УДК: 355.075.7 DOI: **10.33099/2311-7249/2025-54-3-33-54**

ТОЛМАЧОВ В'ячеслав Юрійович,
Національний університет оборони України, Київ, Україна,
https://orcid.org/0000-0002-3927-2041

МЕТОД СИНТЕЗУ РАЦІОНАЛЬНОЇ АРХІТЕКТУРИ ІНФОРМАЦІЙНОЇ СИСТЕМИ СУПРОВОДЖЕННЯ ВИПРОБУВАНЬ ОЗБРОЄННЯ ТА ВІЙСЬКОВОЇ ТЕХНІКИ

Метою статті *є розроблення методу синтезу раціональної архітектури інформаційної системи супроводження випробувань озброєння та військової техніки на основі інтеграції компонентно-ієрархічної технології, модифікованого алгоритму кластеризації CLOPE та теоретико-ігрового підходу.*

Методи дослідження. *Під час проведення дослідження застосовано методи системного аналізу, теорії множин, математичної логіки, а також спеціальні методи, що враховують компонентно-ієрархічну технологію, модифікований алгоритм кластеризації категорійних даних та теоретико-ігровий підхід. Зазначений методичний підхід дає змогу перетворити процес архітектурного синтезу на послідовну, керовану та формалізовану процедуру, що охоплює всі етапи від аналізу вимог до вибору оптимального проєктного рішення.*

Отримані результати дослідження. *Проведено декомпозицію інформаційної системи на чотири функціонально-ієрархічні рівні та п'ять шарів абстракції. Розроблено формалізовані математичні моделі у вигляді кортежів для опису архітектури та компонентного складу системи. Адаптовано модель представлення функціональних вимог на рівні знань, практичне застосування якої продемонстровано на прикладах формування функціональних вимог щодо утворення проєктів плануючого та звітного документів. Детально описано модифікований алгоритм синтезу варіантів архітектурних рішень на основі кластеризації категорійних даних CLOPE, що складається з чотирьох етапів і забезпечує автоматизоване генерування множини семантично узгоджених архітектурних альтернатив. Обґрунтовано вибір значення коефіцієнта відштовхування для формування сервіс-орієнтованої архітектури високої гранулярності. Розглянуто теоретико-ігрову модель для вибору раціональної архітектури через матриці виграшів та пошук рівноваги за Нешем. Сформовано формалізовану математичну модель запису розробленого методу у вигляді композиції п'яти функціональних перетворень, що створює теоретичну основу для його автоматизації.*

Елементи наукової новизни. *Наукова новизна зводиться до розробки цілісного, формалізованого методу синтезу архітектури, який вперше інтегрує компонентно-ієрархічну технологію, алгоритм кластеризації та теорію ігор в єдину наскрізну процедуру.*

Теоретична й практична значущість викладеного у статті. *Теоретичне значення отриманих результатів полягає у розвитку методології проєктування складних інформаційних систем військового призначення через формування підходів доказової архітектури. Практичною цінністю отриманих результатів для воєнно-оборонного сектору є можливість їх використання як методологічної основи для створення інформаційної системи супроводження випробувань, що сприятиме підвищенню ефективності випробувального процесу, оптимізації управлінських рішень та прискоренню прийняття на озброєння нових зразків озброєння та військової техніки.*

Ключові слова: *архітектура інформаційної системи, компонентно-ієрархічна технологія, теорія ігор, модель на рівні знань, семантична мережа, математична модель, система випробувань, супроводження випробувань.*

Вступ

Постановка проблеми. Ефективне функціонування системи озброєння Збройних Сил України та інших військових формувань неможливе без належної системи випробувань озброєння та військової (спеціальної) техніки (далі – ОВТ). Випробування є важливим етапом життєвого циклу будь-якого зразка ОВТ, що забезпечує перевірку його тактико-технічних характеристик, надійності та відповідності сучасним вимогам ведення бойових дій.

Прискорення технологічних циклів, поява нових видів озброєнь та необхідність оперативного впровадження модернізованих зразків висувають безпрецедентно високі вимоги до оперативності випробувального процесу та ефективності системи випробувань в цілому.

Сучасні зразки ОВТ є складними кіберфізичними системами, що інтегрують механічні, електронні, програмні та комунікаційні компоненти. Це призводить до експоненційного зростання обсягів та різноманітності даних, що генеруються під час випробувань: телеметрична інформація, показники великої кількості датчиків, дані балістичних

вимірювань, метеорологічні зведення, результати відеофіксації тощо. Отже, формується важлива суперечність між об'єктивною необхідністю оперативного, повного та достовірного аналізу величезних масивів даних і обмеженими можливостями існуючих, переважно застарілих, підходів до інформаційного супроводження цих процесів [1].

Важливо усвідомлювати, що ці виклики виходять за межі суто технічних проблем. Вони є симптомами глибшої, організаційно-управлінської проблеми – відсутності єдиної методології та стандартизованих процесів управління даними протягом усього циклу випробувань. Отже, автоматизація процесів випробувань за допомогою створення ефективної інформаційної системи супроводження випробувань (далі – ІС СВ) є не просто завданням з розробки програмного забезпечення, а комплексним проєктом з реінжинірингу «бізнес-процесів» спеціалізованої випробувальної організації. ІС СВ ОВТ являє собою складну людино-машинну систему, призначену для інтегрованої підтримки наукомістких процесів, що відбуваються протягом усього життєвого циклу випробувань, та покликана підвищити ефективність випробувань, скоротити час на їх проведення, мінімізувати вплив людського фактору та забезпечити підтримку прийняття рішень на всіх рівнях управління.

Однак створення такої складної, спеціалізованої ІС стикається з низкою науково-технічних проблем. Головною з них є задача синтезу раціональної архітектури системи. Архітектура ІС визначає її фундаментальну організацію, втілену в її компонентах, їх взаємозв'язках один з одним та з оточенням, а також принципи, що визначають її проєктування та еволюцію [2]. Визначення архітектури ІС є невід'ємною складовою технічних процесів життєвого циклу системи та ставить за мету створення альтернативних архітектур системи для обрання однієї або кількох альтернатив, які об'єднують інтереси зацікавлених сторін і відповідають вимогам до системи та висловлюють це в наборі узгоджених бачень [3]. Помилки, допущені на етапі синтезу архітектури, є суттєвими та можуть призвести до створення системи, яка не відповідає вимогам замовника, є негнучкою, нездатною до модернізації та вразливою до зовнішніх і внутрішніх загроз.

Специфіка предметної області випробувань ОВТ висуває до архітектури ІС СВ ОВТ суворі, часто суперечливі вимоги: висока надійність, захищеність даних, модульність, масштабованість, інтероперабельність з іншими системами, адаптивність до змін тощо. Існуючі загальновідомі методи та методології проєктування ІС (наприклад, Rational Unified Process (RUP), Microsoft Solutions Framework (MSF), Structured Analysis and Design Technique (SADT)) є переважно узагальненими і не враховують повною мірою цієї специфіки. Вони не надають формалізованого апарату для об'єктивного, науково обґрунтованого вибору найкращої архітектурної альтернативи з множини можливих.

Тому, наявні суперечності та проблемні аспекти формулюють наукову проблему, яка полягає у відсутності обґрунтованого формалізованого методу синтезу раціональної архітектури ІС СВ ОВТ. Такий метод повинен враховувати специфіку предметної області та забезпечувати системну інтеграцію всіх етапів випробувального процесу. Необхідність розроблення такого методу витікає з висновків попередніх досліджень, які, проаналізувавши існуючі підходи, вказали на їх недостатню ефективність для вирішення даного класу завдань [4].

Аналіз основних досліджень і публікацій. Питання синтезу ІС є предметом численних наукових досліджень, що демонструють значне методичне різноманіття. Нормативно-правовою основою для проєктування будь-якої ІС, зокрема військового призначення, є дотримання вимог національних та міжнародних стандартів, які формують засади для забезпечення надійності, безпеки та сумісності систем [5]. Значний внесок у розробку моделей та методів синтезу архітектури ІС зробили вітчизняні вчені. Зокрема, М. В. Євланов, Н. В. Васильцова, І. Ю. Панфьорова запропонували використання семантичних мереж для представлення вимог та теоретико-ігрових моделей для вибору раціональної архітектури [6]. Питання синтезу інформаційно-технічних структур систем критичного застосування досліджували А. А. Коваленко та Г. А. Кучук, зосереджуючись на аспектах надійності та безпеки [7]. Роботи І. В. Рубан, Г. А. Кучук, О. П. Давікоза присвячені концептуальним підходам до проєктування структур телекомунікаційних мереж [8]. Проблеми багатокритеріального синтезу розглядались у працях В. В. Калачова, С. С. Ткачук, Є. О. Меренті, Д. В. Третяк , де застосовувався метод аналізу ієрархій [9]. Практично орієнтовану реалізацію синтезу архітектури інтегрованої автоматизованої системи управління підприємства запропоновано І. Г. Цмоць, О. В. Скороход та Я. П. Кісь у науковій статті, де авторами розглянуто чотирирівневу структуру системи та принципи її побудови, а також запропоновано застосування компонентно-ієрархічної технології синтезу, яка передбачає поділ процесу розроблення складної системи на ієрархічні рівні та види програмного та апаратного забезпечення [10].

Фундаментальні засади проєктування архітектури ІС закладені у працях закордонних фахівців. Зокрема, спільна робота таких авторів як Р. Казман, М. Кляйн, М. Барбаччі, Т. Лонгстафф, Г. Ліпсон, Д. Карр'єре та їх метод аналізу компромісів в архітектурі (ATAM), стали класичними у сфері оцінки архітектурних рішень за атрибутами якості (Quality Attributes) [11]. М. Шоу та Д. Гарлан провели одну з перших класифікацій архітектурних стилів [12]. Концепції сервіс-орієнтованої архітектури (SOA) та мікросервісів, які детально описані у працях М. Фаулера, Дж. Льюїса [13] та С. Ньюмена [14], визначили сучасні тренди у побудові гнучких та масштабованих систем. Разом із тим, ці підходи здебільшого пропонують набір шаблонів та правил, залишаючи процес вибору та комбінування компонентів на розсуд архітектора.

Слід зазначити, що автором у попередніх дослідженнях [4] було проведено детальний порівняльний аналіз існуючих методичних підходів до синтезу ІС, в межах якого, на основі розробленої системи показників і критеріїв, адаптованої до специфіки функціонування спеціалізованої випробувальної організації, було виконано багатокритеріальне оцінювання широкого спектра методів. За результатами аналізу було зроблено висновок про те, що жоден з розглянутих методів у чистому вигляді не є достатнім для розв'язання задачі синтезу ІС СВ ОВТ. Водночас було обґрунтовано доцільність застосування комбінованого підходу, що інтегрує компонентно-ієрархічну технологію синтезу для структурної декомпозиції системи, алгоритми кластеризації для об'єктивного формування функціональних компонентів на основі вимог, та теоретико-ігрові моделі для формалізованого вибору оптимального архітектурного рішення.

Отже, незважаючи на наявність значного теоретичного доробку та попередньо обґрунтованого вибору перспективних напрямів, наукове завдання розроблення цілісного, формалізованого узагальненого методу, який би поєднував зазначені підходи в єдину логічно послідовну процедуру синтезу раціональної архітектури ІС СВ ОВТ, залишається невирішеним й актуальним.

Мета статті – розроблення методу синтезу раціональної архітектури інформаційної системи супроводження випробувань озброєння та військової техніки на основі інтеграції компонентно-ієрархічної технології, модифікованого алгоритму кластеризації CLOPE та теоретико-ігрового підходу.

Виклад основного матеріалу дослідження

Для досягнення поставленої мети необхідно вирішити низку конкретних завдань, а саме:

1. Формалізувати процес декомпозиції ІС СВ ОВТ на основі компонентно-ієрархічної технології, визначивши рівні ієрархії та склад компонентів.
2. Адаптувати та модифікувати алгоритм кластеризації CLOPE для задачі групування функціональних вимог та автоматизованого формування на їх основі логічних модулів (сервісів) ІС.
3. Розробити теоретико-ігрову модель, що дозволяє здійснити вибір раціонального варіанта архітектури ІС СВ ОВТ в умовах багатовимірності критеріїв оцінювання.
4. Об'єднати запропоновані підходи в єдиний послідовний алгоритм узагальненого методу синтезу архітектури ІС СВ ОВТ.

Одним із ключових завдань на етапі проєктування архітектури є декомпозиція системи – процес її послідовного поділу на менші, керовані та чітко визначені компоненти. Для вирішення цього завдання була обрана компонентно-ієрархічна технологія – методологія, що є не просто набором технічних прийомів, а цілісною філософією проєктування, яка забезпечує потужним інструментарієм для систематизованого розкладання складної системи на рівні абстракції, що дозволяє детально проаналізувати її структуру та функціональність.

В основі цієї технології лежить низка взаємопов'язаних принципів, які визначають її архітектурну логіку, зокрема:

системність (розгляд об'єкта проєктування як єдину цілісну системи, що складається з взаємопов'язаних частин і функціонує в певному середовищі для досягнення визначених цілей);

ієрархічність (полягає у послідовному розбитті системи «згори донизу», починаючи від єдиного абстрактного уявлення системи на найвищому рівні до множини конкретних елементів на найнижчому. Такий підхід дозволяє зосереджуватися на одному рівні абстракції за раз, не перевантажуючись надмірною деталізацією, що є критично важливим при роботі зі складними системами);

модульність (розбиття системи на функціонально завершені, відносно незалежні блоки – компоненти або модулів, з чітко визначеними інтерфейсами та функціональним призначенням. Це дозволяє здійснювати паралельну розробку, незалежне тестування, спрощує подальше обслуговування та модернізацію системи. Заміна або оновлення одного компонента не повинна призводити до каскадних змін в інших частинах системи, що забезпечує її гнучкість та довговічність);

відкритість та сумісність (побудова системи таким чином, щоб вона могла легко інтегруватися з іншими системами та розширюватися за рахунок додавання нових компонентів та модулів, що досягається шляхом стандартизації інтерфейсів та протоколів взаємодії);

можливість розвитку (забезпечує можливість постійного вдосконалення та оновлення складових частин системи: оновлення нормативно-довідкової бази даних новими стандартами, вдосконалення алгоритмів обробки даних чи модернізації користувацьких інтерфейсів тощо).

