окремої функції (або вимоги до цієї функції). Але таке вирішення Підзадачі 1 визнано недоліком, тому що кількість окремих СІ у великих та середніх ІС управління підприємствами та організаціями може сягати кількох сотень. Це значно ускладнює подальший розподіл опису архітектури ІС між окремими командами виконавців ІТ-проєкту розробки цієї системи.

Для ліквідації даного недоліку під час вирішення Підзадачі 2 запропоновано зменшити кількість СІ в описі архітектури ІС. В загальному випадку це зменшення може бути описано як функція мети J , що має вигляд [4, 14]

$$J(M, d, U, C) = \sum_{i=1}^c \sum_{j=1}^k u_{ij} d(m_j, c^{(i)}) \rightarrow \min, \quad (2)$$

де M – базова множина похідних даних; m_j – j -та точка (вектор) похідних даних; k – кількість точок (векторів) похідних даних; d – спосіб визначення відстані між точками (векторами) похідних даних; U – матриця розбиття точок (векторів) похідних даних між кластерами $c^{(i)}$; u_{ij} – елемент матриці розбиття, який встановлює належність точки (вектору) m_j до кластеру $c^{(i)}$; C – множина кластерів, які є результатами вирішення задачі ранньої ідентифікації; $c^{(i)}$ – i -й кластер, який є одним з результатів вирішення задачі ранньої ідентифікації СІ ІС і описує виявлену архітектурну сутність ІС як підмножину окремих функцій; c – кількість сформованих кластерів.

Набір обмежень для функції мети (2) має вигляд [4, 14]:

$$u_{ij} \in \{0, 1\}; \sum_{i=1}^c u_{ij} = 1; \sum_{j=1}^k u_{ij} \leq k. \quad (3)$$

Ці обмеження встановлюють, що:

- кожна похідна точка (вектор) може належати тільки одному з кластерів та не належати іншим кластерам;
- в кожному кластері міститься не менше однієї похідної точки (вектору), але не менше загальної кількості похідних точок (векторів) k .

Як основний алгоритм вирішення Підзадачі 2 у [4] було запропоновано використовувати алгоритм щільнісної кластеризації DBSCAN.

Вибір групи методів щільнісної кластеризації обумовлений такими міркуваннями [14]:

- ці методи, на відміну від багатьох інших методів кластеризації (partitioning methods, ієрархічні методи тощо), можуть знаходити кластери довільної форми;
- ці методи визначають кластери як щільні області об'єктів досліджуваної вибірки, розділені областями з низькою щільністю (що якнайкраще відповідає визначенню архітектурної сутності);
- ці методи можуть відфільтрувати аномалії (тобто описи окремих функцій ІС, включення яких в загальний опис архітектури цієї системи недоцільне із заздалегідь визначених аналітиком причин).

У [14] визнано існування кращих за DBSCAN методів та алгоритмів кластеризації, які дозволяють виявляти кластери довільної форми високої якості (зокрема, метод агломеративної кластеризації Chameleon). Але алгоритм DBSCAN простіший у реалізації і вимагає менших обчислювальних витрат і витрат часу на вирішення задачі кластеризації [15]. Крім того, однією з переваг алгоритму DBSCAN є можливість обробляти дані різноманітної природи. Як показано в [15] з посиланнями на відповідні дослідження: «...DBSCAN ніколи не обмежувався використанням евклідової відстані або точок у \mathbb{R}^d , але завжди призначався для використання з географічними даними, полігонами та іншими типами даних».

У [4] було проведено дослідження з порівняльного аналізу результатів вирішення задачі ранньої ідентифікації СІ, отриманих з використанням щільнісного алгоритму DBSCAN, методів ієрархічної кластеризації DIANA, AGNES та Chameleon, а також алгоритму k-means. Для порівняння було запропоновано критерії «Громізdkість рішення» та «Виявлення відокремлених СІ». Крім цього, було проведено аналіз збігу окремих кластерів, розроблених з використанням зазначених методів та алгоритмів. За результатами порівняльного аналізу, алгоритм DBSCAN визнано кращим з тих методів та алгоритмів кластеризації, які брали участь у аналізі. Використання цього алгоритму дає змогу отримати описи архітектурних сутностей ІС як кластерів, що складаються з описів окремих функцій ІС, схожих за описами їхніх структур даних [4].

