

БАКАЛАВРСЬКА РОБОТА

БР. ІІ - 62.00.00.000 ІІЗ

Група ІІ-22-3

Гаргат Роман

2026

Івано-Франківський національний технічний університет нафти і газу

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Гаргат Роман Ярославович

(прізвище, ім'я, по батькові)

УДК 004.4
(індекс)

БАКАЛАВРСЬКА РОБОТА

Моделі та методи формування користувацько-орієнтованих композицій

сервісів

(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Здобувач освітнього рівня Гаргат Р.Я.
(підпис, ініціали та прізвище здобувача)

Науковий керівник Крихівський Михайло Васильович, к.т.н., доцент
(підпис, прізвище, ім'я, по батькові, науковий ступінь, вчене звання керівника)

Допущено до захисту
Завідувач кафедри

доц. Бандура В.В.
(посада) (підпис) (дата) (ініціали та прізвище)

Івано-Франківськ – 2026

Івано-Франківський національний технічний університет нафти і газу
Інститут, факультет інформаційних технологій
Кафедра інженерії програмного забезпечення
Ступінь вищої освіти бакалавр
Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедрою ІІЗ

доц. В.В. Бандура

“ ” 2026 р.

ЗАВДАННЯ

НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТОВІ

Гаргату Роману Ярославовичу

(прізвище, ім'я, по-батькові)

1. Тема проекту (роботи) “ Моделі та методи формування користувацько-орієнтованих композицій сервісів”

керівник проекту (роботи) Крихівський Михайло Васильович, к.т.н., доцент

затверджені наказом закладу вищої освіти від “ 21 ” квітня 2026р. № 179/7

2. Строк подання студентом проекту (роботи) 10 червня 2026 р.

3. Вихідні дані до проекту (роботи) Результати і матеріали отримані під час проходження переддипломної практики

4. Зміст розрахунково - пояснювальної записки (перелік питань, які потрібно розробити)

1. Аналіз предметної області формування користувацько-орієнтованих композицій сервісів

2. Аналіз прикладних доменів та детермінація профілів у системах композиції сервісів

3. Архітектурна реалізація прототипу системи користувацько-орієнтованої композиції

4. Функціональні вимоги до ядра системи та механізми управління контекстом

5. Програмна реалізація прототипу системи композиції сервісів

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

1. Критерії оцінки доменів для композиції сервісів (рис. 2.1, ст. 28)

2. Характеристика профілю користувача згідно трикомпонентної моделі знань (рис. 2.2, ст. 49)

3. Класифікація зовнішніх умов, що впливають на створення композиції сервісів (рис. 2.3, ст. 32)

4. Сценарій практичної реалізації автоматизація фінансової звітності (рис. 2.4, ст. 36)

5. Архітектура екосистеми розумного будинку у вигляді атомарних сервісів (рис. 2.5, ст.37)

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 21 квітня 2026 р.

Керівник

(підпис)

Завдання прийняв до виконання

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту	Примітка
1	Аналіз предметної області формування користувацько-орієнтованих композицій сервісів	02.05.2026	виконано
2	Аналіз прикладних доменів та детермінація профілів у системах композиції сервісів	15.05.2026	виконано
3	Архітектурна реалізація прототипу системи користувацько-орієнтованої композиції	21.05.2026	виконано
4	Функціональні вимоги до ядра системи та механізми управління контекстом	28.05.2026	виконано
5	Програмна реалізація прототипу системи композиції сервісів	03.06.2026	виконано
6	Оформлення пояснювальної записки бакалаврської роботи завідувачем кафедри	10.06.2026	виконано

Студент – дипломник

Гаргат Р.Я.

(підпис)

Керівник роботи

Крихівський М.В.

(підпис)

АНОТАЦІЯ

Бакалаврська робота містить 83 сторінки, 23 рисунки, список використаних джерел із 34 найменуваннями.

Мета роботи: розробка моделей, методів та програмного прототипу формування користувацько-орієнтованих композицій сервісів на основі семантичного моделювання та адаптивних механізмів взаємодії.

Об'єкт дослідження - процеси композиції сервісів у сервісно-орієнтованих інформаційних системах.

Предмет дослідження - моделі, методи та засоби формування користувацько-орієнтованих композицій сервісів із урахуванням контексту та характеристик користувача.

В першому розділі визначено теоретичні основи та ключові підходи до користувацько-орієнтованої композиції сервісів.

В другому розділі проаналізовано прикладні домени та встановлено вплив контексту і профілів користувачів на процес композиції.

В третьому розділі розроблено архітектурну модель системи та визначено механізми семантичної інтеграції сервісів.

В четвертому розділі сформульовано вимоги до системи та запропоновано механізми управління контекстом і сесіями.

В п'ятому розділі реалізовано програмний прототип системи та підтверджено ефективність запропонованих рішень.

Висновок: у роботі розроблено та обґрунтовано ефективні моделі і методи формування користувацько-орієнтованих композицій сервісів, що забезпечують адаптивність і гнучкість взаємодії користувача із сервісами.

КЛЮЧОВІ СЛОВА: КОМПОЗИЦІЯ СЕРВІСІВ, СЕРВІСНО-ОРІЄНТОВАНА АРХІТЕКТУРА, КОРИСТУВАЦЬКО-ОРІЄНТОВАНИЙ ПІДХІД, СЕМАНТИЧНЕ МОДЕЛЮВАННЯ, КОНТЕКСТ, АДАПТИВНІ ІНТЕРФЕЙСИ, ІНТЕГРАЦІЯ СЕРВІСІВ, ЕЛЕКТРОННА КОМЕРЦІЯ.

ANNOTATION

The bachelor's thesis contains 83 pages, 23 figures, a list of used sources with 34 names.

Purpose of the work: development of models, methods and software prototype for the formation of user-oriented compositional services based on the interaction of semantic modeling and adaptive mechanisms.

The object of research is the processes of service composition in service-oriented information systems.

The subject of research is models, methods and tools for the formation of user-oriented compositional services taking into account the context and user characteristics.

The first section defines the theoretical foundations and key approaches to user-oriented composition of services.

The second section analyzes application domains and establishes the influence of context and user profiles on the composition process.

In the third section, an architectural model of the system is developed and mechanisms for semantic integration of services are defined.

In the fourth section, requirements for the system are formulated and mechanisms for context and session management are proposed.

In the fifth section, a software prototype of the system is implemented and the effectiveness of the proposed solutions is confirmed.

Conclusion: the work has developed and substantiated effective models and methods for forming user-oriented compositional services that ensure adaptability and flexibility of user interaction with services.

KEYWORDS: SERVICE COMPOSITION, SERVICE-ORIENTED ARCHITECTURE, USER-ORIENTED APPROACH, SEMANTIC MODELING, CONTEXT, ADAPTIVE INTERFACES, SERVICE INTEGRATION, ELECTRONIC COMMERCE.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	10
ВСТУП.....	11

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ФОРМУВАННЯ КОРИСТУВАЦЬКО-ОРІЄНТОВАНИХ КОМПОЗИЦІЙ СЕРВІСІВ..... 14

1.1. Концептуальні засади та інструментальні засоби користувацько-орієнтованої композиції сервісів.....	14
1.2. Аналіз тенденцій розвитку веб-сервісів та зміна парадигм їх проектування для кінцевих користувачів.....	17
1.2.1. Концепція функціональної декомпозиції інтерфейсів	17
1.2.2. Необхідність та переваги композиції сервісів.....	18
1.3. Технології машапів у контексті сервісно-орієнтованих архітектур.....	19
1.3.1. Обмеження детермінованої композиції	19
1.3.2. Еволюція до корпоративних машапів та спільного виробництва...	21
1.4. Типологія користувачів та інтелектуальні механізми взаємодії в адаптивних системах композиції сервісів	23
1.4.1. Механізми трансляції вимог та інтелектуальні інтерфейси	23
1.4.2. Аналіз компетенцій та таксономія користувачів	24
1.5 Висновки по розділу.....	26

РОЗДІЛ 2. АНАЛІЗ ПРИКЛАДНИХ ДОМЕНІВ ТА ДЕТЕРМІНАЦІЯ ПРОФІЛІВ КОРИСТУВАЧІВ У СИСТЕМАХ КОМПОЗИЦІЇ СЕРВІСІВ 27

2.1. Критерії та методологія оцінювання доменів.....	27
2.2. Технологічне середовище та прагматика людино-машинної взаємодії в процесах композиції сервісів.....	30

					БР.ІП – 62.00.00.000 ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата	Моделі та методи формування користувацько-орієнтованих композицій сервісів Пояснювальна записка	Літ.	Арк.	Акрушіє	
Розроб.		Гаргат Р.Я.							
Перевір.		Крихівський М.					6		
Реценз.		Григорчук Л.І.				ІФНТУНГ ІП-22-3			
Н. Контр.		Піх М.М.							
Затверд.		Бандура В.В.							

2.2.1. Вимоги до сервісної інфраструктури	31
2.2.2. Ергономічні аспекти та обмеження людино-машинної взаємодії ..	31
2.2.3. Контекстуальні детермінанти процесу композиції.....	31
2.3. Специфіка реалізації сервісно-орієнтованих архітектур у корпоративному сегменті.....	33
2.3.1. Типологія користувачів та рольові моделі	34
2.3.1. Класифікація сервісів та бар'єри інтеграції	34
2.3.3. Умови композиції та людино-комп'ютерна взаємодія	35
2.3.4. Сценарій практичної реалізації автоматизація фінансової звітності	35
2.4. Специфіка та виклики композиції сервісів у доменах домашньої автоматизації	36
2.4.1. Архітектура сервісів та проблема стандартизації.....	37
2.4.2. Аналіз сценарію домашньої автоматизації як прикладу контекстно- залежної композиції	38
2.5. Електронна комерція як динамічний домен для персоналізованої сервісної інтеграції	40
2.5.1. Поточний стан та еволюція сервісної архітектури	40
2.5.2. Технологічні обмеження та відкриті інтерфейси.....	41
2.5.3. Практичний кейс композиції сервісів в електронній комерції.....	42
2.6. e-Health як середовище для впровадження сервіс-орієнтованих систем.....	44
2.6.1. Технологічна конвергенція e-Health та систем автоматизації.....	44
2.6.2. Стандартизація та складність сервісних ланцюжків	45
2.6.3. Кейс: Дистанційний моніторинг критичних станів здоров'я	45
2.7. Порівняльний аналіз прикладних доменів за критеріями користувацько-орієнтованої композиції	47
2.8 Висновки по розділу	49

РОЗДІЛ 3. АРХІТЕКТУРНА РЕАЛІЗАЦІЯ ПРОТОТИПУ СИСТЕМИ КОРИСТУВАЦЬКО-ОРІЄНТОВАНОЇ КОМПОЗИЦІЇ 50

3.1. Представлення архітектури клієнтського додатку.....	50
3.1.1. Функціональні характеристики сервісного ядра композиції.....	51
3.1.2. Семантичне моделювання та репозиторій сервісів	51
3.2. Механізми семантичної трансформації та архітектурні особливості клієнтського рівня.....	52
3.2.1. Адаптивна генерація вхідних інтерфейсів на основі семантичних типів	53
3.2.2. Виконання сервісів та міжпротокольна медіація.....	54
3.3.1. Порівняльний аналіз методів композиції	56
3.3.2. Управління контекстом виконання та станом сесії	59
3.4 Висновки по розділу	59

РОЗДІЛ 4. ФУНКЦІОНАЛЬНІ ВИМОГИ ДО ЯДРА СИСТЕМИ ТА МЕХАНІЗМИ УПРАВЛІННЯ КОНТЕКСТОМ 60

4.1. Концептуалізація контексту в системах композиції сервісів	60
4.1.1. Таксономія контекстної інформації	61
4.1.2. Стратегії актуалізації контексту в архітектурі «Клієнт–Сервер» ...	62
4.1.3. Механізми управління сесіями та персистентність даних.....	63
4.2. Специфікація протоколів взаємодії між інтерфейсним рівнем та середовищем оркестрації	63
4.2.1. Процедура вирішення залежностей.....	64
4.2.2. Верифікація вхідних параметрів.....	66
4.3. Моделювання часової послідовності та динаміки повідомлень у процесах композиції	67
4.3.1. Реалізація сценарію лінійного пошуку	67
4.3.2. Сценарій ітераційної віртуальної композиції.....	68
4.4 Висновки до розділу	69

РОЗДІЛ 5. ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОТОТИПУ СИСТЕМИ	
КОМПОЗИЦІЇ СЕРВІСІВ.....	71
5.1. Реалізація моделі користувацько-орієнтованої композиції на прикладі прототипу «RestaurantFinder».....	71
5.2. Архітектурний аналіз інтерфейсу в контексті інтегрованих сервісів та динамічної фільтрації.....	73
5.3 Висновки по розділу.....	77
ВИСНОВКИ	78
ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	80
БІБЛІОГРАФІЧНА ДОВІДКА	

					БР.ІП – 62.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

BAN — Body Area Network

DRY — Don't Repeat Yourself

FHIR — Fast Healthcare Interoperability Resources

GDPR — General Data Protection Regulation

HIPAA — Health Insurance Portability and Accountability Act

HL7 — Health Level Seven

LBS — Location Based Service

OWL-S — Semantic Markup for Web Services (Ontology Web Language for Services)

REST — Representational State Transfer

SaaS — Software as a Service

SOAP — Simple Object Access Protocol

SPATEL — Service Provisioning and Targeted Electronic Learning (Semantic Service Modeling Language)

					БР.ІП – 62.00.00.000 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Сучасний етап розвитку інформаційних технологій характеризується стрімким зростанням кількості веб-сервісів, підвищенням їх функціональної складності та необхідністю інтеграції в єдині інформаційні середовища. У цих умовах особливого значення набувають підходи, що забезпечують ефективну композицію сервісів із урахуванням потреб кінцевого користувача. Традиційні методи побудови сервісних систем, орієнтовані на жорстко визначені сценарії, виявляються недостатньо гнучкими в умовах динамічного середовища та різноманітності користувацьких запитів. Це обумовлює необхідність розробки нових моделей і методів, які забезпечують адаптивність, персоналізацію та контекстну чутливість процесів композиції.

Важливим напрямом досліджень є інтеграція семантичних технологій, інтелектуальних інтерфейсів та механізмів управління контекстом, що дозволяють автоматизувати процес формування композицій сервісів. У цьому контексті особливу роль відіграє визначення профілів користувачів, їхніх компетенцій і поведінкових характеристик, що дає змогу підвищити ефективність взаємодії з системою. Крім того, актуальним є дослідження прикладних доменів, у яких реалізуються сервісно-орієнтовані архітектури, з метою виявлення специфічних вимог та обмежень.

Таким чином, дана робота присвячена розробці моделей та методів формування користувацько-орієнтованих композицій сервісів, що враховують контекст виконання, особливості користувача та специфіку прикладних середовищ.