Сутність технології полягає у поділі єдиного процесу розробки на два ортогональні напрями: вертикальний (за ієрархічними рівнями абстракції) та горизонтальний (за видами забезпечення – алгоритмічне, програмне, апаратне). На кожному рівні ієрархії розв'язуються задачі відповідної складності, що описуються як одиницями інформації, так і алгоритмами їх обробки. Водночас, кожному рівню притаманний власний рівень деталізації. Зростанню номера рівня ієрархії (збільшенню глибини) відповідає збільшенню деталізації (гранулярності) як алгоритмічних, так і програмно-апаратних засобів. На вищих рівнях ієрархії компоненти являють собою складні, впорядковані сукупності (композиції) нижчерівневих компонентів, об'єднаних єдиним функціональним призначенням. Методологія послідовної декомпозиції, реалізує класичний принцип проектування «згори донизу». У результаті декомпозиції формується багаторівнева ієрархічна структура, в якій кожен компонент верхнього рівня координує та інтегрує функціонування підпорядкованих йому компонентів нижчих рівнів, створюючи тим самим відносини функціонального пріоритету та підпорядкування.

Застосування даної технології дозволяє структурувати складну предметну область, визначити чітку ієрархію управлінських та виконавчих функцій, а також декомпозувати загальну функціональність системи на набір керованих програмних компонентів (модулів). Такий підхід забезпечує не тільки логічність та прозорість архітектури, але й перетворити складну задачу синтезу на послідовність чітко визначених завдань, що вирішуються на відповідних рівнях ієрархії, забезпечуючи системність, модульність та керованість процесу проєктування.

Застосування системного підходу та принципів багаторівневої ієрархії для синтезу ІС СВ ОВТ надало змогу провести стратифікацію складної системи військового призначення та поділити її на чотири відокремлені взаємодіючі функціонально-ієрархічних рівнів (табл. 1). Разом із тим забезпечено відображення основних принципів військового управління класичної школи менеджменту, де функції стратегічного та оперативного планування чітко відокремлені від завдань безпосереднього виконання, що підтримуються єдиною інформаційно-технологічною та інфраструктурною базою забезпечення.

Таблиця 1

Опис функціональних рівнів ієрархії
інформаційної системи супроводження випробувань озброєння і військової (спеціальної) техніки

Ієрархічний рівень	Призначення	Функції	Одиниці інформації
Рівень 1 (стратегічний рівень – командування)	Забезпечення цілісності системи (координація та управління на найвищому рівні, що охоплює довгострокове планування, прийняття ключових рішень і координацію діяльності всієї системи)	загальне управління випробувальною діяльністю; довгострокове та поточне планування наукової та науково-технічної діяльності; здійснення заходів з внутрішнього контролю та управління ризиками; аналіз ефективності витрат та використання ресурсів (полігонно-випробувальної бази, вимірювального обладнання, персоналу, матеріально-технічних засобів); організація зовнішньої взаємодії; формування зведеної статистичної звітності.	Зведені дані, аналітичні звіти, довгострокові плани, прогнози.
Рівень 2 (оперативно-організаційний рівень – управління процесами випробувань)	Управління процесом випробувань у реальному масштабі часу, забезпечення узгодженої роботи всіх підсистем (забезпечення узгодженої роботи, управління ресурсами, координація, управління документами, супроводження процесів випробувань)	планування та координація робіт з проведення випробувань ОВТ; розподіл ресурсів (матеріально-технічних, кадрових); координація логістичних операцій; моніторинг та координація робіт на нижчих рівнях; планування та моніторинг стану полігонно-випробувальної та лабораторно-вимірювальної бази; оперативний контроль за виконанням визначених завдань та ключових етапів випробувань.	Дані про ресурси, логістичні потоки, результати виконання ПМ та інших планів.
Рівень 3 (рівень операційного супроводження випробувань)	Безпосереднє виконання випробувальних циклів, збір та обробка даних, контроль і управління параметрами обладнання, забезпечення стабільності та ефективності випробувальних процесів.	організація усіх етапів випробувань ОВТ; збір та обробка даних від вимірювально-обчислювального комплексу, систем бортових та зовнішньо-траєкторних вимірювань, метеорологічних систем, датчиків, відео- та фотофіксації; діагностика та профілактика несправностей обладнання; контроль відповідності виконуваних операцій затвердженим програмам і методикам; забезпечення синхронізації між різними системами, що забезпечують процеси випробувань ОВТ.	Експериментальні дані від вимірювальних засобів, параметри та характеристики ОВТ.
Рівень 4 (рівень інформаційно-технологічного та інфраструктурного забезпечення процесів випробувань)	Безпосереднє управління апаратними модулями, агрегатами та виконавчими механізмами, що виконують базові операції. Створення та підтримка технологічного фундаменту системи (бази даних, мережі, сервери). Забезпечення захисту і безпеки даних.	реєстрація та попередня обробка результатів випробувань; адміністрування серверів баз даних та файлових сховищ; забезпечення безперебійного функціонування локальної обчислювальної мережі; резервування та відновлення даних; управління правами доступу користувачів відповідно до їх ролей та повноважень; забезпечення взаємодії між програмними модулями вищих рівнів.	Результати вимірювань у цифровій формі, файли баз даних, журнали роботи серверів, ключі доступу та протоколи безпеки.

* Джерело: розроблено автором.

Разом із тим, інформаційні потоки в такій ієрархії рухаються як «згори донизу», так і «знизу догори», що робить архітектуру системи онтологічно узгодженою з реальною операційною діяльністю випробувальної установи. Зокрема, потік управління, який формується на першому рівні ієрархії, актуалізується через стратегічні цілі та плани, які в подальшому трансформуються в конкретні проєкти та завдання для другого рівня. На третьому рівні формуються визначені випробувальні сценарії та команди, які реалізуються за допомогою забезпечуючих засобів четвертого рівня.

Зворотній потік даних, що надходить від нижчих рівнів, забезпечує підґрунтя для прийняття обґрунтованих рішень на вищих щаблях. Він включає передачу сирих експериментальних даних та технічних звітів з четвертого рівня на третій, де вони аналізуються, агрегуються та перетворюються у структуровану інформацію для подальшого аналізу. Ця інформація, в свою чергу, надходить на другий та перший рівні, де слугує основою для корекції планів, перерозподілу ресурсів та оцінки ефективності всієї програми випробувань, замикаючи таким чином контур управління.

Функціональні рівні ієрархії ІС СВ ОВТ можна описати за допомогою впорядкованої множини рівнів, яка відображає концептуальну математичну модель та формально описує архітектуру та компонентний склад інформаційної системи:

$$S_{IC} = \bigcup_{i=1}^{4} L_i \qquad (1)$$

де S_{IC} – загальна система ІС СВ ОВТ;

i – індекс рівня ієрархії ($i = 1,2,3,4$);

L_i – ієрархічний рівень системи.

За таких умов, кожен ієрархічний рівень L_i описується кортежем, що містить його призначення (P_i), множину функцій (F_i) та множину одиниць інформації (I_i), з якими він оперує:

$$L_i = \langle P_i, F_i D_i \rangle \qquad (2)$$

де P_i – призначення i-го ієрархічного рівня, тобто його основну мету та роль у загальній структурі системи;

F_i – множина функцій, яка виконуються на i-го ієрархічному рівні;

D_i – множина одиниць інформації, яка створюється, обробляється або використовується на i-му ієрархічному рівні.

В свою чергу, множину функцій F_i та множину одиниць інформації D_i для кожного рівня можна визначити більш детально так:

$$F_i = \{ f_{i,j} \mid j = 1.2,..., k_i \} \qquad (3)$$

$$D_i = \{ d_{i,j} \mid j = 1,2,..., m_i \} \qquad (4)$$

де $f_{i,j}$ – позначає j-ту конкретну функцію в межах i-го ієрархічного рівня;

$d_{i,j}$ – позначає j-ту конкретну одиницю інформації в межах i-го ієрархічного рівня;

k_i – загальна кількість функцій, що виконуються на i-му ієрархічному рівні;

m_i – загальна кількість одиниць інформації, що використовується на i-му ієрархічному рівні.

Слід зазначити, що таке формалізоване подання дозволяє математично описати структуру ІС СВ ОВТ, полегшуючи аналіз, синтез та моделювання системи. Окрім функціональної декомпозиції системи за рівнями управлінської ієрархії, для повного розуміння її архітектури та взаємозв'язків доцільно розглянути її структуру з точки зору іншого рівня абстрагування, а саме – через призму ієрархічних шарів, що відображають її побудову від загальної концепції до конкретної апаратної реалізації. Такий підхід дозволяє послідовно деталізувати систему, переходячи від загальних вимог до специфічних програмно-технічних рішень.

Концептуальний шар. Це найвищий рівень абстракції, що визначає призначення, межі, цілі та основні сутності системи, які складають стратегічну архітектуру системи. На цьому шарі система розглядається як єдиний об'єкт, що вирішує головну задачу – підвищення ефективності інформаційного супроводження випробувань ОВТ. Головними сутностями, якими оперує система на цьому рівні, є: зразок ОВТ, випробування, програма та методики, протокол випробувань, випробувальна бригада (далі – ВБ), замовник, розробник. Концептуальний шар відповідає на питання «Що система робить?» та «Для кого?», формуючи основу для всіх подальших рівнів проєктування.

Функціональний шар. Цей шар деталізує концептуальну модель, описуючи конкретні функції та процеси, які повинна виконувати система для досягнення своєї мети. Функціональний шар відповідає на питання «Як система це робить?». Важливо зазначити, що фундаментальні питання функціонального моделювання та аналізу інформаційних потоків системи випробувань ОВТ були глибоко досліджені групою науковців у праці [15]. У межах передпроєктного етапу створення ІС СВ ОВТ автори виконали функціональну декомпозицію процесів випробування на глибину 3–4 ієрархічних рівнів, застосувавши для цього технологію SADT, а саме техніку функціонального моделювання IDEF0. Це дало змогу не тільки отримати вичерпний перелік інформаційних потоків, що циркулюють у системі, але й провести їх детальний аналіз: визначити типи зв'язків між функціями, оцінити важливість інформаційних ресурсів та внутрішньосистемну пов'язаність процесів. Такий підхід, що базується на методології IDEF0, допомагає формалізовано описати будь-яку функцію (процес) F_i у системі як перетворення множини вхідних об'єктів (Input, I) на множину вихідних об'єктів (Output, O) під впливом керуючих впливів (Control, C) за допомогою певних механізмів (Mechanism, M).

Математично це можна навести у вигляді кортежу:

$$f_{i,j} = \langle I_{i,j}, C_{i,j}, O_{i,j}, M_{i,j} \rangle \qquad (5)$$

де $f_{i,j}$ – j-та конкретна функція в межах i-го ієрархічного рівня (саме цей елемент є носієм

функціональності, що описується за методологією IDEF0 у [15]);

множина входів $I_{i,j}$ для функції $f_{i,j}$ (інформація або дані про об'єкти, що споживаються та перетворюються функцією):

$$I_{i,j} = \{i_{i,j,l} \mid l = 1.2,..., n_{i,j}\} \quad (6)$$

де: $n_{i,j}$ – кількість елементів у множинах входів для функції $f_{i,j}$;

$i_{i,j,l}$ – l-й конкретний елемент множини входів для j-ої конкретної функції в межах i-го ієрархічного рівня функції $f_{i,j}$;

множина керуючих впливів $C_{i,j}$ для функції $f_{i,j}$ (правила, стандарти, норми та інструкції, що регламентують виконання функції):

$$C_{i,j} = \{c_{i,j,l} \mid l = 1.2,..., p_{i,j}\} \quad (7)$$

де: $p_{i,j}$ – кількість елементів у множинах керуючих впливів для функції $f_{i,j}$;

$c_{i,j,l}$ – l-й конкретний елемент множини керуючих впливів для j-тої конкретної функції в межах i-го ієрархічного рівня функції $f_{i,j}$;

множина виходів $O_{i,j}$ для функції $f_{i,j}$ (інформація або матеріальні об'єкти, що є результатом виконання функції):

$$O_{i,j} = \{o_{i,j,l} \mid l = 1.2,..., q_{i,j}\} \quad (8)$$

де: $q_{i,j}$ – кількість елементів у множинах виходів для функції $f_{i,j}$;

$o_{i,j,l}$ – l-й конкретний елемент множини виходів для j-тої конкретної функції в межах i-го ієрархічного рівня функції $f_{i,j}$;

множина механізмів $M_{i,j}$ для функції $f_{i,j}$ (ресурси, необхідні для виконання функції, які при цьому не перетворюються (наприклад, випробувальна бригада, вимірювальне обладнання, матеріально-технічні засоби):

$$M_{i,j} = \{m_{i,j,l} \mid l = 1.2,..., r_{i,j}\} \quad (9)$$

де: $r_{i,j}$ – кількість елементів у множинах механізмів для функції $f_{i,j}$;

$m_{i,j,l}$ – l-й конкретний елемент множини механізмів для j-тої конкретної функції в межах i-го ієрархічного рівня функції $f_{i,j}$.

Разом із тим, сукупна множина одиниць інформації D_i для всього i-го ієрархічного рівня є об'єднанням усіх інформаційних об'єктів (входів, керуючих впливів та виходів) та усіх функцій, що виконуються на цьому рівні:

$$D_{i,j} = \left(\bigcup_{j=1}^{k_i} I_{i,j}\right) \cup \left(\bigcup_{j=1}^{k_i} C_{i,j}\right) \cup \left(\bigcup_{j=1}^{k_i} O_{i,j}\right) \quad (10)$$

Цей інтегрований формалізм дає змогу системно описати архітектуру ІС СВ ОВТ, поєднуючи макрорівневу ієрархічну структуру з мікрорівневим описом кожної окремої функції, що є необхідною умовою для подальшого детального проєктування та синтезу системи. Крім того, такий формальний підхід сприяє деталізації основних функцій, що реалізуються на цьому ієрархічному шарі:

планування випробувань: автоматизоване створення проєктів програм і методик (далі – ПМ), складання план-графіків, формування складу ВБ;

супроводження випробувань: надання доступу до документації, ведення електронних протоколів, збір телеметричної інформації тощо;

обробка результатів: виконання статистичної обробки даних, розрахунок оцінок тактико-технічних характеристик;

формування звітності: автоматизована генерація актів та інших звітних документів за результатами випробувань;

управління та аналітика: контроль за ходом випробувань, управління ризиками, надання аналітичної інформації керівництву.