Використання алгоритму DBSCAN для вирішення Підзадачі 2 дозволило вдосконалити метод вирішення задачі ранньої ідентифікації СІ ІС. Цей вдосконалений метод представлено як послідовність виконання таких етапів.

Етап 0 (підготовчий). Сформувані множини формальних описів добре сформульованих вимог до ІС, які відображають точки зору Споживача й Постачальника ІТ-послуг [3].

Етап 1. Сформувані матриці виграшів Споживача й Постачальника ІТ-послуг.

Етап 2. Відзначити максимальні елементи в кожному стовпці матриці виграшів Споживача ІТ-послуг. Якщо в деякому стовпці даної матриці кілька максимальних елементів, то необхідно відзначити всі такі елементи.

Етап 3. Відзначити максимальні елементи в кожному рядку матриці виграшів Постачальника ІТ-послуг. Якщо в деякому рядку даної матриці кілька максимальних елементів, то необхідно відзначити всі такі елементи.

Етап 4. Провести пошук перетинання результатів Етапу 2 і Етапу 3.

Етап 5. Якщо результат виконання Етапу 4 не є порожньою множиною, зафіксувати відповідні варіанти описів архітектури створюваної ІС як описи раціональної архітектури даної ІС. У іншому випадку визнати, що опису раціональної архітектури ІС на чистих стратегіях не існує, та завершити застосування методу.

Етап 6. Виділити на зафіксованих описах окремих варіантів описів архітектури створюваної ІС описи архітектурних сутностей ІС як результат вирішення задачі класифікації описів окремих функцій даної ІС з використанням алгоритму щільнісної класифікації DBSCAN [4].

Етап 7. Виділити з результатів виконання Етапу 6 спрощені варіанти опису архітектури створюваної ІС за критерієм мінімальної наявності в описах архітектури ІС, визначених як шум, і надати ці описи Споживачу та Постачальнику ІТ-послуг. Завершити застосування методу.

У запропонованому методі Етап 0 відповідає за формування формальних описів функціональних вимог до створюваної ІС. Етапи 1-5 відповідають за вирішення Підзадачі 1, результатом якого є опис раціональної архітектури створюваної ІС на основі описів окремих функцій, що реалізують функціональні вимоги до цієї ІС. Етапи 6 та 7 відповідають за вирішення Підзадачі 2, результатом якого є спрощений опис архітектури створюваної ІС на основі виділених архітектурних сутностей як кластерів функцій цієї системи.

Результати експериментальної перевірки вирішення задачі ранньої ідентифікації СІ ІС наведено у [4]. Для перевірки було використано дані ІТ-проєкту розробки функціональної задачі «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри». Цю задачу реалізовано у вигляді окремого ІТ-продукту для розвитку можливостей інформаційно-аналітичної системи «Університет» Харківського національного університету радіоелектроніки (Україна). Під час перевірки для зменшення витрат часу було висунуто припущення про те, що в результаті виконання Етапів 0-5 було отримано опис раціональної архітектури функціональної задачі, який складався з десяти окремих функцій. Найменування цих функцій наведено у табл. 1 [4, 11].

Детально описи функціональної задачі «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри» наведено у [11]. Для формального опису функцій задачі було використано діаграми потоків даних та діаграми «Сутність – зв'язок». Для визначення відстаней між окремими функціями задачі було використано модифіковану відстань Чебишева [4, 11]. Усі розрахунки під час вирішення задачі ранньої ідентифікації СІ ІС із застосуванням алгоритму DBSCAN виконано в середовищі MS Excel 2016.