Актуальність роботи

Актуальність теми зумовлена зростаючою потребою у створенні гнучких, масштабованих та інтелектуалізованих інформаційних систем, здатних ефективно інтегрувати різноманітні сервіси. У сучасних умовах цифровізації економіки та суспільства спостерігається значне збільшення

					БР.ІП – 62.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

кількості доступних сервісів, що ускладнює їх використання без належних механізмів автоматизації композиції.

Існуючі підходи до композиції сервісів часто не враховують індивідуальні потреби користувачів, їхній контекст та рівень компетенцій, що призводить до зниження ефективності взаємодії та підвищення складності використання систем. Крім того, недостатній рівень стандартизації та семантичної узгодженості сервісів створює додаткові бар'єри для їх інтеграції.

Особливої актуальності набуває дослідження користувацько-орієнтованих підходів у таких доменах, як електронна комерція, e-Health, корпоративні інформаційні системи та системи домашньої автоматизації, де важливо забезпечити адаптивність і персоналізацію сервісів. Враховуючи зазначене, розробка моделей і методів, що дозволяють формувати композиції сервісів з урахуванням контексту та характеристик користувача, є важливою науково-прикладною задачею.

Метою роботи є розробка моделей, методів та програмного прототипу формування користувацько-орієнтованих композицій сервісів на основі семантичного моделювання та адаптивних механізмів взаємодії.

Об'єкт дослідження - процеси композиції сервісів у сервісно-орієнтованих інформаційних системах.

Предмет дослідження - моделі, методи та засоби формування користувацько-орієнтованих композицій сервісів із урахуванням контексту та характеристик користувача.

Завдання дослідження:

- Провести аналіз предметної області користувацько-орієнтованої композиції сервісів.
- Дослідити прикладні домени та визначити профілі користувачів.
- Розробити архітектурну модель системи композиції сервісів.

					БР.ІП – 62.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

- Сформулювати функціональні вимоги та механізми управління контекстом.

- Реалізувати програмний прототип системи та оцінити його ефективність.

Методи дослідження

У роботі використано методи системного аналізу, семантичного моделювання, об'єктно-орієнтованого проєктування, методи аналізу та синтезу інформаційних систем, а також методи порівняльного аналізу та моделювання процесів композиції сервісів.

Наукова новизна

Наукова новизна отриманих результатів полягає у розробці підходів до формування користувацько-орієнтованих композицій сервісів, що базуються на інтеграції семантичного моделювання, адаптивних інтерфейсів та механізмів управління контекстом. Удосконалено методи трансляції користувацьких вимог у формалізовані запити до системи. Запропоновано механізми адаптивної генерації інтерфейсів на основі семантичних типів. Розроблено підходи до управління сесіями та контекстом у динамічних середовищах композиції сервісів.

Практичне застосування

Практичне значення роботи полягає у можливості використання розроблених моделей і методів для створення інформаційних систем, що забезпечують гнучку інтеграцію сервісів та персоналізовану взаємодію з користувачем. Запропонований прототип може бути застосований у корпоративних системах, електронній комерції, системах домашньої автоматизації та e-Health. Отримані результати сприяють підвищенню ефективності використання сервісів та зниженню складності їх інтеграції.

Бакалаврська робота містить 83 сторінки, 23 рисунки, 5 розділів, список використаних джерел із 34 найменуваннями.

					БР.ІП – 62.00.00.000 ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ФОРМУВАННЯ КОРИСТУВАЦЬКО-ОРІЄНТОВАНИХ КОМПОЗИЦІЙ СЕРВІСІВ

1.1. Концептуальні засади та інструментальні засоби користувацько-орієнтованої композиції сервісів

Композиція сервісів у сучасних програмних архітектурах розглядається як фундаментальна парадигма, що забезпечує інтеграцію атомарних програмних компонентів у цілісні системи для формування сервісів із доданою вартістю (Value-Added Services). В умовах глобальної цифровізації та інтеграції веб-технологій у повсякденну діяльність виникає критична потреба у забезпеченні доступу широкого кола користувачів до персоналізованих сервісних рішень. Основною метою композиції є задоволення складних функціональних вимог кінцевого споживача шляхом синергетичного поєднання можливостей гетерогенних сервісів.

Історично успішним прикладом реалізації даного підходу є машапи (mashups). Проте класична модель розробки машапів передбачає провідну роль кваліфікованого спеціаліста (розробника) у процесі створення та конфігурації композиції, що створює бар'єр для оперативного вирішення специфічних задач кінцевими користувачами.

Користувацько-орієнтована композиція сервісів (User-Centric Service Composition) спрямована на зміну цієї парадигми через надання інструментарію, який дозволяє абстрагуватися від складності програмної реалізації. Даний підхід делегує процес створення композицій безпосередньо кінцевим користувачам, які не мають глибокої технічної експертизи, проте володіють предметними знаннями для формування функціональних вимог.

У межах даної роботи представлено розробку прототипу системи користувацько-орієнтованої композиції, функціонування якого базується на використанні фреймворку DynamiCoS. Вибір даного фреймворку

					БР.ІП – 62.00.00.000 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

обумовлений його здатністю до динамічного виявлення та оркестрації сервісів у мінливих контекстних умовах. Фреймворк DynamiCoS (Dynamic Composition of Services) — це концептуальна модель у сфері сервіс-орієнтованої архітектури (SOA), яка фокусується на автоматизації процесу створення складних сервісів шляхом об'єднання існуючих компонентів у режимі реального часу.

Фреймворк базується на ідеї, що сучасні системи занадто динамічні для статичного програмування. Його робота зазвичай розбивається на кілька етапів:

1. Семантичний опис

Щоб система розуміла, що робить сервіс, використовується семантична анотація (наприклад, OWL-S або WSMO). Це дозволяє фреймворку знати не тільки технічні параметри (вхідні/вихідні дані), а й реальне значення сервісу (наприклад, "бронювання готелю").

2. Динамічний пошук

Система шукає доступні сервіси в реєстрах або мережі, які відповідають конкретним вимогам користувача в поточний момент часу.

3. Автоматична композиція

Це основа фреймворку. Використовуючи алгоритми планування (AI Planning), DynamiCoS будує ланцюжок сервісів.

- Вхід: Поточний стан та бажана мета.

- Процес: Пошук послідовності сервісів, де вихідні дані одного є вхідними для іншого.

- Вихід: Складений бізнес-процес.

4. Виконання та адаптація

Фреймворк не просто запускає процес, а й стежить за ним. Якщо один із сервісів перестає відповідати, DynamiCoS може "на льоту" знайти заміну або перебудувати частину ланцюжка без зупинки всієї системи.

					БР.ІП – 62.00.00.000 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 1.1 – Етапи роботи фреймворку

Як цільові домени для апробації прототипу обрано сфери електронної комерції (e-commerce) та індустрії розваг. Обґрунтування вибору цих галузей базується на наступних чинниках:

- Висока варіативність сценаріїв: необхідність комбінування платіжних систем, логістичних сервісів та медіа-контенту.
- Специфіка цільової аудиторії: переважання користувачів без технічної підготовки, що потребують інтуїтивно зрозумілих інтерфейсів взаємодії.

Архітектурне рішення розробленого прототипу базується на впровадженні специфічних шаблонів взаємодії (interaction patterns) з фреймворком DynamiCoS, що забезпечує семантичну відповідність між запитам користувача та наявними веб-сервісами. Важливим етапом дослідження є емпірична оцінка ефективності системи. У ході тестування було проаналізовано:

- Когнітивне навантаження при створенні композицій;
- Відповідність результуючої функціональності поставленим завданням;
- Юзабіліті інтерфейсу конструювання логіки сервісів.

Результати оцінювання підтверджують життєздатність запропонованого підходу та демонструють можливість ефективного створення складних сервісних ланцюжків користувачами з мінімальним рівнем технічної підготовки.

1.2. Аналіз тенденцій розвитку веб-сервісів та зміна парадигм їх проектування для кінцевих користувачів

Протягом останнього десятиліття спостерігається інтенсивна інтеграція інтернет-сервісів у повсякденну діяльність суб'єктів, що зумовлює формування концепції первазивних обчислень (pervasive computing). Екосистема мережевих послуг демонструє експоненціальне зростання, що висуває перед програмною інженерією нові виклики щодо забезпечення інтероперабельності та адаптивності систем. Сучасні веб-додатки трансформувалися з інструментів суто професійного призначення у засоби підтримки рутинних завдань широкого кола споживачів.

1.2.1. Концепція функціональної декомпозиції інтерфейсів

Ключовим аспектом спрощення доступу до веб-сервісів для кінцевого користувача є стратегічне зниження когнітивного навантаження через мінімізацію складності інтерфейсу користувача. На відміну від традиційного монолітного програмного забезпечення, яке тяжіє до акумуляції надлишкової функціональності в межах єдиної системи (що часто призводить до перевантаження меню та опцій), сучасний підхід базується на принципі атомарності завдань.

Цей підхід передбачає фокусування сервісу на вирішенні однієї конкретної проблеми, що корелює з філософією проектування UNIX («роби щось одне, але роби це добре»). У таблиці 1 наведено порівняльний аналіз традиційного та сервіс-орієнтованого підходів до проектування ПЗ.

					БР.ІП – 62.00.00.000 ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 1.1 - Порівняння архітектурних підходів до розробки ПЗ

Характеристика	Традиційне ПЗ (Monolithic)	Сучасні веб-сервіси (Micro-task oriented)
Функціональний обсяг	Максимальне охоплення вимог в одній системі	Спеціалізація на конкретній задачі
Складність UI	Висока (багаторівневі меню, багато опцій)	Низька (мінімалізм, фокус на дії)
Приклади	Локальні ERP-системи, важкі IDE	GMail, Basecamp, FreshBooks
Гнучкість	Жорстка структура	Висока здатність до інтеграції

Сфера особистого цифрового способу життя стає дедалі більш залежною від хмарної інфраструктури. Зокрема, у сегменті електронної комерції (e-commerce) процес прийняття рішення про покупку декомпонується на низку етапів, кожен з яких обслуговується окремим спеціалізованим сервісом:

- Аналітичний етап: агрегація цін.
- Інформаційний етап: верифікація характеристик та огляди.
- Логістичний етап: геопросторова локалізація точок продажу (наприклад, Google Maps).

Аналогічні тенденції простежуються у галузях e-Health (електронна охорона здоров'я) та цифрових розваг, де сервіси мають адаптуватися до індивідуальних контекстних потреб користувача.

1.2.2. Необхідність та переваги композиції сервісів

Внаслідок функціональної вузькоспеціалізації окремих додатків виникає потреба в їхньому синергетичному об'єднанні. Композиція сервісів виступає як ефективний механізм вирішення комплексних завдань користувача шляхом інтеграції гетерогенних програмних ресурсів.

Розглянемо формальну модель композиції на прикладі бізнес-процесу оплати рахунку:

					БР.ІП – 62.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

$$S_{total} = f(S_{CRM} + S_{PM} + S_{Billing})$$

де:

S_{CRM} — сервіс управління взаємовідносинами з клієнтами (дані контрагента);

S_{PM} — сервіс управління проектами (облік витраченого часу);

$S_{Billing}$ — сервіс проведення транзакцій.

Для кінцевого споживача композиція дозволяє будувати персоналізовані сценарії взаємодії (User Journeys). Наприклад, інтеграція сервісу відгуків з картографічним сервісом дозволяє не просто знайти товар, а й верифікувати репутацію продавця з урахуванням географічної близькості. Таким чином, композиція сервісів є фундаментом для створення персоналізованого цифрового середовища, де програмне забезпечення підлаштовується під специфічні цілі суб'єкта, а не навпаки.

1.3. Технології машапів у контексті сервісно-орієнтованих архітектур

Фундаментом для розробки сучасних розподілених систем є сервісно-орієнтована архітектура (Service-Oriented Architecture, SOA). Дана парадигма забезпечує методологічну базу для проектування, розгортання та повторного використання програмних компонентів у формі автономних сервісів. Одним із найбільш поширених прикладів практичної реалізації SOA на рівні користувацького представлення є машапи (mashups) — веб-додатки, що інтегрують дані або функціональність з декількох сторонніх джерел для створення нового цілісного сервісу.

1.3.1. Обмеження детермінованої композиції

Попри свою популярність, традиційна модель створення машапів характеризується централізованим підходом: розробник (програмний

					БР.ІП – 62.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

інженер) виступає посередником, який інтерпретує вимоги кінцевих користувачів і реалізує жорстко задану логіку композиції. Такий підхід має суттєвий недолік — інформаційну асиметрію між розробником і споживачем, що часто призводить до неповного задоволення специфічних запитів останнього.

Більше того, у середовищі веб-сервісів спостерігається прояв концепції «довгого хвоста» (The Long Tail). У той час як стандартні рішення задовольняють масові запити («голова» розподілу), існує практично необмежена кількість унікальних, нішевих потреб користувачів, що формують «довгий хвіст».

Математично кількість потенційних комбінацій сервісів C при кількості доступних компонентів n зростає експоненціально:

$$C = \sum_{k=2}^n \frac{n!}{k!(n-k)!}$$

Це робить неможливим створення префіксованих (заздалегідь розроблених) машапів для кожного індивідуального сценарію.

Класичним прикладом машапу є ресурс housingmaps.com, який синтезує геопросторові дані Google Maps із оголошеннями платформи Craigslist. Система забезпечує візуалізацію об'єктів нерухомості на карті, що є релевантним для базового пошуку. Однак ефективність такого рішення різко нівелюється при виникненні аддитивних контекстних обмежень. Наприклад:

- Необхідність врахування транспортної доступності (наприклад, радіус 500 метрів від станцій метро);
- Кореляція з даними про криміногенну обстановку або екологічні показники району.

					БР.ІП – 62.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

Для задоволення таких вимог стандартний машап потребує ручного доопрацювання коду розробником, що є економічно недоцільним для одиничних запитів.

1.3.2. Еволюція до корпоративних машапів та спільного виробництва

Часткове розв'язання цієї проблеми запропонували корпоративні машапи (Enterprise Mashups). Вони базуються на принципах спільного виробництва (co-production) та активної співпраці між ІТ-відділом і бізнес-користувачами. У цій моделі бізнес-аналітики отримують інструменти для самостійного конструювання сервісних ланцюжків.

Проте, незважаючи на успіх у корпоративному секторі, такі інструменти залишаються складними для пересічного користувача. Поточний стан індустрії вимагає переходу до систем, які:

1. Абстрагують технічну складність: приховують деталі API та протоколів передачі даних.
2. Забезпечують демократизацію розробки: надають можливість композиції сервісів користувачам без навичок програмування.
3. Підтримують динамічність: дозволяють миттєво змінювати логіку композиції відповідно до нових вимог у «довгому хвості». «Довгий хвіст» - це ситуація, коли існує невелика кількість дуже популярних запитів («голова») і величезна кількість унікальних, нішевих запитів («хвіст»), сукупна значущість яких може перевищувати популярні запити.

Традиційний підхід до розробки ПЗ не може впоратися з «довгим хвостом» через декілька причин:

1. Професійна розробка окремого додатка для кожного нішевого запиту коштує занадто дорого.
2. Кількість способів поєднати різні сервіси (карти, оплата, погода, доставка) зростає експоненціально. Неможливо заздалегідь передбачити всі сценарії.