Інформаційний шар. На цьому шарі визначається структура даних, необхідних для реалізації функцій системи. Він описує моделі даних, їх атрибути, зв'язки між ними та способи їх зберігання. Основою інформаційного шару ІС СВ ОВТ є три інтегровані бази даних [16]:

нормативно-довідкова БД: містить стандарти, класифікатори ОВТ, типові методики, шаблони документів.

технологічна БД: містить дані про полігонно-випробувальну базу, вимірювальне обладнання, їх характеристики та стан.

оперативна БД: зберігає всю поточну інформацію про конкретні випробування: дані про зразки, склад ВБ, електронні протоколи, результати вимірювань. Інформаційний шар забезпечує створення єдиного інформаційного простору та цілісність даних у системі.

Програмний шар. Цей шар описує конкретні програмні засоби, технології та архітектурні рішення, що використовуються для реалізації інформаційного та функціонального шарів. Для ІС СВ ОВТ на цьому шарі визначаються:

серверна частина: може бути розроблена з використанням сучасних мов програмування (Python, Java, SQL, Assembly language, C#, C++, Javascript тощо) та фреймворків (FastAPI, Django та Flask), що забезпечує бізнес-логіку та взаємодію з базами даних;

клієнтська частина: може бути реалізована як веб-додаток, що забезпечує графічний інтерфейс користувача;

система управління базами даних (СУБД): може бути реалізована за допомогою об'єктно-реляційної системи керування базами даних PostgreSQL, яка забезпечує надійне зберігання та управління даними;

файлове сховище: для зберігання документів, фото- та відеоматеріалів, пов'язаних з випробуваннями.

Технічний (апаратний) шар. Це фізичний фундамент системи, що включає апаратні засоби та мережеву інфраструктуру (сервери, мережеве обладнання, робочі станції тощо). Для ІС СВ ОВТ може бути побудована на основі архітектури «клієнт-сервер» і включати [17]:

серверний комплекс: сервер баз даних, комунікаційний сервер, сервер документів (файлове сховище);

локальну обчислювальну мережу: активне та пасивне мережеве обладнання, що об'єднує всі компоненти системи;

автоматизовані робочі місця: робочі станції користувачів (командування, інженерів-випробувачів, адміністраторів) та мобільні пристрої для роботи випробувальних бригад у польових умовах.

Отже, розгляд ІС СВ ОВТ через призму ієрархічних шарів дозволяє системно описати її архітектуру, забезпечуючи логічний перехід від загальної мети до конкретних програмно-технічних засобів її реалізації. Якщо попередній опис фокусувався на стратифікації системи за рівнями управлінської ієрархії та функціональним призначенням, то для переходу до подальших етапів необхідно ввести формальну модель, що описує систему з точки зору її компонентної будови.

Інформаційну систему S_{IC}, синтезовану за компонентно-ієрархічною технологією, можна представити у вигляді формалізованої моделі за допомогою впорядкованого кортежу з п'яти елементів:

$$S_{IC} =< C_{comp}, P_{UI}, E, R_{incl}, R_{\text{int } er} > \quad (11)$$

де C_{comp} – множина компонентів системи. Це універсальна множина всіх «будівельних блоків» системи на всіх рівнях ієрархії. Важливо зазначити, що раніше визначені функції ($f_{i,j}$) та механізми ($M_{i,j}$) є підмножинами цієї універсальної множини компонентів. Тобто, кожен програмний модуль, кожна випробувальна бригада, кожний елемент вимірювального обладнання є компонентом системи. Формально це можна записати за допомогою такого виразу:

$$\left(\bigcup_{i=1}^{4} F_i\right) \cup \left(\bigcup_{i=1}^{4}\bigcup_{j=1}^{k_i} M_{i,j}\right) \subseteq C_{comp} \quad (12)$$

Тому, формула математично об'єднує функціональний та ресурсний аспекти, розглядаючи їх як невід'ємні частини загальної компонентної структури системи.

P_{UI} – множина інтерфейсів (портів). Це множина формально визначених точок взаємодії, через які компоненти обмінюються даними або керуючими сигналами. Інформаційні потоки, описані раніше як множини входів ($I_{i,j}$), керуючих впливів ($C_{i,j}$) та виходів ($O_{i,j}$), реалізуються саме через ці інтерфейси. Кожен інтерфейс $p \in P_{UI}$ належить певному компоненту $c \in C_{comp}$.

E – функція ієрархічного рівня ($E : C \to \{1,2,3,4\}$). Ця функція присвоює кожному компоненту $c \in C_{comp}$ натуральне число, яке позначає його рівень в ієрархії, що повністю відповідає раніше визначеним функціональним рівням L_i. Наприклад, для будь-якої функції $f_{i,j}$ з множини F_i: $\forall f_{i,j} \in F_i \Rightarrow L(f_{i,j}) = i$.

R_{incl} – відношення включення ($R_{incl} \subseteq C_{comp} \times C_{comp}$). Це бінарне відношення, що визначає ієрархію вкладеності, іншими словами пара $(c_a, c_b) \in R_{incl}$ означає, що компонент c_a є частиною (підкомпонентом) компонента c_b, при цьому завжди виконується умова $L(c_a) > L(c_b)$. Наприклад, функція «Розробка програми випробувань» є частиною рівня «Підготовка документації на випробування».

$R_{\text{int } er}$ – відношення взаємодії ($R_{\text{int } er} \subseteq P_{UI} \times P_{UI}$). Це бінарне відношення, що описує функціональні зв'язки між компонентами через їх інтерфейси. Саме це відношення формалізує інформаційні потоки, описані в моделі IDEF0 [15]. Якщо вихід $o \in O_{i,j}$ функції $f_{i,j}$ є входом $i' \in I_{i,j}$ для функції $f_{i',j'}$, це означає, що існує пара $(p_a, p_b) \in R_{\text{int } er}$, де p_a – вихідний інтерфейс компонента функції $f_{i,j}$, а p_b – вхідний інтерфейс компонента функції $f_{i',j'}$.

Отже, формалізована модель поєднує два рівні опису компонентно-ієрархічної технології синтезу складної системи:

функціонально-інформаційний (через $L_i, P_i, F_i, I_i, f_{i,j}$), що відповідає на питання «Що робить система і з якою інформацією?»;

компонентно-структурний (через $C_{comp}, P_{UI}, E, R_{incl}, R_{\text{int } er}$), що відповідає на питання «З чого складається система і як її частини пов'язані між собою?».

Такий подвійний формалізм забезпечує повноту опису ІС СВ ОВТ, створюючи міцний теоретичний фундамент для її подальшого проєктування, розробки та модернізації. Це допомагає аналізувати не тільки логіку виконання процесів, але й створює підґрунтя для побудови структуру програмних та апаратних засобів, що їх реалізують.

Отже, аналіз предметної області та декомпозиція ІС СВ ОВТ на основі компонентно-ієрархічної технології

дозволили визначити загальну структуру системи та її основні функціональні блоки. Цей етап забезпечив макрорівневе розуміння системи, розділивши її на керовані та логічно відокремлені компоненти. Однак для подальшого синтезу раціональної архітектури необхідно перейти від загального структурного опису до формалізованого представлення конкретних функціональних вимог, які система має виконувати в межах цих компонентів. Цей крок є основним, оскільки саме на основі семантично насиченого опису вимог стає можливим їх об'єктивне групування в логічні модулі (сервіси) та подальший обґрунтований вибір архітектурного рішення. Для цього пропонується використати модель представлення вимог на рівні знань, що дає змогу зафіксувати не лише перелік функцій, а й об'єкти, з якими вони оперують, та зв'язки між ними. Такий підхід дає змогу представити вимогу не просто як текстовий опис, а як структуровану сукупність понять (фреймів), їх характеристик, інтерфейсів взаємодії та зв'язків між ними. У загальному вигляді модель представлення функціональної вимоги до ІС на рівні знань має вигляд [6]:

$$K_i^f = \{< d_n^{ij}, \{< d_{el\_fr}^{ij}, d_{el\_fr\_t}^{ij} >\} >, < d_g^{ij}, \{< d_{el\_if}^{ij}, d_{el\_if\_t}^{ij} >,\} >, < d_{fr\_rel\_n}^{ij}, \{< d_{el\_fr\_rel}^{ij}, d_{el\_fr\_rel\_t}^{ij} >\} >\} \quad (13)$$

де d_n^{ij} – опис найменування фрейму;

$d_{el\_fr}^{ij}$ – опис елемента фрейму;

$d_{el\_fr\_t}^{ij}$ – опис типу елемента фрейму;

d_g^{ij} – опис найменування інтерфейсу;

$d_{el\_if}^{ij}$ – опис елемента інтерфейсу;

$d_{el\_if\_t}^{ij}$ – опис типу елемента інтерфейсу;

$d_{fr\_rel\_n}^{ij}$ – опис назви зв'язку між інтерфейсами та/або фреймами;

$d_{el\_fr\_rel}^{ij}$ – опис елемента зв'язку;

$d_{el\_fr\_rel\_t}^{ij}$ – опис типу елемента зв'язку.

Слід зазначити, що між елементами моделі існують певні відношення належності:

відношення належності елементів фрейму конкретному фрейму;

відношення належності елементів інтерфейсу конкретному інтерфейсу;

відношення належності елементів зв'язку конкретному зв'язку.

Для формалізації функціональних вимог, виявлених на етапі аналізу предметної області, як основу пропонується використовувати адаптовану формальну модель, яка в загальному випадку має такий вигляд:

$$K_i^j = \{< d_n^{ij}, \{< d_{el\_fr}^{ij}, d_{el\_fr\_t}^{ij} >\} >, < d_{if}^{ij}, \{< d_{el\_if}^{ij}, d_{el\_if\_t}^{ij} >,\} >, < d_{fr\_rel\_n}^{ij}, \{< d_{el\_fr\_rel}^{ij}, d_{el\_fr\_rel\_t}^{ij} >\} >\} \quad (14)$$

де K_i^j – формалізоване представлення функціональної вимоги до ІС СВ ОВТ;

індекс i позначає порядковий номер конкретної функціональної вимоги у загальному переліку вимог до системи;

індекс j визначає точку зору або рівень представлення вимоги (наприклад: f_{bs} для загальносистемного погляду, $f_{\Pr}$ для погляду з «Постачальника» (розробника), f_U для погляду «Споживача» (користувача);

d_n^{ij} – опис найменування фрейму, який наводить основну сутність або поняття предметної області, що є центральним для даної вимоги (наприклад, «зразок ОВТ», «програма випробувань», «Акт випробувань»);

$< d_{el\_fr}^{ij}, d_{el\_fr\_t}^{ij} >$ – множина описів елементів фрейму та їх типів (елементи фрейму – це атрибути або слоти, що деталізують поняття, представлене фреймом (наприклад, для фрейму «зразок ОВТ» елементами можуть бути «Тактико-технічні характеристики», «Маса», «Калібр»);

$d_{el\_fr}^{ij}$ – опис елемента фрейму (назва атрибуту);

$d_{el\_fr\_t}^{ij}$ – опис типу елемента фрейму (тип даних, наприклад, «текст», «число»);

d_{if}^{ij} – опис найменування інтерфейсу, який описує точку взаємодії фрейму з іншими компонентами системи або зовнішнім середовищем (наприклад, «Затвердження документа», «Введення даних»);

$< d_{el\_if}^{ij}, d_{el\_if\_t}^{ij} >$ – множина описів елементів інтерфейсу та їх типів (елементи інтерфейсу деталізують процес взаємодії (наприклад, для інтерфейсу «Затвердження документа» елементами можуть бути «Посада особи, що затверджує», «Дата», «Підпис»);

$d_{el\_if}^{ij}$ – опис елемента інтерфейсу;

$d_{el\_if\_t}^{ij}$ – опис типу елемента інтерфейсу;

$d_{fr\_rel\_n}^{ij}$ – опис назви зв'язку між інтерфейсами та/або фреймами. Зв'язок визначає відношення між різними поняттями (наприклад, «ґрунтується на», «є частиною», «передається до»);

$< d_{el\_fr\_rel}^{ij}, d_{el\_fr\_rel\_t}^{ij} >$ – множина описів елементів зв'язку та їх типів, що характеризують цей зв'язок;

$d_{el\_fr\_rel}^{ij}$ – опис елемента зв'язку;

$d^{ij}_{el\_fr\_rel\_t}$ – опис типу елемента зв'язку.

Запропонована структуризація дає змогу перейти від неформального опису до машиночитаного подання, що є необхідною умовою для застосування алгоритмічних методів синтезу архітектури. Метод формувань представлень *i*-ої функціональної вимоги до ІС на рівні знань, які відображають погляди Постачальника, Споживача та загальносистемний погляд на створювану ІС, детально розглянуто у дослідженні [18]. На його основі реалізується процес перетворення функціональних вимог, визначених у моделях бізнес-процесів (наприклад, IDEF0), у формалізовані представлення K_i^j, який ґрунтується на переході від інформаційного рівня до рівня знань та складається з кількох послідовних кроків:

1. Ідентифікація ключових сутностей (фреймів) на основі структури даних. Для кожної функціональної роботи, описаної в IDEF0-моделі, та структури даних I_{ij}, з інформаційного представлення, виконується операція формування фрейму, де назва структури даних стає назвою фрейму (d_n^{ij}).

2. Деталізація атрибутів фрейму. На основі аналізу нормативних документів та структури даних визначається склад атрибутів, що описують дану сутність. Отже, атрибути документа або сутності предметної області безпосередньо перетворюються на атрибути (слоти) відповідного фрейму.

3. Ітеративна обробка. Кроки 1 та 2 повторюються для всіх структур даних, ідентифікованих в інформаційному представленні вимоги.

4. Визначення інтерфейсів та зв'язків. Після того, як усі структури даних перетворено на фрейми, аналізуються описи процесів та інформаційні потоки з IDEF0-моделі (вхідні, вихідні та керуючі потоки) для визначення того, як *i*-та функція взаємодіє з іншими процесами та які документи чи дані є для неї керуючими. На основі цього аналізу в діалоговому режимі з експертом предметної області формуються описи інтерфейсів (d_{if}^{ij}) та зв'язків ($d_{fr\_rel\_n}^{ij}$), що відображають взаємодію між створеними фреймами та їх зв'язок з іншими процесами.