Найменування окремих функцій функціональної задачі «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри»

Функція	
Найменування	Позначення
Конвертація розділу «Навчальна робота	<i>CI1</i>
Формування розділу «Наукова робота»	<i>CI2</i>
Формування розділу «Методична робота»	<i>CI3</i>
Формування розділу «Організаційна робота»	<i>CI4</i>
Формування переліку посад та довгострокових доручень	<i>CI5</i>
Формування переліку рекомендованих до виконання робіт	<i>CI6</i>
Формування і ведення нормативно-довідкової інформації про Key Performance Indicators (KPI)	<i>CI7</i>
Формування KPI викладача і частини від KPI кафедри	<i>CI8</i>
Формування зведеної таблиці на навчальний рік	<i>CI9</i>
Формування вихідного документа «Індивідуальний план»	<i>CI10</i>

Під час вирішення задачі ранньої ідентифікації було розглянуто три варіанти опису архітектури функціональної задачі: монолітна архітектура; модульна архітектура; сервіс-орієнтована архітектура [4]. Для цих варіантів під час виконання Етапу 6 було отримано вісім різних рішень. З них за критерієм мінімальної наявності в описах кожного з варіантів архітектури тих CI, що були визначені як шум, під час виконання Етапу 7 було обрано три варіанти рішення, які наведено у табл. 2.

Таблиця 2

Результати вирішення задачі ранньої ідентифікації елементів конфігурації функціональної задачі «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри»

Значення <i>MinPts</i>	Значення ϵ	Рішення (позначення та зміст кластерів)
<i>MinPts</i> = 1 (сервіс-орієнтована архітектура)	$\epsilon = 60\%$	$C1 = \{CI7, CI6\}$
		$C2 = \{CI4, CI1, CI2, CI3, CI8, CI9, CI10, CI5\}$
<i>MinPts</i> = 2 (модульна архітектура)	$\epsilon = 60\%$	$C1 = \{CI4, CI1, CI2, CI3, CI8, CI9, CI10, CI5\}$
		<i>CI6, CI7</i> визнано шумом
<i>MinPts</i> = 1 (монолітна архітектура)	$\epsilon = 70\%$	$C1 = \{CI1, CI2, CI3, CI4, CI5, CI6, CI7, CI8, CI9, CI10\}$

У табл. 2 використані такі параметри алгоритму DBSCAN: *MinPts* – параметр, який повинен визначати поріг щільності найближчих сусідів досліджуваного CI; ϵ – ра-

діус, який повинен визначати межу області існування найближчих сусідів досліджуваних СІ (так звану міру близькості).

Слід визнати, що вдосконалений метод вирішення задачі ранньої ідентифікації має обмеження у застосуванні. Головним обмеженням є необхідність формального опису кожної добре сформульованої функціональної вимоги до створюваної ІС. Але можливості формування подібних описів на ранніх стадіях життєвого циклу ІС є занадто обмеженими. Тому головним напрямом подальших досліджень було визнано адаптацію способів обчислення відстані d між описами окремих функціональних вимог та функцій ІС до можливих варіантів формального опису цих вимог та функцій. Ще одним напрямом подальших досліджень слід визнати розробку моделі універсального опису функціональної вимоги та функції ІС, яка б могла враховувати можливість існування різноманітних описів цієї вимоги (або функції) з різними ступенями формальності.

Висновки. В результаті дослідження було вдосконалено метод вирішення задачі ранньої ідентифікації СІ ІС. Суть вдосконалення полягає у використанні алгоритму щільнісної кластеризації DBSCAN для виявлення архітектурних елементів як кластерів окремих функцій ІС. Використання вдосконаленого методу дозволяє автоматизувати процедуру синтезу опису архітектури ІС. Застосування цієї автоматизованої процедури сприяє підвищенню якості розробки ІС за рахунок виділення для проєктування множини архітектурних сутностей цієї системи, яка є значно меншою, ніж множина елементарних функцій цієї ж ІС.