					БР.ІП – 62.00.00.000 ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

3. Потреби в «хвості» часто виникають «тут і зараз» і так само швидко зникають.



Рисунок 1.2 – Перелік причин неможливості традиційного ПЗ подолати проблему “довгого хвоста”

Для цього є рішення таке як користувачко-орієнтована композиція. Замість того, щоб чекати, поки програміст напише код для конкретної ситуації, система дає інструменти самому користувачу.

- У «голові» ми використовуємо готові додатки (Uber, Amazon, Booking).
- У «довгому хвості» ми використовуємо композицію сервісів, щоб зібрати унікальне рішення з частин (атомарних сервісів) під свій конкретний запит.

Математично розподіл «довгого хвоста» часто описується степеневим законом:

$$P(x) \approx x^{-\alpha}$$

де $P(x)$ — ймовірність виникнення певної вимоги x . Це означає, що хоча ймовірність специфічного запиту низька, загальна кількість таких запитів (площа під кривою «хвоста») є колосальною.

Таким чином, актуальним завданням є розробка архітектурних рішень, що виходять за межі корпоративного сегменту та забезпечують персоналізовану композицію сервісів для широкого кола гетерогенних користувачів.

1.4. Типологія користувачів та інтелектуальні механізми взаємодії в адаптивних системах композиції сервісів

Сучасні корпоративні машапи, попри свою функціональну потужність, зберігають високий поріг входження, залишаючись інструментарієм переважно для бізнес-експертів та ІТ-фахівців. Для подолання цього бар'єру та забезпечення інклюзивності сервіс-орієнтованих систем необхідна розробка спрощених підходів, що базуються на принципах користувацько-орієнтованого проектування (User-Centric Design).

1.4.1. Механізми трансляції вимог та інтелектуальні інтерфейси

Ефективна композиція сервісів у гетерогенному середовищі вимагає реалізації двох ключових функціональних аспектів:

1. Когнітивна інтерпретація вимог. Використання методів обробки природної мови (NLP) та представлення знань (Knowledge Representation) для трансформації неформалізованих запитів користувача у машиночитані специфікації. Це дозволяє нівелювати розрив між семантикою природної мови та формальними протоколами виконання.

					БР.ІП – 62.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

2. Адаптивна взаємодія. Створення інтерфейсів, що динамічно підлаштовуються під рівень компетенцій користувача, забезпечуючи необхідний рівень абстракції на кожному етапі конструювання сервісу.

1.4.2. Аналіз компетенцій та таксономія користувачів

Згідно з дослідженнями [10], релевантність створеної композиції сервісів безпосередньо корелює з двома векторами компетенцій користувача: технічними знаннями та знаннями предметної області (домену).

Технічні знання визначають здатність суб'єкта взаємодіяти з низькорівневими мовами опису бізнес-процесів, такими як BPEL (Business Process Execution Language).

Знання домену включають розуміння логіки процесів у конкретній галузі та обізнаність щодо доступних у ній атомарних сервісів.

На основі поєднання цих факторів виділяється чотирирівнева модель профілів користувачів (таблиця 1.2).

Таблиця 1.2 - Матриця профілів користувачів систем композиції сервісів

Тип користувача	Технічні знання	Знання домену	Потреби у підтримці системи
Початківець	Низькі	Низькі	Повна автоматизація та покрокове керівництво
Експерт домену	Низькі	Високі	Гнучкість у логіці за високого рівня абстракції ПЗ
Технічний експерт	Високі	Низькі	Інструменти валідації та технічного налагодження
Просунутий (Power User)	Високі	Високі	Прямий доступ до коду та складних конфігурацій

Для вирішення проблеми інтеграції користувачів із різним рівнем підготовки використовується відповідний фреймворк [9, 11]. Концептуальна мета фреймворку полягає у реалізації механізму «захисту» користувача від технологічної складності процесу композиції.

Функціонування фреймворку базується на принципі керованого синтезу, де система виступає інтелектуальним агентом-помічником. Процес композиції формалізується як ітеративна процедура уточнення намірів користувача:

- Семантичний пошук компонентів на основі онтологій домену.
- Автоматична перевірка сумісності інтерфейсів сервісів.
- Візуалізація шаблонів взаємодії, адаптованих до обраного профілю користувача.

Повна реалізація користувацько-орієнтованої парадигми вимагає глибинної інтеграції знань про домени та шаблони поведінки. Надання користувачам повноважень щодо самостійного створення сервісів незалежно від їхньої технічної підготовки є ключовим фактором трансформації Інтернету сервісів у персоналізоване цифрове середовище. Це не лише знижує витрати на розробку спеціалізованого ПЗ, а й забезпечує безпрецедентний рівень гнучкості цифрових послуг у щоденному житті.

1.5 Висновки по розділу

У першому розділі проведено системний аналіз предметної області користувацько-орієнтованої композиції сервісів, що дозволило визначити ключові концептуальні підходи та інструментальні засоби її реалізації. Встановлено, що сучасні тенденції розвитку веб-сервісів характеризуються переходом від жорстко детермінованих моделей до гнучких, адаптивних і контекстно-чутливих підходів.

					БР.ІП – 62.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

Обґрунтовано доцільність використання технологій машапів як засобу інтеграції розподілених сервісів, попри їх обмеження у формалізації процесів композиції.

Визначено роль інтелектуальних інтерфейсів у забезпеченні ефективної трансляції користувацьких вимог у формалізовані запити до системи. Сформовано таксономію користувачів, яка є основою для подальшої адаптації механізмів композиції сервісів.

					БР.ІП – 62.00.00.000 ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. АНАЛІЗ ПРИКЛАДНИХ ДОМЕНІВ ТА ДЕТЕРМІНАЦІЯ ПРОФІЛІВ КОРИСТУВАЧІВ У СИСТЕМАХ КОМПОЗИЦІЇ СЕРВІСІВ

Композиція сервісів є універсальною парадигмою, адаптивність якої дозволяє застосовувати її у широкому спектрі прикладних галузей. Дане дослідження сфокусоване на доменах, де суб'єкти з гетерогенним рівнем компетенцій здійснюють синтез та експлуатацію сервісних ланцюжків, зокрема у режимі реального часу. Для побудови релевантного сценарію взаємодії та верифікації шаблонів поведінки користувачів необхідно провести комплексний аналіз потенційних сфер застосування користувацько-орієнтованого підходу.

2.1. Критерії та методологія оцінювання доменів

У межах даної роботи проведено огляд актуальних доменів, інформація про які є верифікованою та публічно доступною. Пріоритет надавався галузям, що містять репрезентативні сценарії для розробки та тестування експериментального прототипу. Репрезентативність сценарію визначається наявністю диференційованих груп користувачів із різним рівнем технічної та доменної підготовки.

Оцінювання доменів здійснювалося на основі чотирьох детермінованих критеріїв:

- Типологія користувачів: аналіз рольових моделей та ідентифікація когнітивних бар'єрів на основі знань домену та технічної грамотності.
- Функціональний склад сервісів: класифікація доступних атомарних сервісів та оцінка потенціалу їхньої композиційної сумісності.
- Ергономіка людино-машинної взаємодії (HCI): аналіз інтерфейсних рішень та апаратного забезпечення (термінали, мобільні пристрої), що визначають спосіб взаємодії із системою підтримки композиції.

					БР.ІП – 62.00.00.000 ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

- Контекстуальні умови композиції: виявлення зовнішніх обмежень (часовий дефіцит, мобільність, мережева інфраструктура), що впливають на зручність експлуатації системи.



Рисунок 2.1 – Критерії оцінки доменів для композиції сервісів

Системи підтримки композиції сервісів орієнтовані на виконання двох фундаментальних функцій: проектування (конструювання) нових сервісних ланцюжків та їхня безпосередня ітераційна реалізація. Відповідно до цих функцій виділяються дві ключові ролі:

- Композитор (Composer): суб'єкт, який формує архітектуру сервісної логіки шляхом агрегації доступних компонентів.
- Кінцевий користувач (End-user): споживач, який використовує готові результати композиції для вирішення прикладних завдань.

Варто зазначити, що у сучасних користувацько-орієнтованих системах ці ролі часто конвергують, формуючи концепцію «прозюмера» (prosumer) — активного споживача, який самостійно адаптує програмний інструментарій під власні потреби.

Для характеристики профілю користувача використовується трикомпонентна модель знань [10]:

- Знання прикладного домену: Розуміння концептуального апарату та бізнес-логіки конкретної галузі. Ці знання дозволяють чітко сформулювати функціональні вимоги до майбутньої композиції.
- Знання сервісного ландшафту: Усвідомлення переліку та можливостей конкретних сервісів, доступних у даному домені. Це знання є критичним для етапу вибору (selection) компонентів.
- Технічна грамотність: Здатність оперувати інструментами формалізації логіки. На сучасному етапі професійна композиція спирається на стандарти WSDL (опис інтерфейсів) та BPEL (виконання бізнес-процесів).



Рисунок 2.2 – Характеристика профілю користувача згідно трикомпонентної моделі знань

Однак, з метою демократизації технології, виникає потреба у створенні абстрактних візуальних середовищ, які б нівелювали потребу у знанні цих специфікацій для нефахівців.

На основі інтеграції вищезазначених типів знань сформовано таксономію користувачів, представлену в таблиці 2.1.

Таблиця 2.1 - Класифікація профілів користувачів за рівнем компетенцій

Тип користувача	Знання домену	Знання сервісів	Технічні знання	Потрібний рівень підтримки системи
Початківець	Відсутні	Відсутні	Відсутні	Максимальний (майстри, шаблони)
Експерт домену	Наявні	Наявні	Відсутні	Середній (візуальне моделювання)
Технічний експерт	Відсутні	Відсутні	Наявні	Мінімальний (кодування, API)
Просунутий (Power User)	Наявні	Наявні	Наявні	Опціональний (пряме керування)

Дана класифікація підкреслює необхідність розробки адаптивних механізмів підтримки (guidance mechanisms). Зокрема, для «профанів» та «експертів домену» система повинна забезпечувати високий рівень абстракції, автоматично пропонуючи варіанти композиції на основі семантичного аналізу вимог, тоді як для технічних фахівців важливо зберегти можливість низькорівневого налаштування параметрів взаємодії.

2.2. Технологічне середовище та прагматика людино-машинної взаємодії в процесах композиції сервісів

Ефективність функціонування системи підтримки композиції сервісів (СПКС) у будь-якому прикладному домені детермінується не лише наявністю атомарних компонентів, а й якістю їхніх інтерфейсів та контекстуальними умовами взаємодії. Для забезпечення формування сервісів із доданою вартістю (Value-Added Services) кожен компонент повинен відповідати принципам модульності та мати чітко верифікований програмний інтерфейс (API).

2.2.1. Вимоги до сервісної інфраструктури

Для успішної участі у процесах динамічної композиції веб-сервіс має відповідати критеріям інтероперабельності. Це передбачає наявність формалізованих описів, що дозволяють системі автоматично ідентифікувати:

- Функціональні параметри (Signature);
- Перед- та післяумови виконання (Pre- and Post-conditions);
- Якість сервісу (Quality of Service, QoS).

2.2.2. Ергономічні аспекти та обмеження людино-машинної взаємодії

Людино-комп'ютерна взаємодія (Human-Computer Interaction, HCI) у контексті користувацько-орієнтованої композиції розглядається як критичний чинник, що визначає здатність користувача успішно сконструювати сервісний ланцюжок.

Ключовим викликом є адаптація користувацького інтерфейсу (UI) до апаратних обмежень пристроїв. Враховуються наступні аспекти:

- Обмеження екранного простору. На мобільних терміналах складні візуальні графі композиції призводять до надмірного когнітивного навантаження. Це потребує впровадження методів прогресивного розкриття інформації та інтелектуальної фільтрації доступних опцій.
- Модальність взаємодії. Перехід від традиційних маніпуляторів («миша») до сенсорних інтерфейсів або голосового керування вимагає перегляду парадигм конструювання логіки сервісів.

2.2.3. Контекстуальні детермінанти процесу композиції

Процес створення композиції сервісів не є статичним і суттєво залежить від зовнішніх умов, які ми класифікуємо за трьома вимірами:

А. Просторовий та часовий виміри (Spatiotemporal context)

Час і місце ініціації запиту визначають тип композиції:

					БР.ІП – 62.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

- Design-time composition - проектування заздалегідь у стаціонарних умовах. Користувач має достатні часові ресурси для детального налаштування параметрів.

- Runtime (On-the-fly) composition - динамічне створення ланцюжка безпосередньо в момент виникнення потреби (наприклад, під час руху містом). Тут пріоритетом є швидкість прийняття рішень та автоматизація вибору.



Рисунок 2.3 – Класифікація зовнішніх умов, що впливають на створення композиції сервісів

Б. Інформаційне забезпечення та підтримка прийняття рішень

Залежно від профілю користувача (див. розділ 2.1), система повинна надавати відповідний рівень когнітивної підтримки (scaffolding). Це може включати:

- Вбудовані інтерактивні підказки;
- Використання прецедентів (рекомендацій на основі попереднього досвіду інших користувачів);
- Семантичну верифікацію кроків композиції для запобігання логічним помилкам.

В. Стійкість та рекурентність композицій (Persistence and Reusability)

Стійкість композиції визначається її здатністю до повторного використання або адаптації при зміні параметрів середовища. Враховуючи високу вартість зусиль (User Effort) на створення складної композиції в умовах обмежених ресурсів (наприклад, на мобільному пристрої), система повинна забезпечувати:

- Можливість збереження успішних конфігурацій для подальшого виклику.
- Здатність системи автоматично замінювати деградовані сервіси у збереженому ланцюжку без втручання користувача.

Таким чином, ефективна СПКС повинна бути контекстно-залежною (Context-aware), адаптуючи методи представлення інформації та рівень автоматизації композиції до поточного стану користувача, його пристрою та часових обмежень. Оптимізація взаємодії між цими чинниками дозволяє знизити бар'єр входження для нетехнічних фахівців та забезпечити стабільність сервісних рішень.

2.3. Специфіка реалізації сервісно-орієнтованих архітектур у корпоративному сегменті

Корпоративний сектор традиційно виступає ключовим полігоном для впровадження сервіс-орієнтованих архітектур (SOA) та технологій композиції сервісів. Основна прагматика використання SOA у бізнесі полягає у мінімізації операційних витрат та скороченні часу виходу продуктів на ринок (Time-to-Market), що дозволяє організаціям оперативно адаптувати бізнес-процеси до динамічних змін ринкової кон'юнктури.

Історично переваги композиції сервісів були доступні переважно великим корпораціям із власними центрами розробки. Проте сучасна парадигма цифрової трансформації та розвиток концепції Software as a

					БР.ІП – 62.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

Service (SaaS) відкрили цей інструментарій для малого та середнього бізнесу (SME).