Проілюструємо викладене вище декількома прикладами, розглянувши наступні функціональні вимоги: ІС СВ ОВТ має автоматизувати процес формування проєкту документу «Методика випробувань» для конкретного зразка ОВТ (табл. 2), а також відпрацювання протоколу випробувань (табл. 3). Застосовуючи описаний метод, її формалізоване представлення K_i^j матиме такий вигляд:

Таблиця 2

Модель представлення функціональної вимоги «Формування Методики випробувань»

Компонент опису моделі	Формальне представлення	Приклад заповнення для вимоги
1	2	3
Фрейм	d_n^{ij}	Методика випробувань
Елемент фрейму	$d_{el\_fr}^{ij}$	1. Об'єкт випробувань. 2. Мета випробувань. 3. Загальні положення. 4. Порядок і умови проведення випробувань. 5. Характеристики та показники для оцінювання і розрахункові співвідношення. 6. Умови і порядок проведення випробувань. 7. Заходи безпеки. 8. Оброблення, аналізування і оцінювання результатів випробувань. 9. Логістична підтримка випробувань. 10. Забезпечення охорони державної таємниці. 11. Звітність.
Тип елементу фрейму	$d_{el\_fr\_t}^{ij}$	Розділ документу
Інтерфейс	d_{if}^{ij}	Інтерфейс користувача для формування методики випробувань
Елемент інтерфейсу	$d_{el\_if}^{ij}$	1. Поле вводу тексту для розділів «Об'єкт, мета...». 2. Випадний список для вибору «Зразок ОВТ». 3. Поле для посилання на нормативний документ. 4. Кнопка «Зберегти проєкт методики».
Тип елемента інтерфейсу	$d_{el\_if\_t}^{ij}$	1. Поле вводу. 2. Випадний список. 3. Кнопка.

Закінчення табл. 2

1	2	3
Зв'язок	$d_{fr\_rel\_n}^{ij}$	1. Базується на. 2. Розробляється для. 3. Використовує знання з.
Елемент зв'язку	$d_{el\_fr\_rel}^{ij}$	1. («Методика випробувань», «Тактико-технічні вимоги»). 2. («Методика випробувань», «Зразок ОВТ»). 3. («Методика випробувань», «Нормативно-довідкова база даних»)
Тип елементу зв'язку	$d_{el\_fr\_rel\_t}^{ij}$	Зв'язок між фреймами

*Джерело: розроблено автором.

Таблиця 3

Модель представлення функціональної вимоги «Формування протоколу випробувань»

Компонент опису моделі	Формальне представлення	Приклад заповнення для вимоги
Фрейм	d_n^{ij}	Протокол випробувань
Елемент фрейму	$d_{el\_fr}^{ij}$	1. Реєстраційний номер документу. 2. Найменування перевірки. 3. Дата та місце проведення. 4. Випробувальна апаратура. 5. Умови проведення випробувань 6. Результати перевірки. 7. Висновки
Тип елементу фрейму	$d_{el\_fr\_t}^{ij}$	1. Розділ документа. 2. Таблиця результатів
Інтерфейс	d_{if}^{ij}	Інтерфейс користувача для відпрацювання Протоколу випробувань
Елемент інтерфейсу	$d_{el\_if}^{ij}$	1. Поле вводу тексту для розділів 2. Таблиця для внесення результатів 3. Кнопка "Зберегти протокол"
Тип елемента інтерфейсу	$d_{el\_if\_t}^{ij}$	1. Поле вводу. 2. Таблиця. 3. Кнопка.
Зв'язок	$d_{fr\_rel\_n}^{ij}$	1. Базується на. 2. Узагальнює. 3. Стосується
Елемент зв'язку	$d_{el\_fr\_rel}^{ij}$	1. («Протокол випробувань», «Методика випробувань»). 2. («Протокол випробувань», «Результати практичних випробувань»). 3. («Протокол випробувань», «Зразок ОВТ»).
Тип елементу зв'язку	$d_{el\_fr\_rel\_t}^{ij}$	Зв'язок між фреймами

* Джерело: розроблено автором.

Після формалізації функціональних вимог у вигляді семантичних мереж , наступним кроком є їх об'єктивне групування у логічно пов'язані компоненти майбутньої архітектури (сервіси, модулі).

$$Arch_{base} = \bigcup_{i=c+1}^{e} K_i^{f_{IS}} \quad (15)$$

де $c+1, e-$ ідентифікатори першої та останньої функціональної вимоги до ІС з сукупності сформульованих функціональних вимог до створюваної ІС СВ ОВТ.

У цьому початковому варіанті кожна функціональна вимога розглядається як окрема, незалежна ІТ-послуга. Завдання зводиться до виявлення прихованих зв'язків та групування цих вимог для формування більш раціональних архітектурних компонентів.

Мережа $Arch_{base}$ описуватиме варіант архітектури створюваної ІС, в якому кожна функціональна вимога буде описувати окрему ІТ-послугу. Для виявлення інших можливих варіантів описів архітектури створюваної ІС як сукупності ІТ-послуг слід враховувати такі особливості мережі:

відповідно до множини патернів, яка визначає правила та особливості формального опису представлень функціональних вимог на рівні знань, варіанти описів архітектури створюваної ІС

планується зберігати в спеціалізованому сховищі даних як сукупність транзакцій, кожна з яких описує окреме представлення $K_i^{f_{IS}}$;

представлення $K_i^{f_{IS}}$ є категорійними даними (формалізовані структуровані описи фреймів, інтерфейсів та зв'язків), числове подання яких ускладнено;

оцінка ступеня дублювання представлень $K_i^{f_{IS}}$ повинна ґрунтуватися на оцінці кількості повторень описів елементів мережі (15) у різних представленнях;

виявлення дублювання функціональних вимог повинно приводити до синтезу нових варіантів описів архітектури, в яких описи нових ІТ-послуг формують, групуючи початкові представлення $K_i^{f_{IS}}$.

Враховуючи зазначені особливості, задачу синтезу варіантів архітектури створюваної ІС треба розглядати як задачу кластеризації категорійних даних, алгоритм розв'язання якої ґрунтується на оптимізації глобального критерію. Це завдання можливо вирішити шляхом адаптації алгоритму кластеризації категорійних даних CLOPE (Clustering with SLOPE). Перевага цього алгоритму зводиться до його швидкості, масштабованості та здатності працювати з транзакційними даними, якими, по суті, є формалізовані функціональні вимоги, представлені як набори категорійних об'єктів (фреймів, інтерфейсів, зв'язків).

Завдання кластеризації полягає у тому, щоб згрупувати ці вимоги-транзакції у кластери IT_{acm_j} так, щоб вимоги всередині одного кластера були максимально схожими (висока внутрішньокластерна зв'язність), а вимоги з різних кластерів – максимально відмінними. Кожен такий кластер IT_{acm_j} є прообразом майбутнього архітектурного компонента (сервісу) ІС СВ ОВТ. Кожна формалізована вимога $K_i^{f_{IS}}$ розглядається як окрема транзакція, що є набором об'єктів ob (фреймів, інтерфейсів та зв'язків), опис представлення якої аналогічний (14). Множина описів ІТ-послуг IT_{acm} – це розбиття мережі $Arch_{base}$, за якого: $\{IT_{acm_1},...,IT_{acm_k}\} = Arch_{base}$ та $\left(IT_{acm_i} \neq \{\}\right) \wedge \left(IT_{acm_i} \bigcap IT_{acm_j} = \{\}\right)$ для $1 \le i$, $j \le k$, де k – кількість ІТ-послуг створюваної ІС.

Кожна ІТ-послуга IT_{acm_j} має такі характеристики:

$D\left|IT_{acm_j}\right|$ – множина унікальних об'єктів ob (фреймів, інтерфейсів та зв'язків);

$Occ(ob, IT_{acm_j})$ – кількість входжень об'єкта ob в опис ІТ-послуги IT_{acm_j};

$S(IT_{acm_j})$ – розмір гістограми j-го кластера, що дорівнює сумарній кількості всіх (не унікальних) об'єктів у всіх вимогах, що входять до цього кластера:

$$S(IT_{acm_j}) = \sum_{ob \in D(IT_{acm_j})} Occ(ob, IT_{acm_j}) \quad (16)$$

Цей показник відображає загальну інформаційну насиченість компонента.

$W(IT_{acm_j})$ – ширина гістограми j-го кластера, що дорівнює кількості унікальних об'єктів ob у вимогах, що входять до цього кластера:

$$W(IT_{acm_j}) = \left|D(IT_{acm_j})\right| \quad (17)$$

Цей показник характеризує різноманітність понять, якими оперує компонент.

$H(IT_{acm_j})$ – висота гістограми кластера, що є мірою його внутрішньої однорідності та зв'язності, що дорівнює відношенню загальної кількості об'єктів до кількості унікальних об'єктів:

$$H(IT_{acm_j}) = \frac{S(IT_{acm_j})}{W(IT_{acm_j})} \quad (18)$$

Цей показник є середньою частотою повторення кожного унікального об'єкта всередині кластера.

В основі алгоритму CLOPE лежить ідея максимізації глобальної цільової функції – «вартості» ($\Pr ofit$) розбиття на кластери, яка враховує висоту та ширину гістограми кожного кластера. Гістограма кластера відображає частоту входження кожного унікального об'єкта предметної області у вимоги, що належать до цього кластера. Алгоритм прагне створювати кластери з високими та вузькими гістограмами, що відповідає високій схожості об'єктів всередині кластера.

Для адаптації до задачі синтезу архітектури ІС СВ ОВТ, функція вартості $\Pr ofit(IT_{acm,r})$ розраховується за формулою:

$$\Pr ofit(IT_{acm,r}) = \frac{\sum_{j=1}^{k} \frac{S(IT_{acm_j})}{W(IT_{acm_j})^r} \times \left|IT_{acm_j}\right|}{\sum_{j=1}^{k} \left|IT_{acm_j}\right|} \quad (19)$$

де IT_{acm_j} – j-й кластер, що являє собою множину згрупованих функціональних вимог і відповідає потенційному архітектурному компоненту (сервісу);

$\left|IT_{acm_j}\right|$ – кількість загальносистемних представлень функціональних вимог на рівні знань $K_i^{f_{is}}$, що згруповані у j-й кластер та описують ІТ-послугу IT_{acm_j};

r – коефіцієнт відштовхування, основний параметр, що керує рівнем подібності всередині кластерів.

Отже, провідним завданням синтезу архітектурних варіантів для ІС СВ ОВТ є знаходження такого групування функціональних вимог (розбиття IT_{acm}), яке максимізує глобальну функцію вартості $\Pr ofit(IT_{acm,r})$ при заданому наборі початкових вимог $D(Arch_{base})$ та коефіцієнті відштовхування r: $IT_{acm} : \Pr ofit(IT_{acm,r}) \to \max$.

Однак, враховуючи, що початковий перелік вимог до складної системи мало коли буває повним, сенс покати один ідеальний максимально можливий варіант втрачається. У зв'язку з цим, постановка задачі адаптується до більш реалістичних умов, а саме: замість одного оптимального рішення, метою стає ідентифікація множини прийнятних архітектурних варіантів.

Прийнятним вважається таке розбиття IT_{acm}, для якого значення функції вартості знаходиться в межах визначеного діапазону від визначеного максимального значення: $\Pr ofit(IT_{acm,r}) \in [\Pr ofit_{\max} - \varepsilon; \Pr ofit_{\max}]$.

Тут $\Pr ofit_{\max}$ є найкращим досягнутим показником, а ε – це величина допустимої похибки. Якщо точне значення похибки неможливо встановити експертним шляхом, доцільно прийняти її на рівні 10% від максимального досягнутого значення прибутку, тобто $\varepsilon = 0{,}1 \cdot \Pr ofit_{\max}$.

Важливу роль у цьому процесі відіграє коефіцієнт відштовхування r, який є не просто технічним параметром, а інструментом архітектурного вибору, що визначає рівень деталізації майбутньої архітектури ІС. Для такої системи, як ІС СВ ОВТ, де критично важливими є модульність, гнучкість та можливість подальшої модернізації, перевага надається сервіс-орієнтованому підходу. Досягнення такої структури вимагає вибору відповідних значень r, які орієнтуються на формуванні більш дрібних та функціонально згуртованих сервісів. Розгляд законів композиції елементів функціональної структури ІС був детально висвітлений групою вітчизняних вчених у дослідженні [19]. Спираючись на матеріали проведеного дослідження рекомендується обрати значення коефіцієнт відштовхування на підставі даних, наведених у табл. 4.

Таблиця 4

Обгрунтування значення коефіцієнта відштовхування r для архітектури інформаційної системи супроводження випробувань озброєння військової (спеціальної) техніки

Значення коефіцієнта відштовхування	Характеристика кластерів, що формуються	Відповідність архітектурний стиль	Застосовність для ІС СВ ОВТ
$r = 1$	Формуються великі кластери, що об'єднують широкий спектр функцій з високою щільністю (елементи тісно згруповані), низькою зв'язністю (слабкі внутрішні зв'язки, елементи слабо пов'язані логічно) та високим зчепленням (сильні зовнішні залежності між кластерами, що ускладнює незалежність).	Монолітна, функціонально-орієнтована архітектура (підсистема)	Низька. Рекомендується уникати для ІС СВ ОВТ через жорсткість структури, низьку гнучкість і складність модернізації; підходить лише для простих систем без потреби в масштабуванні.
$r = 2$	Формуються кластери середнього розміру з помірною щільністю (баланс групування), середньою зв'язністю (помірні внутрішні зв'язки, елементи логічно пов'язані) та середнім зчепленням (помірні зовнішні залежності, з деякими перетинами кластерів).	Модульна архітектура	Середня. Забезпечує базовий рівень модульності, але може бути недостатньо гнучкою для еволюції системи. Рекомендується для перехідних етапів розробки, де потрібен баланс між простотою та модульністю.
$r \geq 3$	Формуються невеликі, високоспеціалізовані кластери з низькою щільністю (розріджені групи), високою зв'язністю (сильні внутрішні зв'язки, елементи функціонально згуртовані) та низьким зчепленням (слабкі зовнішні залежності, кластери автономні). Кожен кластер відповідає за вузьке коло завдань.	Сервіс-орієнтована архітектура (висока гранулярність)	Висока. Найкращим чином відповідає вимогам до масштабованості, гнучкості та можливості незалежного розвитку окремих компонентів ІС СВ ОВТ. Ідеально для систем з високими вимогами до стійкості, адаптивності та інтеграції нових компонентів.