ЛІТЕРАТУРА

1. Guide to the Software Engineering Body of Knowledge. Version 3.0 / eds. Bourque P., Fairley R.E. Washington, DC, United States: IEEE Computer Society, 2014. 335 p.
2. Quigley J.M., Robertson K.L. Configuration Management, Second Edition: Theory and Application for Engineers, Managers, and Practitioners. New York: Auerbach Publications, 2019. 453 p. DOI: <https://doi.org/10.1201/9780429318337>
3. Farayola O.A., Hassan A.O., Adaramodu O.R., Fakeyede O.G., Oladeinde M. Configuration management in the modern era: best practices, innovations, and challenges. *Computer Science & IT Research Journal*. 2023. Vol. 4, No 2. P. 140-157. DOI: <https://doi.org/10.51594/csitrj.v4i2.613>
4. Kozhanov A., Ievlanov M. Definition a clusterization method to solve the task of early identification of IT product configuration elements. *Eastern-European Journal of Enterprise Technologies*. 2026. Vol. 1. №. 2 (139). P. 77–90. DOI: <https://doi.org/10.15587/1729-4061.2026.352878>
5. Faitelson D., Heinrich R., Tyszberowicz Sh. Supporting software architecture evolution by functional decomposition. In *5th International Conference on Model-Driven Engineering and Software Development – MODELSWARD*. SciTePress, 2017. P. 435-442. DOI: <https://doi.org/10.5220/0006206204350442>

6. Cadavid H., Andrikopoulos V., Avgeriou P., Chris Broekema P. 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.infsof.2022.106984>
7. Felfernig A., Le V., Popescu A., Uta M., Tran T.N., Atas M. (2021). An Overview of Recommender Systems and Machine Learning in Feature Modeling and Configuration. *Proceedings of the 15th International Working Conference on Variability Modelling of Software-Intensive Systems (VaMoS '21)*. New York, NY, USA: Association for Computing Machinery, 2021. Article 16, P. 1–8. DOI: <https://doi.org/10.1145/3442391.3442408>
8. Abolfazli A., Spiegelberg J., Palmer G., Anand A. A Deep Reinforcement Learning Approach to Configuration Sampling Problem. *2023 IEEE International Conference on Data Mining (ICDM)*. Shanghai, China, 2023. P. 1-10. DOI: <https://doi.org/10.1109/ICDM58522.2023.00009>.
9. Sellami Kh., Saied M.A., Ouni A. A Hierarchical DBSCAN Method for Extracting Microservices from Monolithic Applications. *EASE '22: Proceedings of the 26th International Conference on Evaluation and Assessment in Software Engineering*. New York, NY, USA: Association for Computing Machinery, 2022. P. 201-210. DOI: <https://doi.org/10.1145/3530019.3530040>
10. Krause A., Zirkelbach C., Hasselbring W., Lenga S., Kroger D. Microservice Decomposition via Static and Dynamic Analysis of the Monolith. *2020 IEEE International Conference on Software Architecture Companion (ICSA-C)*. Salvador, Brazil, 2020. P. 9-16. DOI: <https://doi.org/10.1109/ICSA-C50368.2020.00011>
11. Ievlanov, M., Vasiltcova, N., Neumyvakina, O., Panforova, I. Development of a method for solving the problem of it product configuration analysis. *Eastern-European Journal of Enterprise Technologies*. 2022. Vol. 6. Iss. 2 (120). P. 6–19. DOI: <https://doi.org/10.15587/1729-4061.2022.269133>
12. Santos J.L., Martins L.E.G., Molléri J.S. Requirements extraction from model-based systems engineering: A systematic literature review. *Journal of Systems and Software*. 2025. Vol. 226. № 112407. DOI: <https://doi.org/10.1016/j.jss.2025.112407>
13. Evlanov M. Development of the model and method of selecting the description of rational architecture of information system. *Eastern-European Journal of Enterprise Technologies*. 2016. Vol. 1, No. 2(79). P. 4–12. DOI: <https://doi.org/10.15587/1729-4061.2016.60583>
14. Han J., Kamber M., Pei J. *Data Mining. Concepts and Techniques*. (Third Edition). Waltham: Morgan Kaufmann Publishers. 2011. URL: <https://shop.elsevier.com/books/data-mining-concepts-and-techniques/han/978-0-12-381479-1> (дата звернення 21.03.2026).
15. Schubert E., Sander J., Ester M., Kriegel H.P., Xu, X. DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. *ACM Transactions on Database Systems*. 2017. Vol. 42. Iss. 3. Article No.: 19, P. 1-21. DOI: <https://doi.org/10.1145/3068335>.