Незважаючи на високу доступність, більшість сучасних SaaS-рішень залишаються «функціональними островами», фокусуючись на вузьких завданнях: CRM, управління проектами або білінг. Відтак, виникає гостра потреба у механізмах міжплатформної композиції, які б дозволили підприємцям формувати інтегровані екосистеми з гетерогенних хмарних компонентів.

2.3.1. Типологія користувачів та рольові моделі

У межах даного домену спостерігається чітка кореляція між масштабом підприємства та розподілом ролей у процесі композиції:

- великі підприємства - характеризуються диференціацією ролей. Процес створення композиції закріплений за кваліфікованими ІТ-фахівцями (архітекторами, розробниками), тоді як лінійний персонал виступає виключно як споживач готового сервісу.

- малі підприємства та стартапи - тут спостерігається конвергенція ролей. Власник бізнесу або провідний менеджер часто виступає в ролі «прозюмера», самостійно конструюючи необхідні ланцюжки взаємодії сервісів. Найбільшим викликом у цьому випадку є дефіцит технічних знань (наприклад, розуміння REST API або методів автентифікації), що вимагає від системи підтримки композиції максимально високого рівня абстракції.

2.3.1. Класифікація сервісів та бар'єри інтеграції

Сервісний ландшафт домену поділяється на два вектори:

- Інституційні (приватні) сервіси - внутрішні розробки компаній, що базуються на суворих корпоративних стандартах, що полегшує їхню оркестрацію.

					БР.ІП – 62.00.00.000 ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

- Публічні SaaS-інтерфейси - глобальні сервіси, що пропонують відкриті API. Основною проблемою тут є гармонізація інтерфейсів — невідповідність форматів даних та протоколів взаємодії, що вимагає від СПКС інтелектуальних механізмів медіації (трансформації даних).

2.3.3. Умови композиції та людино-комп'ютерна взаємодія

Корпоративне середовище традиційно є високотехнологічним, що передбачає високий рівень комп'ютерної грамотності користувачів. Проте сучасні тренди вносять корективи:

- Пристрої: спостерігається перехід від стаціонарних робочих станцій до моделі BYOD (Bring Your Own Device). Використання мобільних пристроїв для управління бізнес-процесами накладає вимоги до лаконічності інтерфейсів композиції.

- Темпоральний аспект: у бізнес-доміні домінує design-time composition. Композиції розробляються ретельно, з розрахунком на тривалу експлуатацію та багаторазову повторюваність, що виправдовує початкові часові витрати користувача на етапі конфігурації.

2.3.4. Сценарій практичної реалізації автоматизація фінансової звітності

Розглянемо сценарій інтеграції сервісу управління проектами (наприклад, Jira або Asana) та білінгової системи.

Бізнес-ціль: Автоматичне формування рахунків-фактур на основі закритих завдань у кінці звітного періоду.

Процес композиції:

1. Екстракція даних. Система вилучає атрибут «затрачений час» із сервісу управління проектами для конкретного контрагента.

2. Трансформація. Конвертація одиниць часу у грошовий еквівалент згідно з тарифною сіткою, збереженою у профілі користувача.

					БР.ІП – 62.00.00.000 ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

3. Генерація. Передача сформованого масиву даних у білінг-сервіс для створення та відправки рахунку.



Рисунок 2.4 - Сценарій практичної реалізації автоматизація фінансової звітності

Результатом такої композиції є формування сервісу з доданою цінністю, що виключає людський фактор та скорочує адміністративні витрати на 70–80%.

2.4. Специфіка та виклики композиції сервісів у доменах домашньої автоматизації

Хоча автоматизація будівель тривалий час була прерогативою комерційних та офісних об'єктів, сучасний етап розвитку характеризується експансією технологій domotics у приватний житловий сектор. Ця тенденція зумовлена парадигмою «сталого розвитку» та потребою в енергоефективності. Композиція сервісів у цьому контексті розглядається як ключовий інструмент підвищення продуктивності та спрощення керування гетерогенними мережами побутових пристроїв [28]. Розумний дім

трактується як динамічне середовище, що потребує безперервної реконфігурації сервісів залежно від зміни контексту [32].

2.4.1. Архітектура сервісів та проблема стандартизації

Екосистема розумного будинку базується на взаємодії трьох рівнів компонентів, що виступають як атомарні сервіси:

- Сенсорний рівень: Сервіси збору даних про стан середовища (температура, освітленість, присутність, час).
- Виконавчий рівень: Сервіси керування фізичними об'єктами (термостати, освітлювальні прилади, побутова техніка).
- Логічний рівень: Програмне забезпечення для оркестрації, що координує взаємодію між сенсорами та виконавчими механізмами.



Рисунок 2.5 – Архітектура екосистеми розумного будинку у вигляді атомарних сервісів

Критичним фактором для композиції є стандартизація інтерфейсів. Важливим кроком у цьому напрямку стала ініціатива Open Services Gateway Initiative (OSGi) [25], яка пропонує модульну архітектуру для розгортання

сервісів. Проте ринок залишається фрагментованим через співіснування численних протоколів (Zigbee, Z-Wave, Matter), що створює проблему семантичної та синтаксичної гармонізації під час синтезу складних сценаріїв.

У даному домені переважають користувачі з низьким рівнем технічної грамотності, що накладає особливі вимоги до СПКС:

Експерт домену (Мешканець): Володіє глибоким розумінням бажаних сценаріїв комфорту («знання домену»), але не має навичок програмування. Для такої групи система повинна забезпечувати інтелектуальний супровід.

- Технічний експерт (Інстальатор): Здійснює первинне налаштування складних інженерних систем. Він оперує низькорівневими параметрами композиції, забезпечуючи надійність базової інфраструктури.

Взаємодія в розумному середовищі є багатомодальною. Ми виділяємо два фундаментальні сценарії, що визначають вимоги до людино-комп'ютерної взаємодії (HCI):

Таблиця 2.2 - Порівняльна характеристика сценаріїв композиції в домашній автоматизації

Параметр	Статична композиція	Динамічна композиція
Етап	Встановлення та базова конфігурація	Адаптація до поточної ситуації/погоди
Пристрій	ПК, стаціонарні термінали	Смартфони, смарт-годинники, голос
Часовий ресурс	Високий (ретельне планування)	Мінімальний (потреба «тут і зараз»)
Стійкість	Персистентна	Тимчасова

2.4.2. Аналіз сценарію домашньої автоматизації як прикладу контекстно-залежної композиції

Для ілюстрації практичної реалізації розглядається кейс динамічної реконфігурації мультимедійного сервісу в розумному просторі (eHome):

Сценарій: Користувач (Джон) переміщується між функціональними зонами приміщення (вітальня → кухня).

Логіка композиції:

- 1. Детекція: Сенсори присутності фіксують зміну координати об’єкта.
- 2. Оркестрація: Система eHome ідентифікує профіль Джона та його активний музичний стрім.
- 3. Реконфігурація: Композиція автоматично перемикає вихідний потік (output stream) з динаміків вітальні на вбудовану акустику кухні, зберігаючи позицію відтворення (state persistence).

Цей сценарій демонструє перехід від жорстко заданих алгоритмів до адаптивної сервісної архітектури, де композиція виконується прозоро для користувача, забезпечуючи безперервність цифрового досвіду.

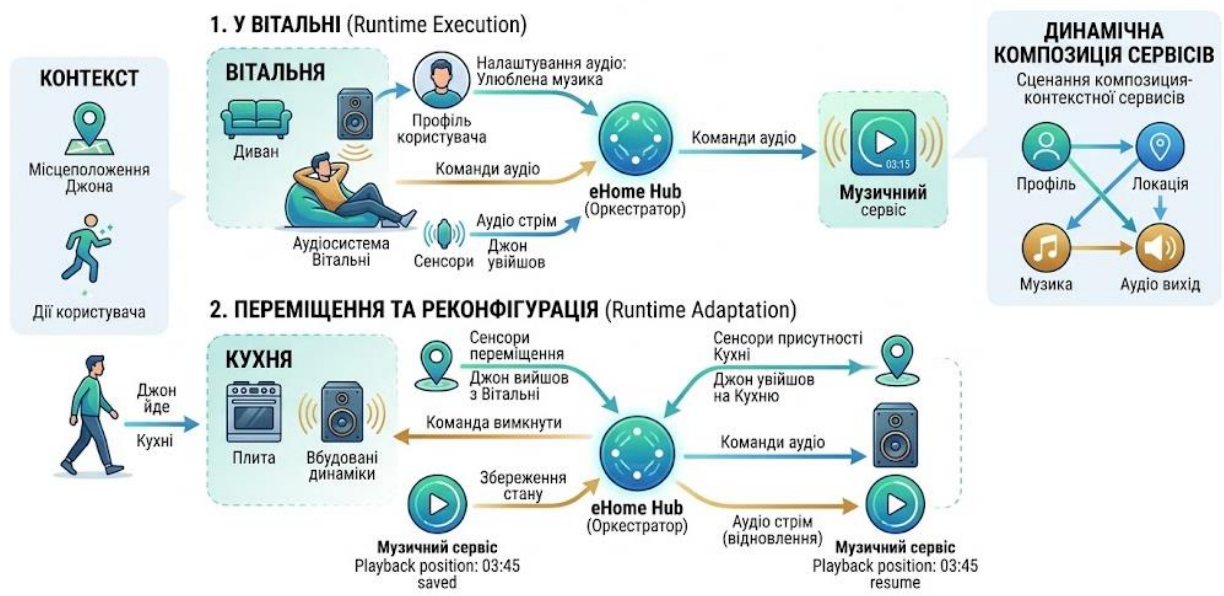


Рисунок 2.6 – Візуалізація сценарію домашньої автоматизації як прикладу контекстно-залежної композиції

Рисунок 2.6 детально ілюструє послідовність дій та взаємодію компонентів eHome (Оркестратора) при переміщенні користувача (Джона) між кімнатами:

1. Детекція присутності (вітальня) - сенсори фіксують вхід Джона. Оркестратор ідентифікує його профіль та активує музичний сервіс на динаміках вітальні.

2. Детекція переміщення - Джон йде на кухню. Сенсори детектують вихід з вітальні та вхід на кухню.

3. Ре конфігурація - оркестратор вимикає динаміки вітальні, зберігає стан відтворення (Playback state: position 03:45) у музичному сервісі та вмикає динаміки кухні, відновлюючи стрім з тієї ж позиції.

Рисунок 2.6 чітко показує потоки даних (стрім музики, контекст) та команд, а також ключові концепції: збереження стану та динамічну реконфігурацію. Результатом є безперервність цифрового досвіду для користувача.

2.5. Електронна комерція як динамічний домен для персоналізованої сервісної інтеграції

Електронна комерція (e-commerce) є одним із найбільш інтенсивних сегментів цифрової економіки, що трансформує традиційні моделі споживання. Сучасні споживачі дедалі частіше делегують складні завдання — від закупівлі електроніки до комплексного планування логістики та відпочинку — спеціалізованим онлайн-платформам. У цьому контексті композиція сервісів виступає як інструмент агрегації цінності, дозволяючи поєднувати пропозиції різних постачальників у межах єдиного інтерфейсу користувача.

2.5.1. Поточний стан та еволюція сервісної архітектури

На сучасному етапі сервіси в електронній комерції переважно функціонують у «фоновому» режимі, де логіка їхнього поєднання жорстко детермінована розробником платформи. Користувач взаємодіє з готовим

					БР.ІП – 62.00.00.000 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

продуктом, не маючи можливості впливати на структуру сервісного ланцюжка.

Виділяють два домінантні сценарії композиції в туристичному сегменті:

- Концепція «Покупки в одному місці» - сервіс аранжування турів, що інтегрує бронювання мультимодальних перевезень. Агрегація кількох послуг в одному кошику часто стимулює лояльність через механізми пакетних знижок.

- Мультиагентне порівняння - сервіси порівняння авіарейсів, що забезпечують прозорість ринку. Висока варіативність вибору підвищує ймовірність задоволення специфічних вимог користувача, що відповідає стратегії охоплення «довгого хвоста» запитів.

Цільова аудиторія e-commerce характеризується мінімальним рівнем технічної підготовки в галузі програмної інженерії. Основним бар'єром для впровадження користувацько-орієнтованої композиції є дефіцит знань про технічну реалізацію API.

Класифікація користувачів у цьому домені переважно відповідає профілю «Початківця»:

- Знання домену: Високі або середні (завдяки маркетингу та особистому досвіду).

- Технічні знання: Майже відсутні.

- Потреба в підтримці: Критична. Система повинна пропонувати абстрактні візуальні інструменти, які приховують складність запитів до серверів авіакомпаній або готельних мереж.

2.5.2. Технологічні обмеження та відкриті інтерфейси

Незважаючи на різноманітність сервісів, багато постачальників дотримуються стратегії «закритих екосистем», обмежуючи доступ до своїх даних через пропріетарні інтерфейси. Це створює проблему Vendor Lock-in

					БР.ІП – 62.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

(залежності від постачальника). Покращення доступності відкритих API та впровадження семантичних описів продуктів є необхідною умовою для переходу до справжньої користувацької композиції.

В електронній комерції ключовим метричним показником є коефіцієнт конверсії (CR). Компанії впроваджують нові інструменти, зокрема композицію сервісів, лише за умови їхнього позитивного впливу на обсяги продажів.

$$CR = \frac{\text{Кількість транзакцій}}{\text{Загальна кількість відвідувачів}} \times 100\%$$

З точки зору людино-комп'ютерної взаємодії, спостерігається чіткий вектор переходу до m-commerce (мобільної комерції). Молодші покоління споживачів надають перевагу транзакціям через мобільні термінали, що вимагає від композиційних систем високої швидкодії та адаптивності інтерфейсів під малі екрани.

2.5.3. Практичний кейс композиції сервісів в електронній комерції

Розглянемо сценарій, де користувач самостійно формує складну послугу в режимі реального часу:

- Крок 1: Бронювання авіарейсу на основі бюджету та часових лімітів.
- Крок 2 (Залежний): Система автоматично використовує метадані квитка (час прибуття, локація аеропорту) для пошуку доступних сервісів оренди авто.
- Крок 3 (Просторовий): Інтеграція з картографічним сервісом (наприклад, Google Maps) для побудови маршруту до готелю.
- Крок 4: Бронювання проживання з урахуванням логістики, сформованої на попередніх етапах.

Така послідовна композиція створює персоналізований сервісний ланцюжок, який є набагато ефективнішим за стандартні пропозиції

					БР.ІП – 62.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

турагентств, оскільки він динамічно адаптується до найменших змін у планах користувача.



Рисунок 2.7 – Практичний кейс динамічної композиції туристичного пакету послуг

Рисунок 2.7 ілюструє послідовність дій користувача та автоматичні процеси оркестрації при самостійному формуванні послуги в режимі реального часу:

1. Бронювання авіарейсу (дизайн-тайм): вибір рейсу на основі бюджету та часових лімітів. Це початковий етап, що генерує метадані квитка.
2. Оренда автомобіля (залежний сценарій): система автоматично використовує метадані квитка (час прибуття, локація) для пошуку доступних сервісів оренди авто.
3. Побудова маршруту (просторовий контекст): інтеграція з картографічним сервісом (наприклад, Google Maps) для побудови маршруту від аеропорту до готелю.
4. Бронювання проживання: Вибір та бронювання готелю на основі попередніх етапів (логістика).