* Джерело: розроблено автором.

Модифікований алгоритм синтезу варіантів описів архітектури створюваної ІС СВ ОВТ складається з таких етапів :

Етап 1. Сформувати початковий варіант опису архітектури створюваної ІС $Arch_{base}$.

Крок 1.1. Прийняти $n = \left| K_i^{f_{IS}} \right|$. На цьому кроці визначається загальна кількість функціональних вимог, які потрібно згрупувати. Це значення є вихідною точкою для всіх подальших циклів та визначає верхню межу для основного циклу перебору всіх вимог.

Крок 1.2. Сформувати множину описів ІТ-послуг IT_{acm}, виконавши операції $IT_{acm_j} = K_i^{f_{IS}}$. Створюється початкова архітектура, де кожна окрема вимога розглядається як самостійна ІТ-послуга (кластер). Це найбільш деталізований стан системи.

Крок 1.3. Для множини представлень $\left\{K_i^{f_{IS}}\right\}$ виконати операцію (15). Формально об'єднуємо всі вимоги в єдину семантичну мережу $Arch_{base}$, яка є початковим, неоптимізованим варіантом архітектури.

Етап 2. Встановити значення коефіцієнта відштовхування r, величини допустимої похибки ε та розрахувати значення функції $\Pr ofit(IT_{acm}, r)$. Тобто, на цьому етапі робиться вибір щодо бажаної гранулярності архітектури (значення r), величини допустимої похибки, яка адаптує максимально допустиме значення прибутку до більш реалістичних умов, та розраховується початкова "вартість" архітектури, яка буде точкою відліку для подальшої оптимізації.

Етап 3. Здійснити синтез кращих та/або прийнятних варіантів описів архітектури створюваної ІС СВ ОВТ.

Крок 3.1. Прийняти $\Pr ofit_{\max} = \Pr ofit(IT_{acm,r})$, $i = 1, j = 1, k = \left| IT_{acm} \right|$, де: i – це основний лічильник циклу, який перебирає кожну функціональну вимогу від 1 до n. Його завдання – забезпечити, щоб кожна вимога була проаналізована на предмет можливого переміщення в інший кластер.

j – це лічильник у циклі, який перебирає всі існуючі кластери від 1 до k. Його завдання – послідовно обирати кластер-джерело IT_{acm_j}, з якого буде тимчасово вилучено вимогу для перевірки можливості її переміщення.

На цьому кроці проводиться ініціалізація основного циклу оптимізації. Поточна вартість архітектури приймається за максимальну, і запускаються лічильники для перебору всіх вимог та кластерів.

Крок 3.2. Вибрати представлення $K_i^{f_{IS}}$. Починається перебір. Обирається перша (або наступна) функціональна вимога для аналізу.

Крок 3.3. Якщо $K_i^{f_{IS}} \in IT_{acm_j}$, то вилучити $K_i^{f_{IS}}$ з IT_{acm_j}. В іншому разі перейти до кроку 3.13. Вимога тимчасово "виймається" зі свого поточного кластера-джерела, щоб перевірити, чи не буде їй "краще" в іншому місці.

Крок 3.4. Прийняти $z = 0$, $m = 0$ – ініціалізація вкладеного циклу (скидається лічильник цільових кластерів z та прапорець покращення m), де:

z – це лічильник у вкладеному циклі, який також перебирає всі існуючі кластери (від 1 до k). Його завдання – послідовно обирати кластер-призначення (IT_{acm_z}), до якого тимчасово переміщується вимога для розрахунку нової вартості ($\Pr ofit$). Алгоритм перевіряє всі можливі «цілі», крім тієї, звідки вимога була взята (крок 3.5).

m – це бінарний прапорець (приймає значення 0 та 1), який слугує індикатором того, чи було знайдено краще рішення протягом повної ітерації по всіх вимогах. На початку кожної ітерації він скидається в 0 (крок 3.4). Якщо знаходиться переміщення, яке збільшує $\Pr ofit_{\max}$, m встановлюється в 1 (крок 3.9).

Крок 3.5. Якщо $z = 1$, то прийняти $z = z + 1$. Цей крок гарантує, що ми не будемо намагатися перемістити вимогу в той самий кластер, з якого її щойно вилучили.

Крок 3.6. Якщо $z > k$ (умова виходу з циклу перебору цільових кластерів.), то перейти до 3.12.

Крок 3.7. Включити $K_i^{f_{IS}}$ до IT_{acmz}, тобто вимога тимчасово "вставляється" в один з можливих цільових кластерів.

Крок 3.8. Розрахувати значення $\Pr ofit(IT_{acm,r})$ за виразом (19). Розраховується, якою стала б "вартість" всієї архітектури, якби це переміщення стало постійним.

Крок 3.9. Якщо $\Pr ofit(IT_{acm}, r) > \Pr ofit_{\max}$, то зафіксувати варіант опису архітектури створюваної ІС як множину IT_{acm}, скориговану з урахуванням виконання кроку 3.3 та кроку 3.7, прийняти $m = 1$ та перейти до кроку 3.12.

Іншим словами, якщо нова вартість виявилася кращою за попередній максимум, ми фіксуємо цей варіант як новий найкращий, встановлюємо прапорець $m = 1$ і припиняємо подальший пошук для поточної вимоги, оскільки вже знайшли для неї краще місце.

Крок 3.10. Якщо $\Pr ofit(IT_{acm}, r) \in [\Pr ofit_{\max} - \varepsilon; \Pr ofit_{\max}]$, то зафіксувати варіант опису архітектури створюваної ІС як множину IT_{acm}, скориговану з врахуванням виконання кроку 3.3 та кроку 3.7. Цей крок реалізує модифікацію алгоритму. Таким чином, зберігається не тільки найкращий варіант, але й усі "достатньо добрі" варіанти, які не сильно гірші за максимум. Це дає множину архітектурних альтернатив для подальшого аналізу.

Крок 3.11. Прийняти $z = z + 1$. Якщо $z \le k$, то

перейти до кроку 3.5 – перехід до наступного цільового кластера для перевірки.

Крок 3.12. Якщо $IT_{acm\,j} = \{\}$ та $m = 1$, то вилучити IT_{acm_j} з множини IT_{acm} і прийняти $k = k - 1$. Відбувається «прибирання» порожніх кластерів. Якщо в результаті успішного переміщення вихідний кластер спорожнів, він видаляється зі списку, а їх загальна кількість k зменшується.

3.13. Прийняття $j = j + 1$. Якщо $j \le k$, то перейти до кроку 3.3 – перехід до наступного вихідного кластера.

3.14. Прийняття $i = i + 1$. Якщо $i \le n$, то перейти до кроку 3.2 – перехід до наступної функціональної вимоги.

3.15. Якщо $m = 1$, то прийняти $i = 1$ та $j = 1$, після чого перейти до кроку 3.2. В протилежному випадку завершити виконання етапу методу.

Умова завершення всього процесу оптимізації. Якщо за повний прохід по всіх вимогах було знайдено хоча б одне покращення ($m = 1$), весь процес запускається знову з нової, кращої конфігурації. Якщо ж ні ($m = 0$), це означає, що подальші переміщення не покращують рішення, і процес зупиняється.

Етап 4. Вилучити з розгляду всі зафіксовані на етапі 3 варіанти описів архітектури створюваної ІС СВ ОВТ IT_{acm}, для яких не виконується умова $\Pr ofit(IT_{acm}, r) \in [\Pr ofit_{\max} - \varepsilon; \Pr ofit_{\max}]$. Завершити виконання методу.

Оскільки в процесі пошуку $\Pr ofit_{\max}$ міг збільшуватися, деякі раніше збережені “прийнятні” варіанти могли випасти з діапазону $[\Pr ofit_{\max} - \varepsilon; \Pr ofit_{\max}]$. Цей крок залишає тільки ті варіанти, які є дійсно близькими до фінального найкращого рішення.

Результатом виконання методу синтезу варіантів описів архітектури ІС є множина архітектурних альтернатив $\{Arch_1, Frch_2, ..., Arch_k\}$ – набір з N прийнятних та нетривіальних варіантів архітектури ІС СВ ОВТ. У кожній з цих альтернатив ступінь дублювання описів предметної галузі випробувань ОВТ у різних функціональних вимогах зведено до необхідного мінімуму, що забезпечує логічну цілісність та усуває надлишковість. Кожен з цих варіантів опису являє собою унікальну множину ІТ-послуг IT_{acm}. Своєю чергою, кожна з ІТ-послуг IT_{acm_j} є результатом об’єднання, аналогічного виразу (14), окремих формалізованих представлень функціональних вимог $K_i^{f_{IS}}$, які були включені до опису цієї ІТ-послуги за результатами виконання етапу 3 запропонованого методу.

Наявність множини формально обґрунтованих варіантів архітектури, отриманих на попередньому етапі, ставить нову задачу – вибір єдиного, найбільш раціонального рішення. Цей вибір не є суто технічним, оскільки він повинен враховувати часто суперечливі інтереси ключових стейкхолдерів проєкту. З одного боку, Замовник (спеціалізована випробувальна організація) зацікавлений у системі, що максимально точно відповідає унікальним і специфічним процесам випробувань ОВТ. З іншого боку, Розробник, який прагне до максимальної ефективності, мінімізації витрат і часу розробки шляхом повторного використання існуючих, універсальних компонентів та рішень.

Для вирішення цієї задачі багатокритеріальної оптимізації в умовах конфлікту інтересів пропонується використати теоретико-ігрову модель [6, 140], адаптовану до модифікованого алгоритму синтезу варіантів описів архітектури створюваної ІС СВ ОВТ. Ця модель розглядає процес вибору архітектури, що моделюється як безкоаліційна гра двох гравців з ненульовою сумою, де гравцями є Замовник (Споживач, U) та Розробник (Постачальник, Pr).

Гравець 1: Споживач (U). Ця роль представляє інтереси кінцевих користувачів ІС (керівний та інженерно-випробувальний склад спеціалізованої випробувальної організації). Їх глобальна мета – отримання системи з набором ІТ-послуг, який максимально повно, точно та зручно реалізує їх функціональні потреби та підтримує процеси випробувальної діяльності.

Гравець 2: Постачальник (Pr). Ця роль уособлює інтереси команди розроблення та супроводження системи (архітекторів, програмістів). Їхня глобальна мета – створення технічно досконалої, надійної, гнучкої та легко підтримуваної системи з таким набором ІТ-послуг, який найкраще відповідає вимогам Споживача.

Стратегіями кожного гравця є вибір одного з N варіантів архітектури, згенерованих модифікованим алгоритмом CLOPE. Результат гри (виграш) для кожного гравця при виборі певної архітектурної альтернативи визначається функцією виграшу, яка кількісно оцінює ступінь досягнення їх глобальних цілей.

Формально гра G_{IS} описується таким виразом:

$$G_{IS} =< \{\Pr, U\}, \{X_j\}_{j \in \{\Pr, U\}}, \{f_j\}_{j \in \{\Pr, U\}} > \quad (20)$$

де $\{\Pr, U\}$ – множина гравців, які беруть участь у грі, з яких:

U – Споживач – спеціалізована випробувальна організація, що представляє унікальні вимоги предметної області;

Pr – Постачальник – організація-розробник ІС;

$\{X_j\}_{j \in \{\Pr, U\}}$ – множина стратегій гри, де кожна стратегія – це вибір одного з N варіантів архітектури ІС;

$\{f_j\}_{j \in \{\Pr, U\}}$ – множина функцій виграшів для гравців.

Використання загальносистемних представлень функціональних вимог до ІС на рівні знань $K_i^{f_{IS}}$ забезпечує можливість інтерпретації операторів задоволення потреб $r^{\Pr}(K_i^{f_{\Pr}})$ та $r^{U}(K_i^{f_U})$ як функції, що здійснюють порівняння між загальносистемним представленням *i*-ї функціональної вимоги на рівні знань та її відповідними інтерпретаціями на рівнях знань Постачальника і Споживача.

Загалом ці функції виражаються у такій формі:

$$r^{\Pr}(K_i^{f_{\Pr}}) = f(K_i^{f_{IS}}, K_i^{f_{\Pr}}) = 1 - \frac{\left|K_i^{f_{IS}} \setminus K_i^{f_{\Pr}}\right|}{\left|K_i^{f_{IS}}\right|} \quad (21)$$

$$r^{U}(K_i^{f_U}) = f(K_i^{f_{IS}}, K_i^{f_U}) = 1 - \frac{\left|K_i^{f_{IS}} \setminus K_i^{f_U}\right|}{\left|K_i^{f_{IS}}\right|} \quad (22)$$

де: $K_i^{f_{\Pr}}$ – представлення *i*-ї функціональної вимоги до створюваної ІС на рівні знань Постачальника, формалізований опис якого аналогічний (14);

$K_i^{f_U}$ – представлення *i*-ї функціональної вимоги до створюваної ІС на рівні знань Споживача, формалізований опис якого аналогічний (14).

Для побудови функцій виграшу необхідно інтерпретувати загальносистемні вимоги $K_i^{f_{IS}}$ з позицій кожного гравця:

1. Представлення виграшу з погляду Постачальника:

розроблення нових ІТ-послуг та ІТ-сервісів, які задовольняють вимоги, що висуваються Споживачем;

адаптація розроблених раніше ІТ-послуг та ІТ-сервісів до особливостей вимог Споживача.

Тобто, Постачальник фокусується на технічній інтерпретація вимоги. Під час формування функції виграшу акцент робиться на максимальному повторному використанні існуючих програмних компонентів, застосуванні стандартних патернів проєктування, уніфікації інтерфейсів та мінімізації зв'язків між модулями. Наприклад, якщо вимога стосується формування звіту, Постачальник буде прагнути реалізувати її за допомогою універсального генератора звітів, який вже використовується в інших частинах системи, навіть якщо це вимагатиме від користувача додаткових кроків для налаштування.

2. Представлення виграшу з погляду Споживача:

формулювання вимог до ІС, ІТ-послуг та ІТ-сервісів, які враховують загальні та індивідуальні особливості автоматизації випробувальних процесів;

аналіз та вибір ІТ-послуг та ІТ-сервісів, які пропонуються Постачальником та відповідають сформульованим вимогам.