REFERENCES

1. Guide to the Software Engineering Body of Knowledge. Version 3.0 / eds. Bourque P., Fairley R.E. Washington, DC, United States: IEEE Computer Society, 2014. 335 p.

2. Quigley J.M., Robertson K.L. Configuration Management, Second Edition: Theory and Application for Engineers, Managers, and Practitioners. New York: Auerbach Publications, 2019. 453 p. DOI: <https://doi.org/10.1201/9780429318337>
3. Farayola O.A., Hassan A.O., Adaramodu O.R., Fakeyede O.G., Oladeinde M. Configuration management in the modern era: best practices, innovations, and challenges. *Computer Science & IT Research Journal*. 2023. Vol. 4, No 2. P. 140-157. DOI: <https://doi.org/10.51594/csitj.v4i2.613>
4. Kozhanov A., Ievlanov M. Definition a clusterization method to solve the task of early identification of IT product configuration elements. *Eastern-European Journal of Enterprise Technologies*. 2026. Vol. 1. №. 2 (139). P. 77–90. DOI: <https://doi.org/10.15587/1729-4061.2026.352878>
5. Faitelson D., Heinrich R., Tyszberowicz Sh. Supporting software architecture evolution by functional decomposition. In *5th International Conference on Model-Driven Engineering and Software Development – MODELSWARD*. SciTePress, 2017. P. 435-442. DOI: <https://doi.org/10.5220/0006206204350442>
6. Cadavid H., Andrikopoulos V., Avgeriou P., Chris Broekema P. 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.infsof.2022.106984>
7. Felfernig A., Le V., Popescu A., Uta M., Tran T.N., Atas M. (2021). An Overview of Recommender Systems and Machine Learning in Feature Modeling and Configuration. *Proceedings of the 15th International Working Conference on Variability Modelling of Software-Intensive Systems (VaMoS '21)*. New York, NY, USA: Association for Computing Machinery, 2021. Article 16, P. 1–8. DOI: <https://doi.org/10.1145/3442391.3442408>
8. Abolfazli A., Spiegelberg J., Palmer G., Anand A. A Deep Reinforcement Learning Approach to Configuration Sampling Problem. *2023 IEEE International Conference on Data Mining (ICDM)*. Shanghai, China, 2023. P. 1-10. DOI: <https://doi.org/10.1109/ICDM58522.2023.00009>.
9. Sellami Kh., Saied M.A., Ouni A. A Hierarchical DBSCAN Method for Extracting Microservices from Monolithic Applications. *EASE '22: Proceedings of the 26th International Conference on Evaluation and Assessment in Software Engineering*. New York, NY, USA: Association for Computing Machinery, 2022. P. 201-210. DOI: <https://doi.org/10.1145/3530019.3530040>
10. Krause A., Zirkelbach C., Hasselbring W., Lenga S., Kroger D. Microservice Decomposition via Static and Dynamic Analysis of the Monolith. *2020 IEEE International Conference on Software Architecture Companion (ICSA-C)*. Salvador, Brazil, 2020. P. 9-16. DOI: <https://doi.org/10.1109/ICSA-C50368.2020.00011>
11. Ievlanov, M., Vasilcova, N., Neumyvakina, O., Panforova, I. Development of a method for solving the problem of it product configuration analysis. *Eastern-European Journal of Enterprise Technologies*. 2022. Vol. 6. Iss. 2 (120). P. 6–19. DOI: <https://doi.org/10.15587/1729-4061.2022.269133>