2.6. e-Health як середовище для впровадження сервіс-орієнтованих систем

Електронна охорона здоров'я (e-Health) представляє собою сучасну парадигму медичної практики, що базується на інтеграції цифрових процесів та мережових комунікацій у систему надання медичної допомоги. Вона охоплює взаємодію між широким спектром суб'єктів: від профільних медичних установ та фахівців до пацієнтів і споживачів профілактичних послуг. Розвиток інтелектуальних систем моніторингу та персоналізованих рекомендацій сприяє трансформації пацієнта з пасивного об'єкта лікування на активного учасника процесу управління власним здоров'ям.

2.6.1. Технологічна конвергенція e-Health та систем автоматизації

Концептуально композиція сервісів у домені e-Health має спільні риси з домашньою автоматизацією будівель, проте характеризується вищими вимогами до надійності та безпеки. В основі лежить взаємодія натільних сенсорних мереж (Body Area Networks, BAN) та виконавчих механізмів (інфузійних pomp, кардіостимуляторів тощо).

Даний підхід дозволяє реалізувати архітектуру дистанційного моніторингу, де дані з сенсорів (ЧСС, рівень кисню, ЕКГ) у реальному часі агрегуються системою підтримки композиції для прийняття діагностичних рішень.

У домені e-Health виділяються дві основні групи користувачів, кожна з яких має специфічні вимоги до системи композиції:

1. Медичні фахівці - володіють вичерпними знаннями домену, але часто виявляють консерватизм щодо зміни усталених клінічних протоколів.

Система повинна інтегруватися в існуючий робочий процес, не створюючи надлишкового адміністративного навантаження. Для цієї групи

					БР.ІП – 62.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

технічна складність композиції має бути прихована за медичною семантикою.

2. Пацієнти та споживачі - низький рівень технічних знань та фрагментарне розуміння медичних сервісів.

Максимальний рівень автоматизації та інтуїтивно зрозумілі інтерфейси, орієнтовані на профілактику та підтримку здорового способу життя.

2.6.2. Стандартизація та складність сервісних ланцюжків

Сервісна інфраструктура охорони здоров'я критично залежить від інтероперабельності. На відміну від домашньої автоматизації, де домінують протоколи Zigbee або Matter, e-Health спирається на галузеві стандарти обміну медичними даними, такі як HL7 (Health Level Seven) та FHIR (Fast Healthcare Interoperability Resources).

Складність композиції в даному домені зумовлена високою розмірністю вхідних параметрів. Окрім технічних характеристик сервісу, система повинна враховувати юридичні та етичні обмеження (наприклад, відповідність регламенту GDPR або HIPAA щодо захисту персональних даних).

На відміну від комерційних доменів, умови композиції в e-Health часто є Safety-Critical (критичними для безпеки):

- виникає потреба у жорсткому реальному часі для обробки екстрених ситуацій.

- враховуючи високу відповідальність, медичні працівники потребують спеціалізованої підготовки та сертифікації для роботи з системами динамічної композиції.

2.6.3. Кейс: Дистанційний моніторинг критичних станів здоров'я

Кейс реалізує сценарій інтелектуальної оркестрації сервісів для пацієнтів із хронічними захворюваннями.

					БР.ІП – 62.00.00.000 ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

Структура композиції:

- Сервіс BAN (Body Area Network) - безперервне зняття життєво важливих показників через мобільні сенсори.
- LBS-сервіс (Location Based Service) - визначення геопозиції пацієнта.
- Логічний шлюз (Orchestrator) - аналіз даних. Якщо показники виходять за межі норми, а пацієнт знаходиться поза межами медичного закладу, система ініціює наступний крок.
- Сервіс екстрених повідомлень - надсилання критичного сповіщення лікарю та виклик бригади швидкої допомоги з передачею точних координат.

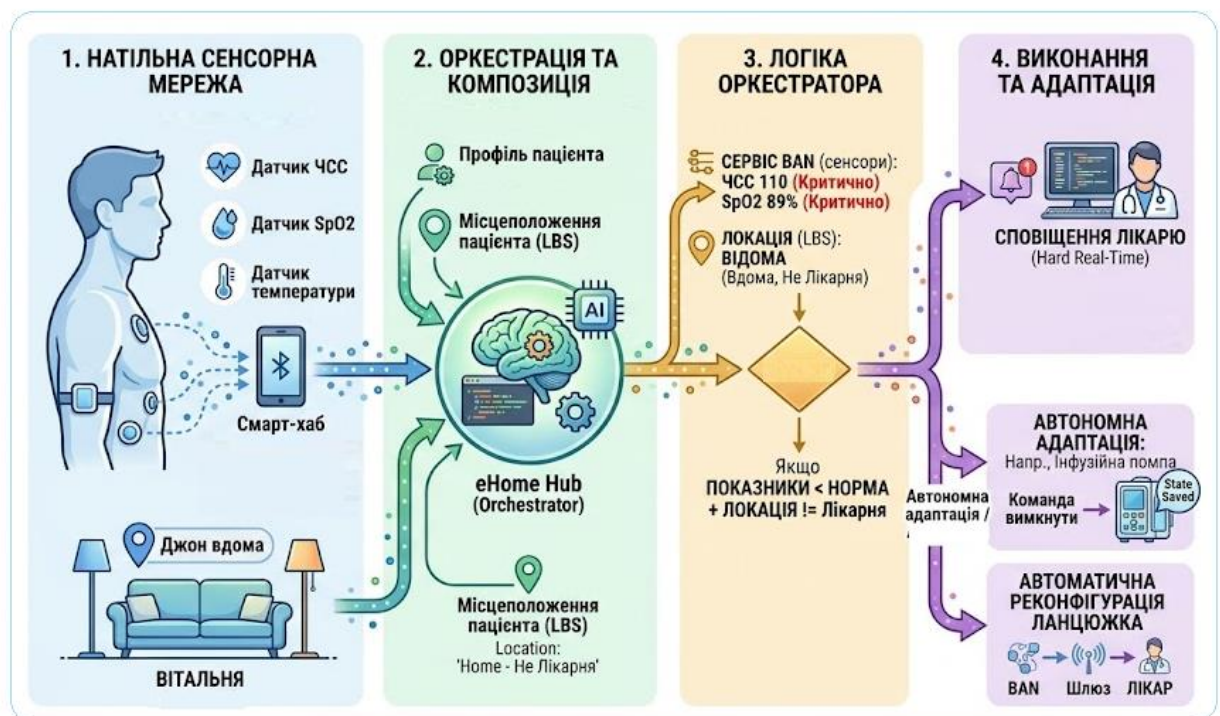


Рисунок 2.8 – Приклад використання композиції сервісів моніторингу стану здоров'я

Рисунок 2.8 ілюструє послідовність дій та взаємодію компонентів eHome Hub (Оркестратора) при моніторингу пацієнта у нелікарняному середовищі:

1. Натільна сенсорна мережа: Початковий етап, що показує збір життєво важливих показників та їх передачу на смарт-хаб.

2. Оркестрація та композиція: дані надходять до ядра оркестратора, який також використовує профіль пацієнта та LBS-сервіс (геопозиція: вдома, не лікарня).

3. Логіка оркестратора (Критичний аналіз): Діаграма показує порогові значення критичного стану та логіку блоку прийняття рішень: «Якщо ПОКАЗНИКИ < НОРМА + ЛОКАЦІЯ != Лікарня => СПОВІЩЕННЯ ЛІКАРЮ».

4. Виконання та адаптація (Response) - цей етап ілюструє миттєву реакцію:

- медичний працівник отримує деталізоване сповіщення про стан пацієнта.
- команда вимикає (або зменшує дозу) напр., інфузійної помпи, зберігаючи стан відтворення та ініціюючи автоматичну реконфігурацію ланцюжка.

Даний приклад демонструє, як композиція атомарних сервісів формує систему з високою соціальною цінністю, забезпечуючи безпеку життєдіяльності пацієнта в неконтрольованому середовищі.

2.7. Порівняльний аналіз прикладних доменів за критеріями користувацько-орієнтованої композиції

Для завершення аналітичного огляду та структурування отриманих даних нижче наведено підсумкову таблицю. Вона відображає ключові відмінності між доменами за критеріями, що визначають архітектурні вимоги до системи користувацько-орієнтованої композиції.

Проведений аналіз дозволяє виділити специфічні патерни для кожного домену, що безпосередньо впливають на проектування прототипу, що показано в таблиці 2.3.

					БР.ІП – 62.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

**Таблиця 2.3 - Порівняльна характеристика доменів застосування
композиції сервісів**

Критерій порівняння	Підприємництво (SaaS)	Домашня автоматизація	Електронна комерція	Електронна медицина
Домінантний профіль користувача	Просунутий користувач / Експерт домену	Початківець	Початківець (Споживач)	Медичний фахівець / Пацієнт
Типовий момент композиції	Design-time (проектування)	Runtime (динамічна адаптація)	Runtime (в процесі вибору)	Runtime (критичний стан)
Основний тип пристрою	ПК / Ноутбук (стаціонарно)	Смартфон / Голос / Сенсори	Смартфон / Веб-браузер	Спеціалізовані датчики / Планшети
Ключовий драйвер цінності	Ефективність (ROI), автоматизація	Комфорт, енергозбереження	Конверсія (CR), персоналізація	Безпека життя (Safety-critical)
Рівень когнітивної підтримки	Низький (інструменти аналітики)	Високий (повна автоматизація)	Високий (рекомендаційні системи)	Максимальний (верифікація протоколів)
Стандарти та протоколи	REST, SOAP, JSON	OSGi, Matter, Zigbee	Web Services, HTTP	HL7, FHIR, GDPR
Стійкість композицій	Висока (рекурентні процеси)	Середня (динамічні сценарії)	Низька (одноразові транзакції)	Висока (безперервний моніторинг)

Найбільшим спільним викликом для всіх доменів (окрім великого бізнесу) є відсутність технічних знань у користувача-комполітора. Це підтверджує необхідність використання семантичних анотацій (OWL-S), які дозволяють системі "розуміти" намір користувача без необхідності написання коду.

Встановлено, що умови композиції в реальному часі домінують у більшості сучасних сценаріїв. Це висуває вимоги до системи щодо автоматичного збору контекстних даних (геопозиція, стан пристрою, параметри здоров'я) для автоматизації вибору сервісів.

Таким чином, розробка прототипу буде базуватися на гібридній моделі, яка поєднує гнучкість налаштувань для експертів та інтелектуальний супровід для рядових користувачів, забезпечуючи стабільне функціонування сервісних ланцюжків у динамічних середовищах.

2.8 Висновки по розділу

У другому розділі здійснено аналіз прикладних доменів, що дозволило визначити специфіку застосування композиції сервісів у різних галузях. Встановлено, що ефективність реалізації користувацько-орієнтованих систем значною мірою залежить від рівня стандартизації сервісів та доступності відкритих інтерфейсів. Виявлено ключові ергономічні та технологічні обмеження, що впливають на процес людино-машинної взаємодії. Проведено порівняльний аналіз доменів, який показав відмінності у складності реалізації композицій залежно від контексту використання. Обґрунтовано необхідність урахування профілів користувачів та контекстних факторів при проєктуванні сервісних систем.

РОЗДІЛ 3. АРХІТЕКТУРНА РЕАЛІЗАЦІЯ ПРОТОТИПУ СИСТЕМИ КОРИСТУВАЦЬКО-ОРІЄНТОВАНОЇ КОМПОЗИЦІЇ

3.1. Представлення архітектури клієнтського додатку

У даному розділі представлена архітектурна модель клієнтського додатка, розробленого для програмної реалізації сценаріїв, детермінованих у попередніх етапах дослідження. На рисунку 3.1 візуалізовано концептуальну схему прототипу, що ілюструє взаємодію між користувацьким інтерфейсом та модулем оркестрації сервісів.

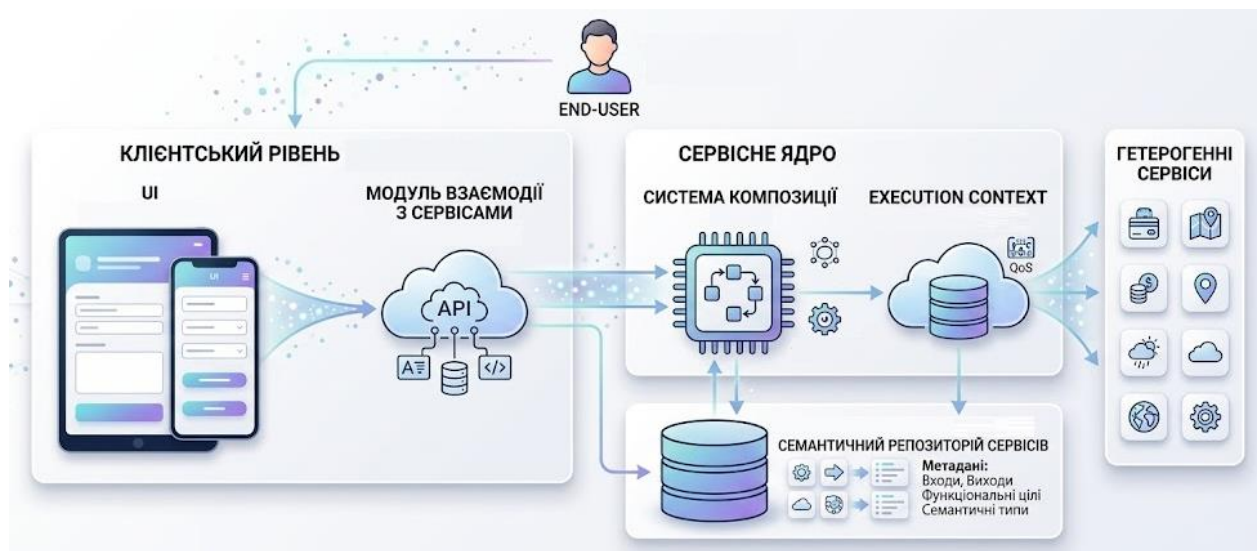


Рисунок 3.1 - Концептуальна архітектура прототипу системи підтримки композиції

Прототип складається з двох ключових рівнів:

- Клієнтський рівень - містить адаптивний графічний інтерфейс користувача (GUI) та модуль взаємодії із сервісним ядром.
- Сервісний рівень (Оркестратор): Виконує функції динамічного синтезу сервісних ланцюжків та управління життєвим циклом композиції.

Детальний опис механізмів генерації користувацьких інтерфейсів для доступу до атомарних сервісів та алгоритмічне забезпечення процесів композиції та верифікації зв'язків представлено у наступних підрозділах.

3.1.1. Функціональні характеристики сервісного ядра композиції

Центральним елементом архітектури є інтелектуальне сервісне ядро, яке реалізує універсальне рішення для абстрагування технічної складності процесів інтеграції від кінцевого користувача. Ключовою метою ядра є підтримка гетерогенних груп зацікавлених сторін через надання прозорого доступу до програмних ресурсів, здатних задовольнити їхні специфічні потреби.