Отже, Споживач зосереджується на функціональній інтерпретації вимог, орієнтуючись на випробувальний процес. Під час формування функцій виграшу акцент робиться на повноті реалізації конкретної задачі користувача, зручності інтерфейсу та мінімізації дій для досягнення результату. Наприклад, для тієї ж вимоги щодо звіту, Споживач очікує спеціалізований інструмент, який за один клік генерує звіт точно встановленої форми, як це передбачено відповідним стандартом чи методикою випробувань. Отже, функції виграшу будуються на основі порівняння загальносистемного представлення вимог $K_i^{f_{IS}}$ з їх інтерпретаціями. Функцію виграшу Постачальника можна описати за допомогою такого виразу:

$$F_{\Pr} = \sum_{i=c+1}^{e} 1 - \frac{\left|K_i^{f_{IS}} \setminus K_i^{f_{\Pr}}\right|}{\left|K_i^{f_{IS}}\right|} \to \max \quad (23)$$

де *i* – ідентифікатор першої та останньої функціональної вимоги;

$\left|K_i^{f_{IS}} \setminus K_i^{f_{\Pr}}\right|$ – різниця множин, яка вказує на кількість елементів (фреймів, інтерфейсів, зв'язків) у загальносистемній вимозі, яких немає у представленні Постачальника, що, по суті, обсяг «унікальної» роботи, яку потрібно виконати з нуля;

$\frac{\left|K_i^{f_{IS}} \setminus K_i^{f_{\Pr}}\right|}{\left|K_i^{f_{IS}}\right|}$ – частка вимоги, яка не покривається готовими компонентами Постачальника;

вираз $\left(1 - \frac{\left|K_i^{f_{IS}} \setminus K_i^{f_{\Pr}}\right|}{\left|K_i^{f_{IS}}\right|}\right)$ – частка вимоги, яка покривається готовими компонентами та являється виграшом Постачальника по *i*-й вимозі.

Функція виграшу Споживача в грі матиме такий вигляд:

$$F_U = \sum_{i=c+1}^{e} 1 - \frac{\left|K_i^{f_{IS}} \setminus K_i^{f_U}\right|}{\left|K_i^{f_{IS}}\right|} \to \max \quad (24)$$

де $\frac{\left|K_i^{f_{IS}} \setminus K_i^{f_U}\right|}{\left|K_i^{f_{IS}}\right|}$ – частка загальносистемної вимоги, яка не відповідає специфічним потребам Споживача;

вираз $\left(1 - \frac{\left|K_i^{f_{IS}} \setminus K_i^{f_U}\right|}{\left|K_i^{f_{IS}}\right|}\right)$ – частка відповідності реалізації *i*-ї вимоги унікальним потребам Споживача.

Беручи до уваги наведене вище, матриця виграшів Постачальника набуває такого вигляду:

$$\mathrm{Pr}=\begin{pmatrix} 1-\dfrac{\left|K_{1(c+1)}^{f_{IS}} \setminus K_{1(c+1)}^{f_{\mathrm{Pr}}}\right|}{\left|K_{1(c+1)}^{f_{IS}}\right|} & \ldots & 1-\dfrac{\left|K_{1i}^{f_{IS}} \setminus K_{1i}^{f_{\mathrm{Pr}}}\right|}{\left|K_{1i}^{f_{IS}}\right|} & \ldots & 1-\dfrac{\left|K_{1e}^{f_{IS}} \setminus K_{1e}^{f_{\mathrm{Pr}}}\right|}{\left|K_{1e}^{f_{IS}}\right|} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1-\dfrac{\left|K_{j(c+1)}^{f_{IS}} \setminus K_{j(c+1)}^{f_{\mathrm{Pr}}}\right|}{\left|K_{j(c+1)}^{f_{IS}}\right|} & \ldots & 1-\dfrac{\left|K_{ji}^{f_{IS}} \setminus K_{ji}^{f_{\mathrm{Pr}}}\right|}{\left|K_{ji}^{f_{IS}}\right|} & \ldots & 1-\dfrac{\left|K_{je}^{f_{IS}} \setminus K_{je}^{f_{\mathrm{Pr}}}\right|}{\left|K_{je}^{f_{IS}}\right|} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1-\dfrac{\left|K_{k(c+1)}^{f_{IS}} \setminus K_{k(c+1)}^{f_{\mathrm{Pr}}}\right|}{\left|K_{k(c+1)}^{f_{IS}}\right|} & \ldots & 1-\dfrac{\left|K_{ki}^{f_{IS}} \setminus K_{ki}^{f_{\mathrm{Pr}}}\right|}{\left|K_{ki}^{f_{IS}}\right|} & \ldots & 1-\dfrac{\left|K_{ke}^{f_{IS}} \setminus K_{ke}^{f_{\mathrm{Pr}}}\right|}{\left|K_{ke}^{f_{IS}}\right|} \end{pmatrix} \quad (25)$$

Матриця виграшів Споживача має аналогічний вигляд, але з використанням представлення $K_i^{f_U}$:

$$U=\begin{pmatrix} 1-\dfrac{\left|K_{1(c+1)}^{f_{IS}} \setminus K_{1(c+1)}^{f_U}\right|}{\left|K_{1(c+1)}^{f_{IS}}\right|} & \ldots & 1-\dfrac{\left|K_{1i}^{f_{IS}} \setminus K_{1i}^{f_U}\right|}{\left|K_{1i}^{f_{IS}}\right|} & \ldots & 1-\dfrac{\left|K_{1e}^{f_{IS}} \setminus K_{1e}^{f_U}\right|}{\left|K_{1e}^{f_{IS}}\right|} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1-\dfrac{\left|K_{j(c+1)}^{f_{IS}} \setminus K_{j(c+1)}^{f_U}\right|}{\left|K_{j(c+1)}^{f_{IS}}\right|} & \ldots & 1-\dfrac{\left|K_{ji}^{f_{IS}} \setminus K_{ji}^{f_U}\right|}{\left|K_{ji}^{f_{IS}}\right|} & \ldots & 1-\dfrac{\left|K_{je}^{f_{IS}} \setminus K_{je}^{f_U}\right|}{\left|K_{je}^{f_{IS}}\right|} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1-\dfrac{\left|K_{k(c+1)}^{f_{IS}} \setminus K_{k(c+1)}^{f_U}\right|}{\left|K_{k(c+1)}^{f_{IS}}\right|} & \ldots & 1-\dfrac{\left|K_{ki}^{f_{IS}} \setminus K_{ki}^{f_U}\right|}{\left|K_{ki}^{f_{IS}}\right|} & \ldots & 1-\dfrac{\left|K_{ke}^{f_{IS}} \setminus K_{ke}^{f_U}\right|}{\left|K_{ke}^{f_{IS}}\right|} \end{pmatrix} \quad (26)$$

де k – кількість варіантів описів архітектури створюваної ІС, які сформовано за допомогою (14).

Рядки матриці виграшів Постачальника (25) відображають значення функції виграшу, що характеризує реалізацію сукупності ІТ-послуг у j-му варіанті опису архітектури ІС. Водночас стовпці цієї матриці показують значення функції виграшу Постачальника від реалізації окремої i-ї функціональної вимоги до ІС у різних варіантах архітектурних описів. Аналогічно, рядки матриці виграшів Споживача (26) відповідають значенням функції виграшу від реалізації сукупності ІТ-послуг у j-му варіанті опису архітектури, а стовпці – значенням виграшу Споживача від реалізації i-ї функціональної вимоги у різних архітектурних варіантах.

У випадку, коли всі описи фреймів, інтерфейсів та зв'язків, що відображають предметну галузь у межах i-ї функціональної вимоги й використовуються для побудови представлення $K_{ji}^{f_{IS}}$, вибрані з бібліотеки раніше реалізованих описів вимог до ІТ-послуг, виграш Постачальника від i-ї функціональної вимоги у j-му архітектурному варіанті дорівнюватиме 1. Якщо ж усі ці описи є новими для Постачальника, його виграш дорівнюватиме 0. У разі часткової новизни описів або їхньої побудови на основі абстрактних термінів предметної галузі, наявних у бібліотеці описів фреймів, інтерфейсів та зв'язків, значення виграшу Постачальника від реалізації i-ї функціональної вимоги у j-му архітектурному варіанті перебуватиме в межах від 0 до 1.

Аналогічно, якщо всі описи фреймів, інтерфейсів та зв'язків, що характеризують предметну галузь у межах i-ї функціональної вимоги до ІС й формують представлення $K_{ji}^{f_{IS}}$, відповідають концептуальному опису Споживача, виграш Споживача від реалізації i-ї функціональної вимоги у j-му архітектурному варіанті опису ІС дорівнюватиме 1. Якщо ж ці описи повністю нові для Споживача і не використовуються ним у власних моделях предметної галузі, виграш дорівнюватиме 0. У випадках, коли новими є лише частина описів фреймів, інтерфейсів та зв'язків або вони абстрагують сформульовані Споживачем фрейми чи інтерфейси, значення виграшу Споживача від реалізації i-ї функціональної вимоги у j-му архітектурному варіанті визначатиметься у діапазоні від 0 до 1.

Отже, раціональна архітектура ІС як сукупність ІТ-послуг формується шляхом знаходження результату гри (20), розв'язком якої є знаходження рівноваги за Нешем для біматричної гри в чистих стратегіях. Тобто пошук такої ситуації (пари стратегій), за якої жоден з гравців не може збільшити свій виграш, змінивши свою стратегію в односторонньому порядку, за умови, що інший гравець своєї стратегії не змінює. Архітектурний варіант, що

відповідає точці рівноваги за Нешем, і вважається раціональним, оскільки він забезпечує найкращий з можливих компроміс між інтересами розробників та кінцевих користувачів, що є запорукою успішного впровадження та довготривалої експлуатації системи.

Для визначення рівноважного стану за Нешем у процесі синтезу раціональної архітектури ІС необхідно враховувати множину стратегій гри (20). Як випливає з формул опису функцій виграшів (23) та (24), стратегії можуть бути подані через множину загальносистемних представлень функціональних вимог ІС на рівні знань, що відображають можливі варіанти архітектурних описів майбутньої системи. У цьому контексті будь-яку стратегію $X_j^{\{\Pr,U\}} \in \{X\}^{\{\Pr,U\}}$ доцільно інтерпретувати як семантичну мережу, яка описує результат виконання операції (15) над множиною загальносистемних представлень $\{K_{ji}^{f_{IS}}\}_{i=(c+1),...,e}^{\{\Pr,U\}}$.

Відповідно, рівновага за Нешем у грі (20) відповідає такому варіанту (або множині варіантів) архітектурних описів створюваної ІС $\{K_{ji}^{f_{IS}}\}_{i=(c+1),...,e}^{\{\Pr,U\}}$, для яких виконується умова:

$$\{K_{ji}^{f_{IS}}\}_i^{\{\Pr,U\}} \in \underset{\{K_{ji}^{f_{IS}}\}_i^{\{\Pr,U\}} \in \{\{K_{1i}^{f_{IS}}\},...,\{K_{ji}^{f_{IS}}\}\}_i^{\{\Pr,U\}}}{\arg\max} f^{\{\Pr,U\}}\left(\{K_{-ji}^{f_{IS}}\}^{\{\Pr,U\}} \middle\| \{K_{ji}^{f_{IS}}\}^{\{\Pr,U\}}\right) \quad (27)$$

де $\{K_{ji}^{f_{IS}}\}_i^{\{\Pr,U\}}$ – представлення i-ї функціональної вимоги у j-му варіанті опису архітектури ІС, що розглядається у просторі стратегій Постачальника та Споживача;

$\underset{\{K_{ji}^{f_{IS}}\}_i^{\{\Pr,U\}} \in \{\{K_{1i}^{f_{IS}}\},...,\{K_{ji}^{f_{IS}}\}\}_i^{\{\Pr,U\}}}{\arg\max}$ – оператор, що виділяє аргумент (множину варіантів архітектурних описів), на яких функція досягає максимального значення;

$\{\{K_{1i}^{f_{IS}}\},...,\{K_{ji}^{f_{IS}}\}\}_i^{\{\Pr,U\}}$ – множина всіх можливих представлень функціональних вимог у різних варіантах описів архітектури ІС, сформованих на основі бібліотеки знань для Постачальника і Споживача;

$f^{\{\Pr,U\}}$ – функція виграшу, яка враховує одночасно інтереси Постачальника та Споживача та визначає, наскільки ефективним є вибраний варіант архітектури з позиції обох сторін;

$\left(\{K_{-ji}^{f_{IS}}\}^{\{\Pr,U\}} \middle\| \{K_{ji}^{f_{IS}}\}^{\{\Pr,U\}}\right)$ – умовна нотація, що позначає паралельне узгодження або спільний розгляд представлень у контексті двох сторін. Іншими словами, це об'єднання поглядів Постачальника та Споживача на одну й ту саму функціональну вимогу.

$j = 1,..., k$ – індекс варіанта опису архітектури ІС зі сторони Постачальника, а k – кількість таких варіантів;

$i = (c+1),..., e$ – індекс варіантів опису архітектури ІС зі сторони Споживача, від першої $(c+1)$ до останньої e.

Умова формалізує вибір такого представлення функціональних вимог $\{K_{ji}^{f_{IS}}\}_i^{\{\Pr,U\}}$, яке максимізує виграшну функцію $f^{\{\Pr,U\}}$, враховуючи узгоджені інтереси Постачальника і Споживача. Тобто, шукається той варіант архітектурного опису ІС, що буде найбільш оптимальним одночасно для обох сторін.

Алгоритм пошуку рівноваги за Нешем у чистих стратегіях біматричної гри (20) можна описати за допомогою таких етапів:

Крок 1. Побудувати матрицю виграшів Постачальника (25) та Споживача (26).

Крок 2. У кожному стовпці матриці (26) позначити найбільші значення. Якщо у стовпці цієї матриці таких елементів декілька, то позначити всі такі елементи.

Крок 3. У кожному рядку матриці (25) знайти максимальні значення та зафіксувати їх. Якщо у рядку цієї матриці таких елементів декілька, то необхідно відмітити всі такі елементи.

Крок 4. Здійснити перетин результатів попередніх двох кроків, визначивши ті елементи, які одночасно є максимальними в рядках і стовпцях (тобто максимальні елементи, які розміщені на тих самих позиціях у кожній матриці).

Крок 5. Якщо отримана множина є непорожньою, відповідні архітектурні описи ІС визнаються раціональними. Якщо ж перетин відсутній, робиться висновок, що опис раціональної архітектури ІС у чистих стратегіях не існує. Завершити виконання методу.