12. Santos J.L., Martins L.E.G., Molléri J.S. Requirements extraction from model-based systems engineering: A systematic literature review. *Journal of Systems and Software*. 2025. Vol. 226. № 112407. DOI: <https://doi.org/10.1016/j.jss.2025.112407>
13. Evlanov M. Development of the model and method of selecting the description of rational architecture of information system. *Eastern-European Journal of Enterprise Technologies*. 2016. Vol. 1, No. 2(79). P. 4–12. DOI: <https://doi.org/10.15587/1729-4061.2016.60583>
14. Han J., Kamber M., Pei J. *Data Mining. Concepts and Techniques*. (Third Edition). Waltham: Morgan Kaufmann Publishers. 2011. URL: <https://shop.elsevier.com/books/data-mining-concepts-and-techniques/han/978-0-12-381479-1> (дата звернення 21.03.2026).
15. Schubert E., Sander J., Ester M., Kriegel H.P., Xu, X. DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. *ACM Transactions on Database Systems*. 2017. Vol. 42. Iss. 3. Article No.: 19, P. 1-21. DOI: <https://doi.org/10.1145/3068335>.

Received 30.03.2026.

Accepted 02.04.2026.

Published 30.04.2026

Improvement of the method for solving the task of early identification of information system configuration items

This study investigates the process that controls the IT project configuration.

The study addressed the problem of early identification of configuration items (CI) within an enterprise management information system (IS). Modern research in this area is mainly aimed at solving the task of services early identification during refactoring of software systems. The issue of applying artificial intelligence methods and, in particular, clustering to detect CI at the early stages of the IS life cycle has not been studied sufficiently.

The aim of the study is to develop a new or improve an existing method for solving the task of early identification of IS CI.

To achieve the aim of the study, the existing method of synthesizing descriptions of rational architecture creating IS based on descriptions of functional requirements for this IS was improved. The essence of this improvement is to use the DBSCAN clustering algorithm to detect architectural entities as clusters of individual IS functions. The results of solving the clustering task make it possible to simplify the description of the architecture creating IS by significantly reducing the number of CIs, provided that descriptions of architectural entities as clusters of individual IS functions will act as CIs.

The main features of the course and results of the experimental verification of the improved method are considered. The descriptions of ten functions of the task “Formation and maintenance of an individual plan of a scientific and pedagogical employee of the department” were used as the derived data for this verification. The result of using the method is three solution options, presented in the form of sets of one or two clusters. These options describe monolithic, modular and service-oriented IS architecture.

The use of the obtained results allows you to automate the procedure for synthesizing the description of the IS architecture. This automation will increase the quality of IS development by allocating a set of architectural entities of this IS for design. This set is much smaller than the set of elementary IS functions.

Keywords: information system, configuration item, early identification task, function, architectural entity, DBSCAN, architecture description.

Кожанов Адріан Євгенійович – аспірант кафедри інформаційних управляючих систем, Харківський національний університет радіоелектроніки.

ORCID <https://orcid.org/0009-0002-3586-6886>

Євланов Максим Вікторович – д.т.н., професор, професор кафедри інформаційних управляючих систем, Харківський національний університет радіоелектроніки.

ORCID <https://orcid.org/0000-0002-6703-5166>

Мороз Дмитро Максимович – доктор філософії, доцент кафедри програмного забезпечення комп'ютерних систем, Національний технічний університет «Дніпровська політехніка».

ORCID <https://orcid.org/0000-0003-2577-3352>

Kozhanov Adrian Yevgeniyovich – postgraduate student of the Department of Information Control Systems, Kharkiv National University of Radio Electronics.

ORCID <https://orcid.org/0009-0002-3586-6886>

Ievlanov Maksym Viktorovich – doctor of technical sciences, prof., professor of the Department of information control systems, Kharkiv National University of Radio Electronics.

ORCID <https://orcid.org/0000-0002-6703-5166>

Moroz Dmytro Maksymovich – PhD of technical sciences, Associate Professor of the Department of Software Engineering, Dnipro University of Technology.

ORCID <https://orcid.org/0000-0003-2577-3352>