Взаємодія з ядром організована за моделлю Service-as-a-Service (SaaS): воно функціонує як незалежний веб-сервіс, доступ до якого здійснюється через стандартизовані типи взаємодії (API). Такий підхід забезпечує кросплатформеність клієнтського додатка та можливість його розгортання на різних типах терміналів (мобільних пристроях, стаціонарних станціях тощо).

3.1.2. Семантичне моделювання та репозиторій сервісів

Ефективність динамічної композиції базується на роботі ядра з семантичним репозиторієм сервісів. Даний репозиторій акумулює метадані про всі доступні в мережевому середовищі програмні компоненти. Для забезпечення високого рівня автоматизації описи сервісів базуються на мовах семантичного моделювання.

Семантичний опис кожного сервісу включає наступні критичні атрибути:

- Вхідні та вихідні параметри (I/O) - технічні специфікації даних, необхідні для забезпечення синтаксичної сумісності компонентів.
- Функціональні цілі (Service Goals) - декларативний опис того, який бізнес-результат або дію виконує сервіс.

					БР.ІП – 62.00.00.000 ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

- Семантичні типи даних. Використання спільних словників та онтологій для опису сутностей (наприклад, «час», «місце», «ціна»), що дозволяє системі здійснювати логічне виведення при побудові ланцюжків.

Застосування семантичної типізації вирішує два фундаментальні завдання:

- На рівні ядра забезпечує точність алгоритмів автоматичного планування (AI Planning) та виявлення сервісів (Discovery).

- На рівні інтерфейсу надає метадані для динамічної побудови форм введення даних та візуалізації результатів, адаптуючи UI під конкретний контекст сервісу без втручання програміста.

3.2. Механізми семантичної трансформації та архітектурні особливості клієнтського рівня

Клієнтський додаток у системі користувацько-орієнтованої композиції виконує роль інтелектуального посередника між семантичним ядром оркестрації та кінцевим користувачем. Основним завданням цього компонента є реалізація універсального інтерфейсу, здатного трансформувати складні семантичні описи сервісів у зрозумілі та функціональні елементи керування.

В основі архітектурного рішення клієнтського додатка лежить концепція подолання семантичного розриву через два фундаментальні процеси:

- Grounding (Конкретизація) - процес перетворення високорівневого семантичного опису (абстрактних концептів) у синтаксичні параметри та конкретні елементи інтерфейсу користувача.

- Lifting (Абстрагування) - процес інтерпретації синтаксичних результатів виконання сервісу та їх перетворення на семантичні сутності, придатні для подальшої обробки механізмами композиції.

					БР.ІП – 62.00.00.000 ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

Для забезпечення максимальної адаптивності було прийнято рішення делегувати виконання процесів grounding та lifting безпосередньо клієнтському додатку. Це дозволяє здійснювати гнучку генерацію інтерфейсу на трьох етапах: при формуванні вхідних форм, безпосередньо під час виклику сервісу та при візуалізації результатів.

3.2.1. Адаптивна генерація вхідних інтерфейсів на основі семантичних типів

Кожен атомарний сервіс у репозиторії характеризується набором вхідних параметрів. Клієнтський додаток забезпечує динамічне відображення відповідних віджетів (UI-елементів) для збору даних. Замість статичних форм, система використовує семантичні типи, надані модулем оркестрації, для вибору найбільш релевантного способу введення.

Наприклад, якщо параметр сервісу має семантичний тип «Категорія_Кухні», система автоматично генерує список (Drop-down list) або набір тегів, замість стандартного текстового поля. Це мінімізує помилки введення та знижує когнітивне навантаження на користувача. Процес вибору елемента UI базується на матриці відповідності семантичних доменів та типів контролерів (див. таблицю 3.1).

Таблиця 3.1 - Матриця відображення семантичних типів на елементи інтерфейсу

Семантичний тип	Елемент інтерфейсу	Приклад застосування
Enumeration / Class	Випадаючий список / Radio-buttons	Вибір типу страви, класу готелю
Spatial / Coordinate	Інтерактивна карта (Map Picker)	Вибір точки на карті, локація
Temporal / DateTime	Календар / Time-picker	Час прибуття, дата бронювання
Boolean	Перемикач (Toggle / Checkbox)	Наявність Wi-Fi, вегетаріанське меню

3.2.2. Виконання сервісів та міжпротокольна медіація

Децентралізація управління виконанням та його перенесення на сторону клієнта обумовлене високою гетерогенністю сучасного сервісного середовища. Реалізація універсального виконавця на стороні серверного ядра ускладнюється різноманітністю протоколів, методів автентифікації та специфічних для домену особливостей.

Сучасні веб-сервіси базуються на різних архітектурних стилях, таких як REST, SOAP або XML-RPC. Клієнтський додаток дозволяє інтегрувати специфічні бібліотеки та SDK, що спрощує взаємодію з популярними API (наприклад, Google Geocode через спеціалізовані модулі). Це забезпечує нативну підтримку особливостей кожного сервісу без перевантаження логіки центрального оркестратора.

Реалізація Grounding та Lifting у процесі виклику:

1. Grounding вхідних даних.

Система трансформує складні семантичні структури у формат, зрозумілий конкретному API. Наприклад, семантичний тип `IOTypes.owl:Coordinate`, що містить пару $P=\langle \text{longitude}, \text{latitude} \rangle$, може бути розбитий на два окремі HTTP-параметри, якщо цього вимагає специфікація цільового веб-сервісу.

2. Lifting вихідних даних.

Результати виконання сервісу часто надлишкові. Наприклад, JSON-відповідь від сервісу пошуку ресторанів може містити десятки полів (ідентифікатори, метадані зображень, технічні прапорці). Процес lifting фільтрує ці дані, виділяючи лише ті властивості (наприклад, `location` або `price_range`), які мають цінність для наступних ланок у ланцюжку композиції.

3.2.3. Класифікація та обробка результатів виконання

Клієнтський додаток повинен здійснювати предиктивну класифікацію отриманих значень для визначення подальшого сценарію взаємодії.

					БР.ІП – 62.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

Виділяється три основні категорії результатів:

1. Значення для представлення - дані, що мають інформативну цінність для користувача, але не можуть бути використані як вхідні параметри для інших сервісів (наприклад, текстовий опис погоди або медіа-контент).

2. Атомарні значення - технічні дані, що використовуються для автоматизації ланцюжка. Наприклад, координати, отримані від сервісу геокодування, передаються напряму в наступний сервіс (наприклад, сервіс виклику таксі) без необхідності візуалізації для користувача.

3. Набори значень - результати пошукових сервісів (списки готелів, рейсів). У цьому випадку клієнтський додаток забезпечує інтерфейс вибору, дозволяючи користувачу власноруч детермінувати об'єкт, з яким буде продовжена робота в межах композиції.

3.3. Стратегії сервісної композиції та механізми управління контекстом виконання

Клієнтський додаток у запропонованій архітектурі виконує функцію інтелектуального інструментарію, що дозволяє кінцевому користувачу конструювати складні композиції сервісів для вирішення багатокрокових завдань. Процес композиції базується на виявленні та усуненні семантичних розривів між вхідними параметрами цільового сервісу та наявними даними користувача.

На рисунку 3.2 продемонстровано логічну модель композиції двох гетерогенних сервісів: FindRestaurant (пошук закладу) та FindLocation (визначення координат). У даному сценарії сервіс FindRestaurant потребує точних географічних координат як вхідного параметра. Оскільки користувач може не володіти цією технічною інформацією в явному вигляді, система ініціює виклик допоміжного сервісу FindLocation. Останній трансформує неформалізовані дані про місцезнаходження (наприклад, назву міста або

					БР.ІП – 62.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

адресу) у синтаксично коректний вихідний параметр (координати), який автоматично передається на вхід основного сервісу.

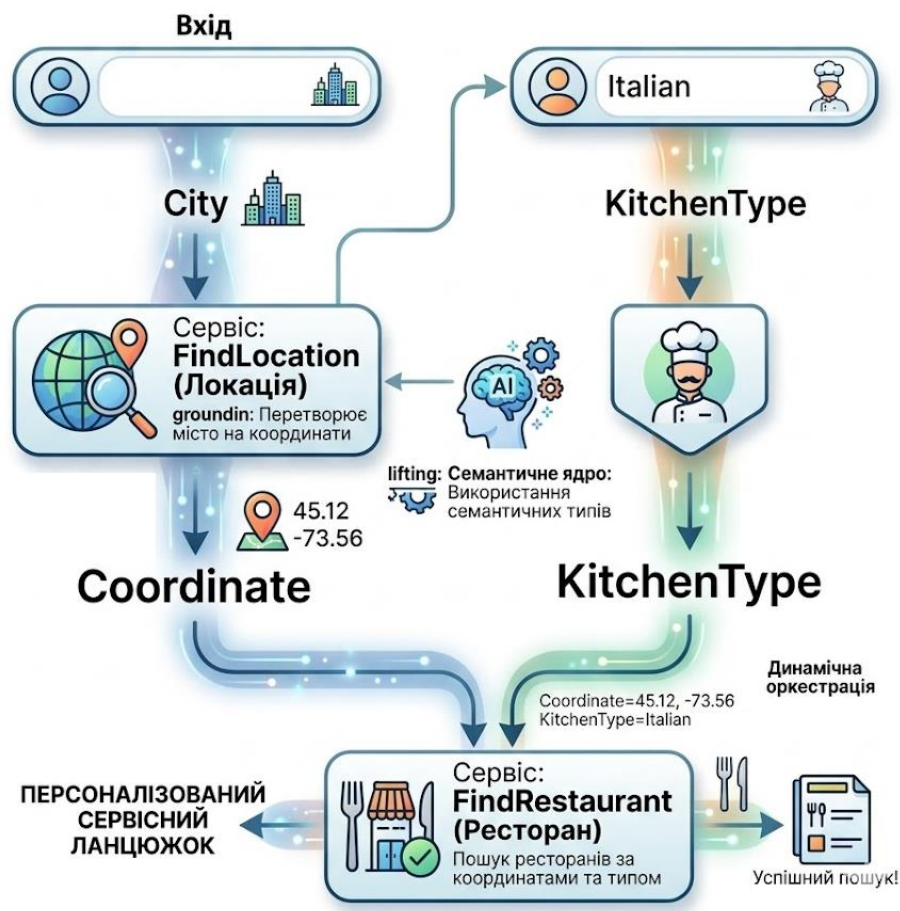


Рисунок 3.2 - Логічна схема композиції сервісів FindRestaurant та FindLocation на прикладі пошуку ресторану

У ході дослідження було визначено та проаналізовано два фундаментальні методи реалізації композиційних процесів: повна (статична) та віртуальна (динамічна) композиція.

3.3.1. Порівняльний аналіз методів композиції

Повна композиція сервісів (рисунок 4.3) передбачає транзакційну модель виконання, за якої весь ланцюжок сервісів активується лише після того, як будуть детерміновані всі вхідні параметри для кожного елемента композиції. За такої стратегії користувач може залишити частину полів вводу

порожніми, делегуючи системі пошук альтернативних джерел даних. Центральний модуль оркестрації автоматично підбирає сервіси-посередники для заповнення прогалин у графі параметрів, проте кінцеве виконання відкладається до моменту повної готовності графа.



Рисунок 3.3 - Модель повної композиції з відкладеним виконанням ланцюжка

Віртуальна композиція сервісів (рисунок 4.4) базується на принципі ітераційного (крокового) вирішення залежностей. У цьому випадку допоміжні сервіси можуть виконуватися проміжно, негайно надаючи результат для візуалізації в оригінальній формі запиту. Це дозволяє користувачу бачити процес «заповнення» відсутніх даних у реальному часі.

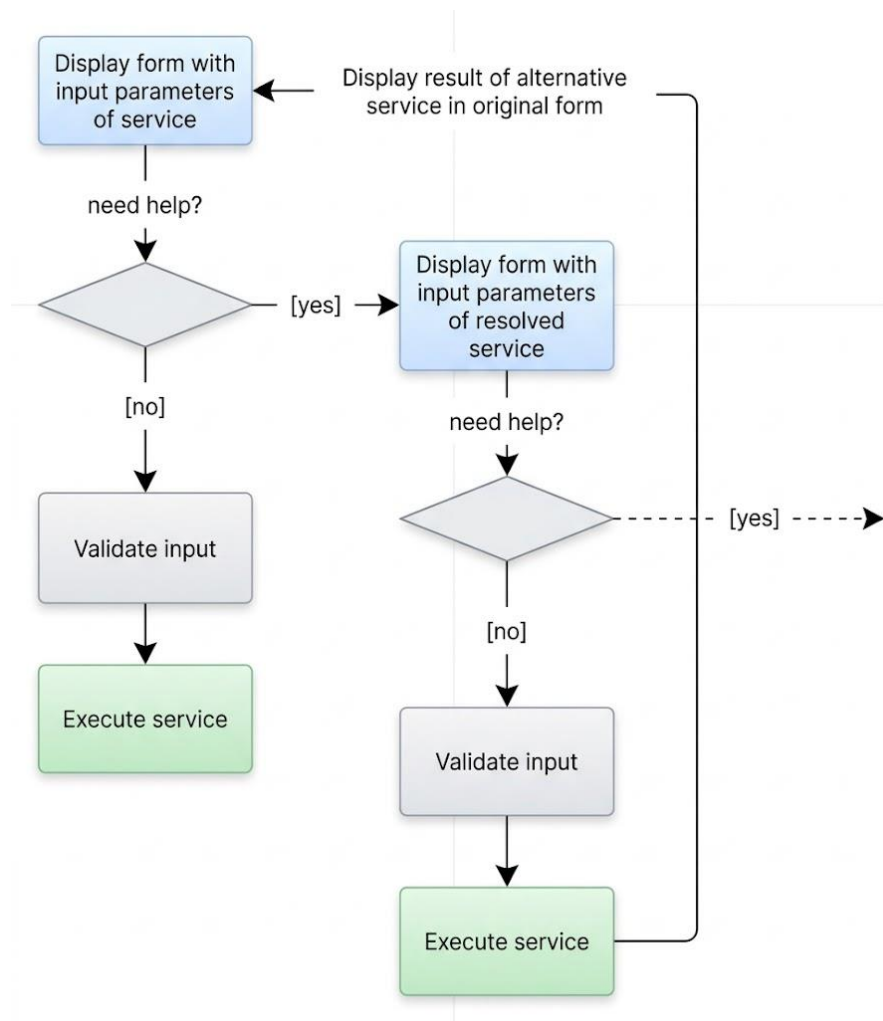


Рисунок 3.4 - Модель віртуальної композиції з ітераційним вирішенням параметрів

Для реалізації в клієнтському додатку було обрано метод віртуальної композиції, що обґрунтовано наступними перевагами:

- Покращений зворотний зв'язок, оскільки користувач контролює кожен етап підбору даних, що підвищує рівень довіри до системи.
- Гнучкість конфігурації. Наявність інтерактивних елементів (кнопок вирішення параметрів) дозволяє користувачу самостійно ініціювати пошук нових сервісів для специфічних полів, не порушуючи загальну логіку форми.
- Зниження когнітивного навантаження: Відсутність потреби залишати поля порожніми «наосліп» робить інтерфейс більш інтуїтивним.