Коли множини максимальних елементів матриць (25) та (26) не перетинаються, рівновага в чистих стратегіях недосяжна. У цьому випадку застосовують аналіз у змішаних стратегіях, що дозволяє враховувати ймовірнісні комбінації дій. Для цього найчастіше використовують методи лінійного програмування, зокрема, симплекс-метод або процедуру Лемке–Хаусона [6, 143].

Слабким місцем застосування методу пошуку рівноваги за Нешем у чистих стратегіях біматричної гри є зростання рівня невизначеності та ризиків під час ухвалення рішень, що супроводжують процес створення ІС. Водночас у задачі синтезу раціональної архітектури ІС як цілісної сукупності ІТ-послуг цей недолік не має визначального значення. Це пояснюється тим, що побудова архітектури передбачає подальшу координацію дій Постачальника й Споживача після завершення гри, з метою погодження та затвердження проєктної документації на розроблення ІС СВ ОВТ.

Застосування теоретико-ігрової моделі є завершальним аналітичним етапом у процесі синтезу, який перетворює множину технічно прийнятних архітектурних варіантів на єдине, стратегічно обґрунтоване рішення. Такий підхід формалізує неминучий конфлікт інтересів між прагненням до максимальної функціональної відповідності (позиція Споживача) та необхідністю оптимізації ресурсів розробки (позиція Постачальника). Модель дозволяє кількісно оцінити кожен архітектурний варіант з обох точок зору, переводячи суб'єктивні переваги у площину об'єктивних числових показників. Отже, вибір раціональної архітектури для ІС СВ ОВТ перестає бути інтуїтивним актом і перетворюється на пошук обґрунтованого компромісу, що забезпечує найкращий з можливих баланс між специфічними потребами випробувальної діяльності та практичними обмеженнями процесу розробки.

Після детального розгляду окремих складових – методу формалізації вимог на основі компонентно-ієрархічної технології, методу генерації архітектурних альтернатив за допомогою алгоритму кластеризації та методу вибору раціонального рішення на основі теорії ігор – виникає необхідність об'єднати ці підходи в єдину, цілісну та послідовну методологію. Наступним кроком є формулювання узагальненого методу, який би інтегрував усі розглянуті етапи в єдиний алгоритм, що забезпечує наскрізний процес синтезу раціональної архітектури ІС СВ ОВТ від початкового аналізу до фінального, обґрунтованого проєктного рішення.

Процес синтезу раціональної архітектури, що об'єднує описані вище етапи, може бути представлений у вигляді єдиної формалізованої моделі. Ця модель описує весь процес як послідовне застосування функціональних перетворень, де вихід одного етапу є входом для наступного. Використовуючи принципи теорії множин та математичної логіки, зокрема оператор композиції функцій ($\circ$), узагальнений метод синтезу можна визначити як функцію, що перетворює початкову множину неструктурованих вимог до інформаційної системи випробувань ($\mathrm{Re}\, g_{TS}$) на єдину раціональну архітектуру ($Arch_{rational}$).

Узагальнений вираз для методу має такий вигляд:

$$Arch_{rational} = (Nash_{Eq} \circ G_{form} \circ CLOPE_{\mathrm{mod}}(r,\varepsilon) \circ K_{form} \circ Decomp_{CHT})(\mathrm{Re}\, g_{TS}) \quad (28)$$

Цей вираз формально описує наскрізний процес, який складається з п'яти послідовних функціональних перетворень, що детально розглядаються нижче.

1. Функція декомпозиції ($Decomp_{CHT}$).

Першим кроком є застосування компонентно-ієрархічної технології, що формалізується функцією:

$$Decomp_{CHT} : \mathrm{Re}\, g_{TS} \to S_{IC} \quad (29)$$

Ця функція приймає на вхід початкову, неформальну множину вимог до системи ($\mathrm{Re}\, g_{TS}$) і перетворює її на структуровану, формалізовану модель інформаційної системи S_{IC}. Модель S_{IC} описується кортежем із п'яти елементів $S_{IC} =< C_{comp}, P_{UI}, E, R_{incl}, R_{\mathrm{int}\,er} >$, як визначено у виразі (11). Цей кортеж включає множину всіх компонентів системи, їх інтерфейси, ієрархічні рівні та відношення включення і взаємодії, створюючи повну структурну карту системи.

2. Функція формалізації знань (K_{form}).

Наступний крок переводить структурну модель системи на рівень знань, що необхідно для подальшого семантичного аналізу:

$$K_{form} : S_{IC} \to \{K_i^j\} \quad (30)$$

Функція K_{form} приймає на вхід структуровану модель S_{IC} і генерує множину $\{K_i^j\}$ – сукупність усіх функціональних вимог, представлених на рівні знань. Кожна вимога $K_i^{f_{IS}}$ є формалізованим описом, що включає фрейми, їх атрибути, інтерфейси та зв'язки, відповідно до моделі, представленої у виразі (14). Цей етап перетворює структурні елементи на семантично насичені об'єкти, придатні для подальшої кластеризації.

3. Функція генерації архітектурних альтернатив ($CLOPE_{\mathrm{mod}}(r,\varepsilon)$).

Маючи множину формалізованих вимог, застосовується модифікований алгоритм CLOPE для їх групування:

$$CLOPE_{\mathrm{mod}}(r,\varepsilon) : \{K_i^j\} \to \{Arch_k\} \quad (31)$$

Ця функція представляє роботу алгоритму кластеризації. Вона приймає на вхід множину вимог $\{K_i^j\}$ та два ключові параметри: коефіцієнт відштовхування r, що визначає бажану гранулярність архітектури та допуск ε, що визначає діапазон прийнятності рішень. Алгоритм ітеративно групує вимоги у кластери (IT_{acm}), максимізуючи глобальну функцію вартості $\mathrm{Pr}\,ofit_{\max} = \mathrm{Pr}\,ofit(IT_{acm}, r)$. Результатом є множина $\{Arch_k\}$ – набір з k архітектурних альтернатив, кожна з яких є унікальним розбиттям початкових вимог і відповідає критерію прийнятності $\mathrm{Pr}\,ofit(IT_{acm}, r) \in [\mathrm{Pr}\,ofit_{\max} - \varepsilon; \mathrm{Pr}\,ofit_{\max}]$.

4. Функція формування гри (G_{form}).

Отримана множина архітектурних альтернатив стає основою для побудови теоретико-ігрової моделі:

$$G_{form} : \{Arch_k\} \to G_{IC} \quad (32)$$

Функція G_{form} перетворює множину архітектурних варіантів $\{Arch_k\}$ на повну формальну модель безкоаліційної гри G_{IC}. Архітектурні альтернативи стають множиною стратегій $\{X_j\}_{j\in\{\Pr,U\}}$ для двох гравців: Постачальника ($\Pr$) та Споживача (U). Функція також включає розрахунок матриць виграшів для кожного гравця (вирази (25) та (26)) на основі їх індивідуальних функцій виграшу $F_{\Pr}$ та F_U, які кількісно оцінюють відповідність кожної архітектури їх інтересам.

5. Оператор пошуку рівноваги Неша ($Nash_{Eq}$).

Завершальним етапом є розв'язання сформованої гри для знаходження оптимального рішення:

$$Nash_{Eq} : G_{IC} \to Arch_{rational} \quad (33)$$

Оператор $Nash_{Eq}$ являє собою концепцію розв'язку гри. Він приймає на вхід повністю визначену гру G_{IC} і знаходить ту пару стратегій (тобто ту архітектурну альтернативу), яка задовольняє умові Рівноваги за Нешем, як це формалізовано у виразі (27). Ця архітектура, що відповідає точці рівноваги, і є кінцевим результатом усього процесу – раціональною архітектурою $Arch_{rational}$, яка забезпечує найкращий можливий компроміс між інтересами сторін.

Інтеграція компонентно-ієрархічної технології, модифікованого алгоритму CLOPE та теоретико-ігрового підходу в єдиний послідовний алгоритм створює потужну методологію, яка надає ряд суттєвих переваг, що є вкрай важливими для проєктування такої складної системи, як ІС СВ ОВТ:

Відслідковуваність трансформацій: алгоритм створює чіткий, документований та аудиторський шлях від початкових, часто нечітких, вимог до кінцевого, формально обґрунтованого архітектурного рішення. Кожен крок трансформації даних є явним і логічно пов'язаним з попереднім.

Відтворюваність: за умови однакових вхідних даних та параметрів, процес гарантовано дасть однаковий результат. Це усуває варіативність, пов'язану з людським фактором, і робить процес проєктування більш науковим та менш залежним від індивідуальної інтуїції архітектора.

Об'єктивність: метод замінює суб'єктивні, евристичні рішення на кожному етапі формалізованими, керованими процедурами. Групування вимог базується на математичній оптимізації, а фінальний вибір – на розв'язанні формальної гри, що мінімізує упередженість.

Обґрунтованість: кінцеве архітектурне рішення являється математично доведеним результатом, який демонструє найкращий можливий баланс між конфліктуючими інтересами зацікавлених сторін. Це надає архітектурному вибору потужну доказову базу, що є вкрай важливим у проєктах з високою вартістю помилки.

Отже, вираз (28) формалізує наскрізний ітераційний процес визначення майбутньої конфігурації ІС в математичну модель синтезу раціональної архітектури ІС СВ ОВТ, яка описує весь послідовний процес функціональних перетворень множини неструктурованих вимог до ІС випробувань ($\mathrm{Re}\, g_{TS}$) на єдину раціональну архітектуру ($Arch_{rational}$).

Висновки й перспективи подальших досліджень

Синергетичне поєднання підходів з трьох різних наукових галузей: системної інженерії (компонентно-ієрархічна технологія), інтелектуального аналізу даних (модифікований алгоритм кластеризації CLOPE) та економічної теорії (теоретико-ігрові моделі) дало змогу вирішити поставлену наукову задачу щодо формалізації цілісного інтегрованого методу синтезу раціональної архітектури для такої складної системи як інформаційна система супроводження випробувань озброєння та військової техніки. Отримані результати дослідження зводяться до такого:

на основі компонентно-ієрархічної технології синтезу виділено чотири відокремлені функціонально-ієрархічні рівні інформаційної системи супроводження випробувань озброєння та військової техніки системи, а саме: стратегічний рівень командування, оперативно-організаційний рівень управління процесами випробувань, рівень операційного супроводження випробувань та рівень інформаційно-технологічного й інфраструктурного забезпечення. Таке структурування забезпечує онтологічну узгодженість архітектури з реальною операційною діяльністю спеціалізованої випробувальної установи та відображає основні принципи військового управління;

послідовна деталізація інформаційної системи супроводження випробувань озброєння та військової техніки через призму п'яти ієрархічних шарів: концептуального, функціонального, інформаційного, програмного та технічного дало змогу від загальних вимог, що висуваються до таких систем, перейти до конкретної апаратно-програмної реалізації;

використання математичної моделі у вигляді кортежів дало змогу формалізувати інформаційну систему як об'єднання чотирьох ієрархічних рівнів. Для кожного рівня ієрархії, якому відповідає свій кортеж, визначено основну мету та роль у загальній структурі системи, множину функцій та одиниць інформації, з якими він оперує, а також компонентно-структурне подання через множини компонентів інформаційної системи, інтерфейсів, функцій ієрархічних рівнів та відношень включень та взаємодії між її компонентами;

для семантичного опису вимог запропоновано адаптовану модель представлення функціональних вимог до інформаційної системи на рівні знань, практичне застосування якої продемонстровано на прикладах утворення таких функціональних вимог як формування методики та протоколу випробувань;

на основі кластеризації категорійних даних CLOPE, який забезпечує автоматизоване генерування множини семантично угоджених архітектурних альтернатив і вибору значення коефіцієнту відштовхування, деталізовано опис модифікованого алгоритму синтезу варіантів описів архітектурних компонентів інформаційної системи;

для вибору раціональної архітектури системи використана теоретико-ігрова модель, яка формалізує конфлікт інтересів між користувачами та розробником інформаційної системи через матриці виграшів і пошук рівноваги за Нешем. Зазначене дає змогу кількісно оцінити не лише кожен архітектурний варіант компонентів інформаційної системи з позиції обох сторін, але й знайти науково обґрунтований компроміс між гравцями;

формалізація об'єднання за компонентно-ієрархічною технологією синтезу, опису функціональних вимог інформаційної системи на рівні знань, генерації альтернатив через модифікований алгоритм CLOPE, формування гри та пошуку рівноваги Неша дало змогу розробити метод синтезу раціональної архітектури інформаційної системи супроводження випробувань озброєння та військової техніки.

Практична значущість отриманих результатів зводиться до можливості їх використання як методологічної основи для створення спеціалізованої інформаційної системи, яка сприятиме підвищенню ефективності випробувального процесу, оптимізації управлінських рішень та прискоренню прийняття на озброєння нових (модернізованих) зразків військової та військової техніки.

Перспективи подальших досліджень вбачаються у зосередженні на двох взаємопов'язаних напрямах. В першу чергу, це розроблення методики синтезу інформаційної системи для підтримки прийняття рішень в системі супроводження випробувань озброєння та військової техніки для деталізації кожного етапу розробленого у статті методу синтезу, визначити конкретні процедури та критерії прийняття рішень на кожному кроці, а також встановити взаємозв'язки між етапами. Крім того, логічним продовженням роботи є розроблення детального алгоритму синтезу такої інформаційної системи, який перетворить розроблений метод на практичний інструментарій формальних правил виконання кожної операції, умови переходів між етапами та механізми верифікації проміжних результатів. Це створить основу для подальшої автоматизації процесу архітектурного синтезу та створення відповідного програмного забезпечення супроводження випробувального процесу.