3.3.2. Управління контекстом виконання та станом сесії

Для підтримки стабільної роботи віртуальної композиції клієнтський додаток реалізує механізм контекстної обізнаності. Оскільки процес може включати переходи між декількома сервісами-посередниками, система повинна зберігати стан основної задачі для коректного повернення користувача до початкового етапу.

Архітектурне рішення передбачає розподіл функцій управління станом:

1. Центральний оркестратор зберігає інформацію про семантичний діалог та структуру композиції. Це дозволяє зробити клієнтський додаток «безстаново» щодо глобальної логіки планування, що спрощує його масштабування та кросплатформену адаптацію.

2. Локальне сховище клієнта використовується для буферизації результатів проміжних обчислень. Це необхідно для:

3. Забезпечення швидкого доступу до результатів виконання без повторних запитів до мережі;

4. Реалізації інтерфейсів вибору, де користувач повинен обрати один об'єкт із набору значень, повернутих сервісом-посередником.

Така дворівнева модель управління контекстом забезпечує безперервність користувацького досвіду та цілісність даних при виконанні складних, розгалужених сервісних сценаріїв.

3.4 Висновки по розділу

У третьому розділі розроблено архітектурну модель прототипу системи користувацько-орієнтованої композиції сервісів. Запропонована архітектура базується на використанні семантичного моделювання та адаптивних механізмів формування інтерфейсів. Реалізовано функціональне сервісне ядро, яке забезпечує динамічну інтеграцію сервісів.

					БР.ІП – 62.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		59

РОЗДІЛ 4. ФУНКЦІОНАЛЬНІ ВИМОГИ ДО ЯДРА СИСТЕМИ ТА МЕХАНІЗМИ УПРАВЛІННЯ КОНТЕКСТОМ

У даному розділі формалізовано вимоги до ядра системи оркестрації, виконання яких є критичним для забезпечення користувацько-орієнтованої підтримки в процесах композиції. Архітектура системи повинна базуватися на принципах контекстної обізнаності, що передбачає здатність платформи відстежувати стан взаємодії, акумулювати результати виконання атомарних сервісів та здійснювати інтелектуальне управління життєвим циклом композиції.

4.1. Концептуалізація контексту в системах композиції сервісів

Процес формування сервісної композиції ініціюється кінцевим користувачем шляхом декларативного визначення цілей. Проте для досягнення високого рівня релевантності система повинна враховувати не лише експліцитні запити, а й імпліцитну інформацію про поточну ситуацію — контекст користувача.

Контекст — це будь-яка інформація, що характеризує ситуацію сутності (особи, місця або об'єкта), яка є релевантною для взаємодії між користувачем та програмним додатком.

Впровадження механізмів управління контекстом у процеси композиції дозволяє реалізувати предиктивну підтримку користувача. Контекст є лабільним: він постійно збагачується та змінюється в міру проходження користувачем кроків композиційного сценарію [7].

Сценарій контекстного збагачення (приклад):

Розглянемо випадок послідовної реалізації двох цілей: «відвідування заходу» та «організація харчування».

					БР.ІП – 62.00.00.000 ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

При виборі конкретної події через сервіс «Пошук фестивалів» система автоматично екстрагує просторово-часові метадані (координати локації, дата та час проведення).

Ці дані інтегруються в актуальний контекст сесії.

Коли користувач переходить до другої цілі, система ініціює сервіс «Пошук ресторанів», автоматично підставляючи параметри локації фестивалю як фільтри за замовчуванням. Таким чином, композиція адаптується до ситуації користувача без необхідності повторного введення даних.

4.1.1. Таксономія контекстної інформації

На основі аналізу інформаційних потоків у системі визначено три основні категорії даних, що формують контекст (рис. 4.1).

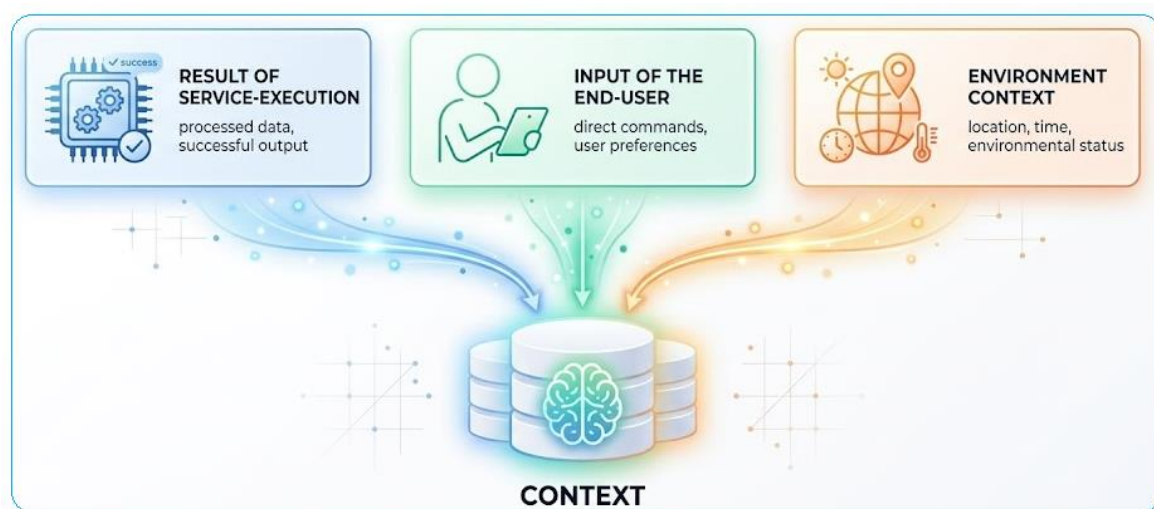


Рисунок 4.1 - Класифікація типів контекстної інформації в системі

1. Результати виконання сервісів. Вихідні дані попередніх етапів композиції часто є необхідними вхідними параметрами для наступних ланок. Оскільки реальні сервіси повертають надлишкові масиви даних, система повинна здійснювати селективну екстракцію лише тих властивостей, що мають семантичну цінність для подальшої оркестрації.

2. Вхідні дані користувача. Параметри, введені користувачем вручну, депонуються в контексті для забезпечення принципу DRY (Don't Repeat Yourself). Це дозволяє системі пропонувати ці значення як підказки при конфігурації нових сервісів у межах однієї або суміжних сесій.

3. Контекст середовища. Дані, що зчитуються автоматично без прямої участі користувача. Сюди належать:

- просторові дані: геопозиція, отримана через GPS-модуль пристрою або API браузера.
- профільні дані: ідентифікаційні дані користувача, його преференції, технічні характеристики пристрою та історія попередніх композицій.

4.1.2. Стратегії актуалізації контексту в архітектурі «Клієнт–Сервер»

Реалізація контекстної обізнаності може бути досягнута двома архітектурними підходами:

- Автономна обізнаність ядра: Платформа самостійно перехоплює та аналізує всі транзакції.
- Делегована обізнаність: Клієнтський додаток виступає активним джерелом актуалізації даних.

З огляду на результати аналізу, представлені у попередньому розділі, де виконання сервісів визначено як доменно-специфічну дію, перший підхід є малоефективним через надмірну складність уніфікації всіх можливих форматів відповідей на стороні сервера.

Більш доцільною є модель, де клієнтський додаток відповідає за збагачення контексту. Це забезпечує гнучкість системи, оскільки саме клієнт володіє логікою візуалізації та інтерпретації результатів. У межах API взаємодії впроваджується метод `add_to_context`, який дозволяє клієнту передавати верифіковані результати виконання та специфічні знання про ситуацію до семантичного ядра.

					БР.ІП – 62.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

4.1.3. Механізми управління сесіями та персистентність даних

Враховуючи багатокористувацький характер сервісної платформи та можливість одночасного виконання кількох сценаріїв одним суб'єктом, система повинна реалізувати надійний механізм управління сесіями.

Кожен запит від клієнтського додатка супроводжується унікальним ідентифікатором сесії (Session ID). На основі цього токена ядро оркестрації здійснює мапування запиту на відповідний сегмент у сховищі контекстів. Такий підхід гарантує ізоляцію даних користувачів та забезпечує цілісність композиційного процесу навіть у безстановому середовищі передачі даних.

4.2. Специфікація протоколів взаємодії між інтерфейсним рівнем та середовищем оркестрації

Для забезпечення ефективної комунікації між клієнтським програмним забезпеченням та ядром системи композиції розроблено набір стандартних типів взаємодії. Ці взаємодії регламентують обмін повідомленнями, що містять метадані про доступні компоненти.

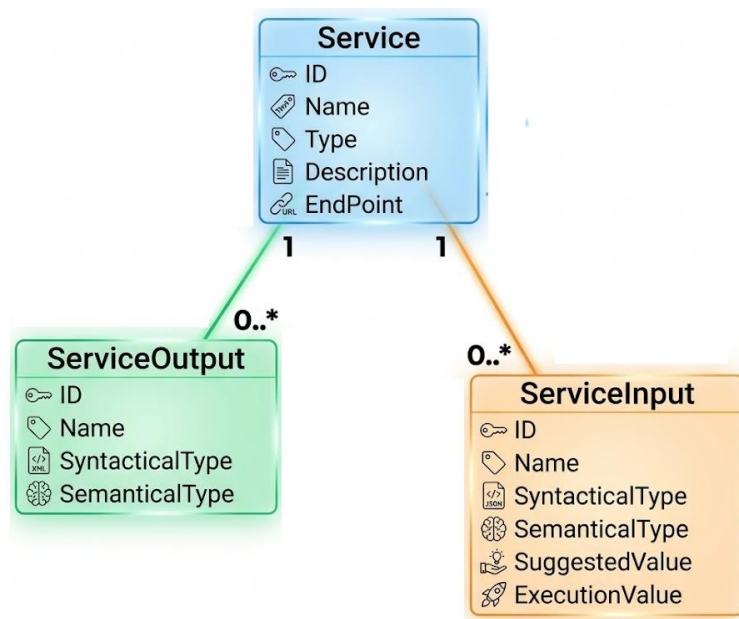


Рисунок 4.2 - Структурна модель метаданих сервісу

На рисунку 4.2 представлено структурну модель опису сервісу, яка є основою для всіх наступних операцій у межах сесії користувача.

Система підтримує два ієрархічні рівні виявлення сервісів: на основі семантичних цілей (інформаційний пошук) та на основі специфікації типів.

Пошук на основі цілей дозволяє користувачеві формулювати запит природною мовою або через вибір абстрактної мети. Система здійснює мапування (mapping) цієї мети на відповідні функціональні типи сервісів. Наприклад, при ініціації потреби в «організації харчування» система ідентифікує як релевантні типи FindSupermarket (питейл), так і FindRestaurant (громадське харчування).

Пошук на основі типів - низькорівневий метод, що оперує безпосередньо ідентифікаторами типів, зареєстрованими у репозиторії.

Специфікацію взаємодії для виявлення за типом наведено у таблиці 4.1.

Таблиця 4.1 - Специфікація взаємодії ServTypeDiscovery

Параметр	Опис
Назва	ServTypeDiscovery
Функціональне призначення	Ідентифікація сервісів на основі відповідності заданому типу в реєстрі.
Інтерфейс	Вхід: List{ServType} Вихід: List{ServTypes, List{MatchServs}}
Залежності	Параметр ServType повинен корелювати з концептами онтології цілей ядра системи.

4.2.1. Процедура вирішення залежностей

Взаємодія типу Resolve service ініціює алгоритм динамічної композиції. Якщо для виконання обраного сервісу бракує вхідних даних, система активує пошук компонентів, чий вихідні параметри семантично відповідають дефіцитним входам. Користувач отримує перелік сервісів-кандидатів і має здійснити вибір для фіксації структури композиції.

Таблиця 4.2 - Специфікація взаємодії ResolveServ

Параметр	Опис
Назва	ResolveServ
Функціональне призначення	Пошук сервісів, здатних генерувати вихідні дані для задоволення передумов раніше вибраного сервісу.
Інтерфейс	Вхід: List{ServiceID, List{InputToResolve}} Вихід: List{Servs}
Залежності	Цільовий сервіс повинен бути попередньо ідентифікований у поточному контексті.

Взаємодія Selection фіксує намір користувача щодо використання конкретного сервісу. У відповідь платформа повертає розширені метадані (див. рис. 4.2), зокрема параметри SuggestedValue. Ці пропозиції генеруються на основі аналізу контексту виконання: система автоматично підставляє значення, що семантично відповідають входам сервісу з історії попередніх дій користувача.

У процесі композиції цей тип взаємодії дозволяє встановити зв'язок (linkage) між виходом сервісу-донора та входом сервісу-акцептора.

Таблиця 4.3 - Специфікація взаємодії SeleServ

Параметр	Опис
Назва	SeleServ
Функціональне призначення	Фіксація обраного сервісу та отримання його параметрів для подальшого виконання.
Інтерфейс	Вхід (SeleForType): List{SeleServID, Type} Вхід (SeleForInput): List{SeleServID, Match-ServID} Вихід: List{ServInfo}
Залежності	Сервіси повинні бути попередньо виявлені через ServTypeDiscovery.

4.2.2. Верифікація вхідних параметрів

Перед активацією сервісу ядро системи проводить валідацію наданих даних. Ця операція не лише перевіряє синтаксичну коректність, а й дозволяє тимчасово депонувати параметри у контексті під час віртуальної композиції. У разі невідповідності даних семантичному або синтаксичному типу, система повертає дескриптор InvalidValue.

Таблиця 4.4 - Специфікація взаємодії ValidateInputs

Параметр	Опис
Назва	ValidateInputs
Функціональне призначення	Перевірка введених користувачем даних на відповідність специфікаціям сервісу.
Інтерфейс	Вхід: List{ServiceID, List{InputID, InputedValue}} Вихід: List{ValidationResult}

Тип взаємодії ExecServ визначає готовність сервісного ланцюжка до активації. Логіка ядра передбачає три стани результату:

- ReadyForExec=true: Наявність усіх необхідних даних дозволяє негайний запуск.
- ReadyForExec=false: Дефіцит параметрів потребує повернення до форми введення або ініціації процесу вирішення залежностей (ResolveServ).
- Null: Відсутність доступних шляхів виконання.

Взаємодія AddToContext забезпечує персистентність результатів у межах сесії. Клієнтський додаток передає значення вихідних параметрів (ServiceOutputValue) до ядра оркестрації для їх подальшого використання як «пропозицій» у наступних ітераціях композиції. Також цей метод використовується для маркування сервісів як успішно завершених.

Останній тип взаємодії забезпечує коректне завершення сеансу. Він видаляє всі тимчасові дані з пам'яті сервера, що сприяє оптимізації ресурсів

платформи та забезпечує конфіденційність даних користувача при переході до нового композиційного циклу.