Список бібліографічних посилань

1. Скиба О. В., Доманов І. О., Рибачок Д. В., Скиба А. О. Підходи щодо підвищення ефективності підготовки і проведення випробувань шляхом створення єдиної інформаційно-довідкової системи та автоматизації діяльності науковців інституту. *Збірник наукових праць ДНДІ ВС ОВТ.* 2023. Вип. 4(18). С. 104–111. DOI: 10.37701/dndivsovt.18.2023.15. **2. ДСТУ ISO/IEC/IEEE 42010:2018** Інженерія систем і програмних засобів. Опис архітектури (ISO/IEC/IEEE 42010:2011, IDT). URL: https://online.budstandart.com/ua/catalog/doc-page?id_doc=77960 (дата звернення 05.09.2025). **3. ДСТУ ISO/IEC/IEEE 15288:2016** Інженерія систем і програмного забезпечення. Процеси життєвого циклу систем (ISO/IEC/IEEE 15288:2015, IDT). URL: https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=71827 (дата звернення 05.09.2025). **4. Толмачов В. Ю.** Методичний підхід щодо вибору методу синтезу інформаційної системи супроводження випробувань озброєння та військової техніки. *Сучасні інформаційні технології у сфері безпеки та оборони*. 2025. Вип. 53(2). С. 70–83. DOI: 10.33099/2311-7249/2025-53-2-70-83. **5. Перелік Національних стандартів України для створення, впровадження та супроводження автоматизованих і інформаційних систем**. URL: http://nbuv.gov.ua/node/1469 (дата звернення 08.09.2025). **6. Євланов М. В., Васильцова Н. В., Панфьорова І. Ю.** Моделі та методи синтезу опису раціональної архітектури інформаційної системи. *Вісник наукового університету «Львівська політехніка». Серія «Інформаційні системи та мережі».* 2015. № 829. **7. Коваленко А. А., Кучук Г. А.** Методи синтезу інформаційної та технічної структур системи управління об'єктом критичного застосування. *Сучасні інформаційні системи.* 2018. Т. 2. № 1. С. 22–27. DOI: 10.20998/2522-9052.2018.1.04. **8. Рубан І. В., Кучук Г. А., Давікоза О. П.** Концептуальний підхід до синтезу структури інформаційно-телекомунікаційної мережі. *Системи обробки інформації.* 2013. №7(114). С. 106–112. **9. Калачова В. В., Ткачук С. С., Меренті Є. О., Третяк Д. В.** Багатокритеріальний синтез організаційної структури білінгової інформаційної системи методом аналізу ієрархій. *Системи обробки інформації.* 2020. № 2(161). С. 22–28. DOI: 10.30748/soi.2020.161.03. **10. Цмоць І. Г., Скорохода О. В., Кісь Я. П.** Синтез інтегрованих автоматизованих систем управління підприємством. *Вісник Національного університету «Львівська політехніка». Серія: Інформаційні системи та мережі.* 2015. № 829. С. 246–254. **11. Kazman R., Klein M., Barbacci M., Longstaff T., Lipson H., Carriere J.** The Architecture Tradeoff Analysis Method. URL: https://insights.sei.cmu.edu/documents/1186/1998_005_001_16646.pdf (Accessed: 08 September 2025). **12. Mary S.** Software architecture: Perspectives on an emerging discipline. Upper Saddle River, N.J : Prentice Hall, 1996. 242 p. DOI: 10.5555/231003. **13. Фаулер М.** Microservices a definition of this new architectural term URL: https://martinfowler.com/articles/microservices.html (дата звернення: 08.09.2025). **14. Newman S.** Building Microservices. O'Reilly Media, Incorporated, 2015. 280. DOI: 10.5555/2904388. **15. Корнієнко І. В, Корнієнко С. П., Шевага В. В, Руденко О. В., Жирна О. В.** Функціональне моделювання й аналіз інформаційних потоків системи випробувань озброєння та військової техніки. *Збірник наукових праць Державного науково-дослідного інституту випробувань і сертифікації озброєння та військової техніки*. 2021. Вип. 10(4), С. 83–93. DOI: 10.37701/dndivsovt.10.2021.09. **16. Корнієнко І. В., Корнієнко С. П., Казначей С. М., Руденко О. В., Толмачов В. Ю.** Узагальнена структура інформаційно-довідкової бази даних ОВТ інформаційної системи

супроводження випробувань. *Збірник наукових праць Державного науково-дослідного інституту випробувань і сертифікації озброєння та військової техніки*. 2022. Вип. 11(1). С. 66–77. DOI: 10.37701/dndivsovt.11.2022.08. **17. Скиба О. В., Доманов І. О., Рибачок Д. В., Скиба А. О.** Підходи щодо підвищення ефективності підготовки і проведення випробувань шляхом створення єдиної інформаційно-довідкової системи та автоматизації діяльності науковців інституту. *Збірник наукових праць ДНДІ ВС ОВТ.* 2023. Вип. 4(18). С. 104–111. DOI: 10.37701/dndivsovt.18.2023.15. **18. Евланов М. В.** Разработка методов формирования представления сформулированного требования к информационной системе на уровне знаний. *Восточно-европейский журнал передовых технологий*. 2015. Вып. № 4/3(76). С. 4–11. URL: http://nbuv.gov.ua/UJRN/Vejpte_2015_4%283%29__2 (дата звернення: 16.09.2025). **19. Левыкин В. М., Евланов М. В.** Метамодель функциональной структуры информационной системы. *Нові технології.* 2006. Вип. 1(12). С. 67–72. URL: http://openarchive.nure.ua/handle/document/3958 (дата звернення: 16.09.2025).

METHOD OF SYNTHESIS OF THE RATIONAL ARCHITECTURE OF AN INFORMATION SYSTEM FOR SUPPORTING TESTS OF ARMAMENT AND MILITARY EQUIPMENT

TOLMACHOV Vyacheslav, National Defence University of Ukraine, Kyiv, Ukraine,
https://orcid.org/0000-0002-3927-2041

Formulation of the problem in general. *The purpose of this article is to develop a generalised method for synthesising a rational architecture for an information system supporting the testing of Armament and Military Equipment, based on the integration of component-hierarchical technology, the modified CLOPE clustering algorithm, and a game-theoretical approach.*

Research methods. *During the research, methods of systems analysis, set theory, and mathematical logic, as well as specialised methods including component-hierarchical technology, a modified algorithm for clustering categorical data, and a game-theoretical approach, were applied. This methodological approach enables the transformation of the architectural synthesis process into a sequential, manageable, and formalised procedure that encompasses all stages, from requirements analysis to the selection of an optimal design solution.*

Literature review. *The analysis of scientific research demonstrates substantial methodological contributions in the field of information system synthesis. Both domestic and international studies have established the theoretical foundations of architectural design, proposing a variety of approaches – from semantic modelling of requirements and compromise assessment methods to modern paradigms of service-oriented architecture. However, a comparative analysis shows that none of the existing methods, in their pure form, provides a comprehensive solution to the problem of synthesising an information system for supporting tests of Armament and Military Equipment, which highlights the necessity of developing an integrated formalised method.*

Research results. *The article performs a decomposition of the information system into four functional-hierarchical levels and five layers of abstraction. Formalised mathematical models in the form of tuples are developed to describe the system's architecture and component composition. A model for representing functional requirements at the knowledge level is adapted, with its practical application demonstrated through examples of forming functional requirements for creating draft planning and reporting documents. A modified algorithm for synthesising architectural design variants, based on CLOPE categorical data clustering, is described in detail. This algorithm consists of four stages and ensures the automated generation of a set of semantically consistent architectural alternatives. The selection of the repulsion coefficient value for forming a high-granularity, service-oriented architecture is justified. A game-theoretic model is considered for selecting a rational architecture by using payoff matrices and searching for a Nash equilibrium. Finally, a formalised mathematical model of the developed method is formulated as a composition of five functional transformations, creating the theoretical basis for its automation.*

Research novelty. *The scientific novelty lies in the development of a holistic, formalised method for architecture synthesis, which for the first time integrates component-hierarchical technology, a clustering algorithm, and game theory into a single end-to-end procedure.*

Theoretical and practical significance. *The theoretical significance of the obtained results lies in the advancement of the methodology for designing complex military information systems through the development of evidence-based architecture approaches. The practical value of the results for the defence sector lies in their potential use as a methodological foundation for creating an information system to support testing, which will enhance the efficiency of the testing process, optimise managerial decision-making, and accelerate the adoption of new Armament and military equipment into service.*

Conclusions and future work. *The prospects for further research are seen in focusing on two interrelated directions. First and foremost, this involves the development of a methodology for synthesising an information system to support decision-making within the test support system for weapons and military equipment. Such a methodology will enable the detailed description of each stage of the developed synthesis method, the definition of specific procedures and decision-making criteria at every step, and the establishment of interconnections between the stages. In addition, a logical continuation of this work is the development of a detailed algorithm for synthesising such an information system, which will transform the developed method into a practical toolkit of formal rules for performing each operation, conditions for*

transitions between stages, and mechanisms for verifying intermediate results. This will lay the foundation for further automating the architectural synthesis process and developing corresponding software to support the testing process.

Keywords: *information system architecture, component-hierarchical technology, game theory, knowledge-level model, semantic network, mathematical model, testing system, test support.*

References

1. Skyba, O. V., Domanov, I. O., Rybachok, D. V., Skyba, A. O., (2023). Approaches to increase the efficiency of training and testing by creating a uniform information and reference system and automating the activities of the institute's scientists. *Zbirnyk naukovykh prats DNDI VS OVT,* 4(18), 104-111. DOI: 10.37701/dndivsovt.18.2023.15. **2. DSTU ISO/IEC/IEEE 42010:2018,** (2018). Systems and Software Engineering. Architecture Description. DSTU ISO/IEC/IEEE 42010:2011, IDT. Available at: https://online.budstandart.com/ua/catalog/doc-page?id_doc=77960 [Accessed: 05 September 2025]. **3. DSTU ISO/IEC/IEEE 15288:2016,** (2018). Systems and Software Engineering. Systems Lifecycle Processes. DSTU ISO/IEC/IEEE 15288:2015, IDT. Available at: https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=71827 [Accessed: 05 September 2025]. **4. Tolmachov, V.,** (2025). Methodical approach to selecting a synthesis method for an information system for Armament and Military equipment testing support. *Suchasni informatsiini tekhnolohii u sferi bezpeky ta oborony,* 2(53)/2025, 70-83. DOI: 10.33099/2311-7249/2025-53-2-70-83. **5. National Standards of Ukraine for the Creation, Implementation and Maintenance of Automated and Information Systems. Available** [online], (2025). Available at: http://nbuv.gov.ua/node/1469 [Accessed: 08 September 2025]. **6. Yevlanov, M. V., Vasyltsova, N. V., Panforova, I. Yu.,** (2015). Models and methods for synthesizing descriptions of rational architecture of an information system. *Visnyk naukovoho universytetu «Lvivska politekhnika». Seriia «Informatsiini systemy ta merezhi»,* 829, 135-152. **7. Kovalenko, A. A., Kuchuk, H. A.,** (2018). Methods for synthesising informational and technical structures of the critical application object's control system. *Suchasni informatsiini systemy*, 2, 1, 22–27. DOI: 10.20998/2522-9052.2018.1.04. **8. Ruban, I. V., Kuchuk, H. A., Davikoza, O. P.,** (2013). Conceptual approach to the structure design of a telecommunication network. *Systemy obrobky informatsii,* 7(114), 106–112. **9. Kalachova, V. V., Tkachuk, S. S., Merenti, Ye. O., Tretiak, D. V.,** (2020). Multi-criterion synthesis of the organisational structure of the billing information system by the method of analysis of hierarchies. *Systemy obrobky informatsii,* 2(161), 22-28. DOI: 10.30748/soi.2020.161.03. **10. Tsmots, I. H., Skorokhoda, O. V., Kis, Ya. P.,** (2015). Synthesis of integrated enterprise automated management systems, *Visnyk Natsionalnoho universytetu «Lvivska politekhnika». Seriia: Informatsiini systemy ta merezhi.* 829, 246-254. **11. Kazman, R., Klein, M., Barbacci, M., Longstaff, T., Lipson, H., Carriere, J.** The Architecture Tradeoff Analysis Method, [online] Available at: https://insights.sei.cmu.edu/documents/1186/1998_005_001_16646.pdf [Accessed: 08 September 2025]. **12. Mary, S.,** (1996). Software architecture: Perspectives on an emerging discipline. Upper Saddle River, N.J : Prentice Hall. DOI: 10.5555/231003. **13. Fowler, M.** Microservices a definition of this new architectural term, [online] Available at: https://martinfowler.com/articles/microservices.html [Accessed: 08 September 2025]. **14. Newman, S.** (2015). *Building Microservices.* O'Reilly Media, Incorporated. DOI: 10.5555/2904388. **15. Korniienko, I. V, Korniienko, S. P, Shevaha, V. V, Rudenko, O. V., Zhyrna, O. V.,** (2021). Functional simulation and data flow analysis of the Armament and Military equipment testing system. *Zbirnyk naukovykh prats Derzhavnoho naukovo-doslidnoho instytutu vyprobuvan i sertyfikatsii ozbroiennia ta viiskovoi tekhniky,* 10(4), 83-93. DOI: 10.37701/dndivsovt.10.2021.09. **16. Korniienko, I. V, Korniienko, S. P., Kaznachei, C. M., Rudenko, O. V., Tolmachov, V. Yu.,** (2022). Generalized structure of the information and reference database of Armament and Military equipment of the information system of testing support. *Zbirnyk naukovykh prats Derzhavnoho naukovo-doslidnoho instytutu vyprobuvan i sertyfikatsii ozbroiennia ta viiskovoi tekhniky,* 11(1), 66-77. DOI: 10.37701/dndivsovt.11.2022.08. **17. Skyba, O. V., Domanov, I. O., Rybachok, D. V., Skyba, A. O.,** (2023). Approaches to increase the efficiency of training and testing by creating a uniform information and reference system and automating the activities of the institute's scientists. *Zbirnyk naukovykh prats DNDI VS OVT,* 4(18), 104-111. DOI: 10.37701/dndivsovt.18.2023.15. **18. Evlanov, M .V.,** (2015). Development of methods for forming a representation of the formulated requirement for an information system at the knowledge level. *Vostochno-evropeiskyi zhurnal peredovykh tekhnolohyi,* 4/3 (76), 4-11. [online]. Available at: http://nbuv.gov.ua/UJRN/Vejpte_2015_4(3)__2. [Accessed: 16 September 2025]. **19. Levykyn, V. M., Evlanov, M. V.,** (2006). Metamodel of the functional structure of an information system. *Novi tekhnolohii,* 1(12), 67-72. [online]. Available at: http://openarchive.nure.ua/handle/document/3958 [Accessed: 16 September 2025].

Рукопис надійшов до редакції *29.09.2025*
Рукопис прийнято до друку після рецензування *12.11.2025*
Дата публікації *30.12.2025*