4.3. Моделювання часової послідовності та динаміки повідомлень у процесах композиції

Для верифікації логічної цілісності системи та аналізу темпоральних аспектів взаємодії компонентів було розроблено діаграми послідовностей. В основу моделювання покладено сім базових типів взаємодії, формалізованих у підрозділі 4.2, що забезпечують реалізацію функціональних сценаріїв.

Метою побудови цих моделей є візуалізація транзакційного обміну повідомленнями між чотирма ключовими сутностями:

- Кінцевий користувач: Ініціатор запиту, що визначає цілі та здійснює вибір сервісів.
- Клієнтський додаток: Інтерфейсний медіатор, що відповідає за візуалізацію даних та виконання сервісів.
- Ядро системи (Оркестратор): Семантичний центр, що здійснює пошук, верифікацію та управління контекстом сесії.
- Зовнішні сервіси: Гетерогенні програмні компоненти, що надають прикладну функціональність.

4.3.1. Реалізація сценарію лінійного пошуку

На рисунку 4.3 представлено діаграму послідовності для сценарію «Пошук ресторану». Цей сценарій ілюструє стандартний цикл виявлення сервісу на основі семантичної мети. Процес починається з ініціації запиту користувачем, після чого клієнтський додаток звертається до ядра для виявлення відповідних типів сервісів. Особливістю даної моделі є асинхронний характер отримання метаданих, що дозволяє динамічно оновлювати інтерфейс користувача без переривання сесії.

					БР.ІП – 62.00.00.000 ПЗ	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		

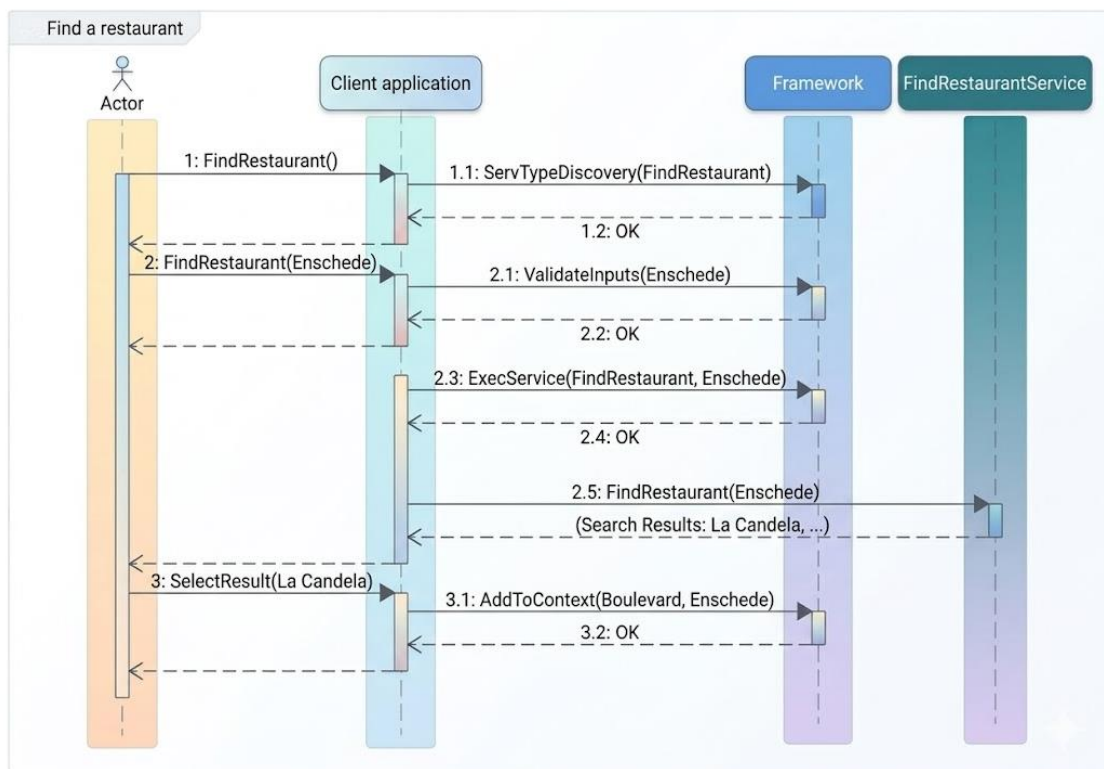


Рисунок 4.3 - Діаграма послідовності процесу виявлення та вибору сервісу пошуку ресторанів

4.3.2. Сценарій ітераційної віртуальної композиції

Більш складний процес взаємодії відображено на рисунку 4.4, де продемонстровано композицію для пошуку активностей. Цей сценарій включає механізми вирішення залежностей (Service Resolution).

Логіка діаграми демонструє ітераційний цикл:

- При виявленні дефіциту вхідних даних (missing inputs) клієнтський додаток надсилає запит на вирішення (ResolveServ).
- Ядро системи ідентифікує допоміжні сервіси-донори.
- Після вибору користувачем відповідного компонента, ядро оновлює граф композиції, а результати проміжних обчислень інтегруються в загальний контекст виконання.

Така модель забезпечує прозорість процесу для користувача, дозволяючи йому контролювати формування складного сервісного ланцюжка на кожному етапі.

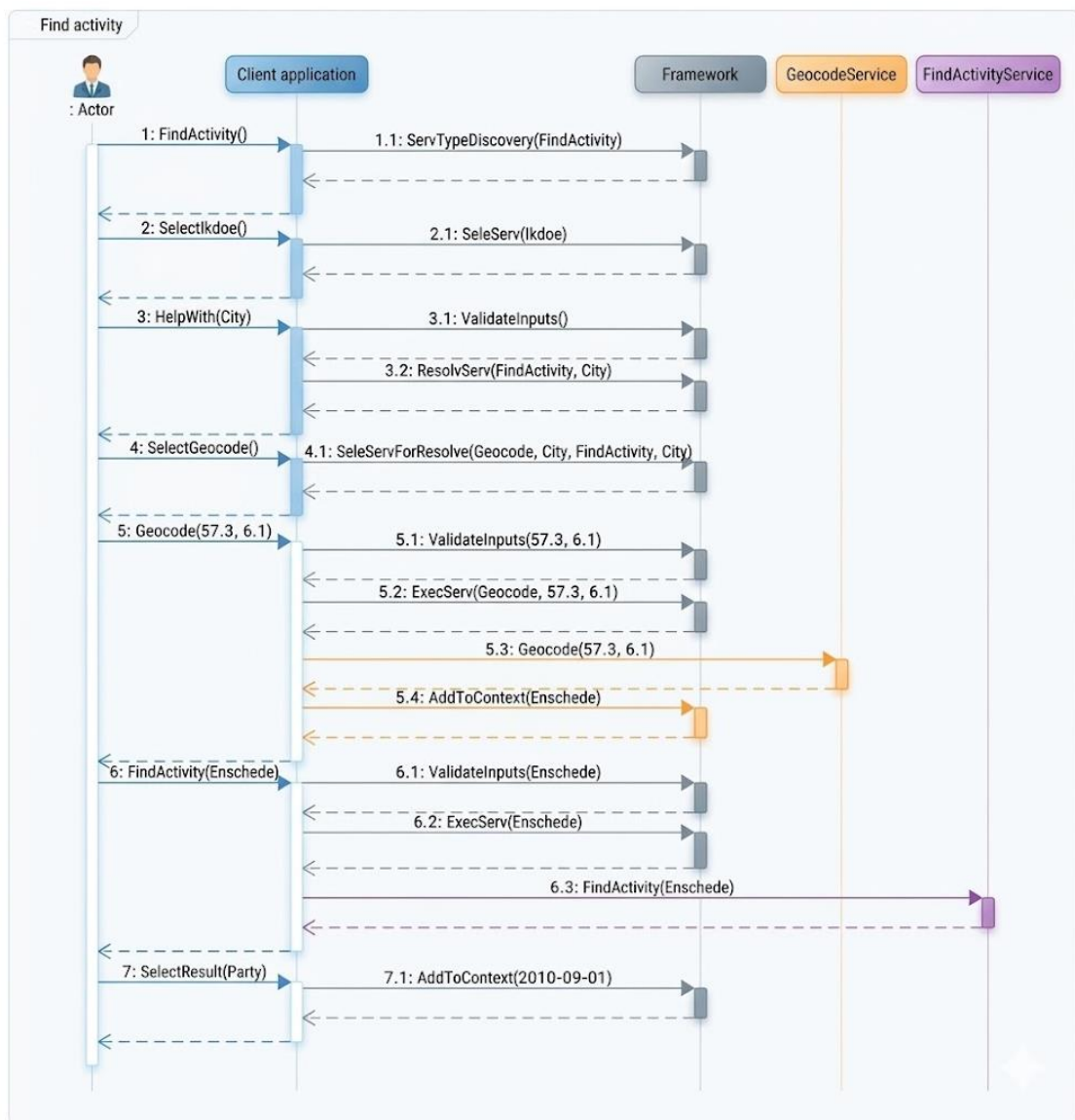


Рисунок 4.4 - Діаграма послідовності ітераційної композиції сервісів для пошуку активностей

4.4 Висновки по розділу

У четвертому розділі сформульовано функціональні вимоги до ядра системи та досліджено механізми управління контекстом. Запропоновано модель контексту, яка враховує різні типи інформації та забезпечує її актуалізацію в процесі виконання композицій. Розроблено підходи до управління сесіями та забезпечення персистентності даних у клієнт-серверній архітектурі.

Визначено протоколи взаємодії між інтерфейсним рівнем і середовищем оркестрації сервісів. Описано моделі часової динаміки виконання композицій, що підвищують керованість та передбачуваність системи.

					БР.ІП – 62.00.00.000 ПЗ	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 5. ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОТОТИПУ СИСТЕМИ КОМПОЗИЦІЇ СЕРВІСІВ

5.1. Реалізація моделі користувацько-орієнтованої композиції на прикладі прототипу «RestaurantFinder»

На рисунку 5.1 подано графічний інтерфейс користувача прототипу «RestaurantFinder», який є практичною реалізацією концепції користувацько-орієнтованої композиції сервісів. У межах аналізу цей інтерфейс слід розглядати не просто як веб-сторінку, а як фронтенд-оркестратор, що інтегрує результати роботи кількох гетерогенних атомарних сервісів у єдиний ціннісний ланцюжок.

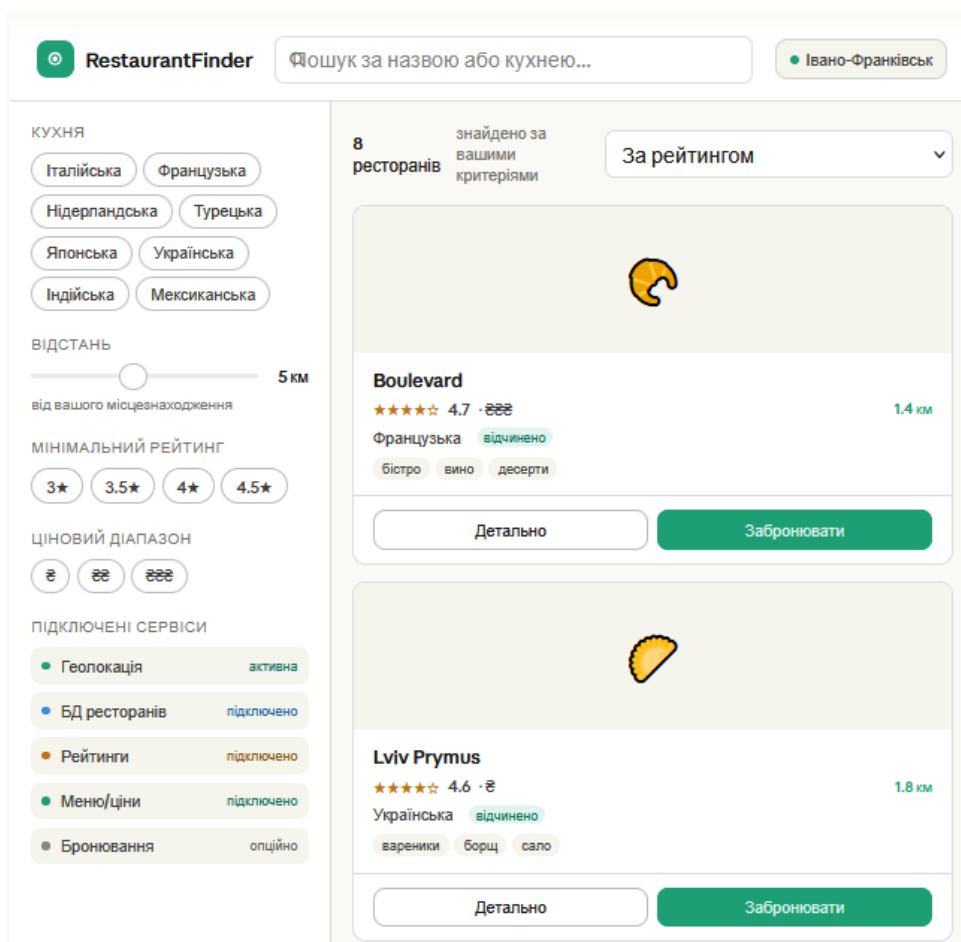


Рисунок 5.1 – Інтерфейс головної сторінки

Нижче наведено структурний опис інтерфейсу крізь призму сервіс-орієнтованої архітектури.

1. Рівень конфігурації та параметризації

Ліва частина інтерфейсу виступає як інструментарій для формування семантичного запиту до системи композиції. Кожен блок фільтрів (Кухня, Відстань, Рейтинг) відповідає конкретним вхідним параметрам (ServiceInputs), які будуть використані для фільтрації та ранжування в базі даних.

Динамічна селекція: Вибір тегів (наприклад, «Французька», «Українська») ініціює процес grounding, перетворюючи візуальний вибір користувача на синтаксичні фільтри для запиту до API.

Контекстна обізнаність: Повзунок відстані та індикатор локації («Івано-Франківськ» у хедері) свідчать про використання LBS-сервісу (Location Based Service), який передає поточні координати користувача в контекст виконання для розрахунку параметрів доступності в реальному часі.

2. Моніторинг активних сервісів (Блок «Підключені сервіси»)

Найбільш репрезентативним елементом з точки зору композиції є блок у лівому нижньому куті. Він візуалізує статус оркестрації та підтверджує, що підсумковий результат сформований шляхом агрегації даних від незалежних постачальників:

- Геолокація (активна): Сервіс визначення координат.
- БД ресторанів та Меню/ціни: Базові інформаційні сервіси, що надають статичні та напівдинамічні дані про об'єкти.
- Рейтинги: Зовнішній сервіс (агрегатор відгуків), що надає дані для сортування за якістю.
- Бронювання (опційно): Виконавчий сервіс (Actuator), який активується лише за потреби користувача, ілюструючи гнучкість композиційного ланцюжка.

					БР.ІП – 62.00.00.000 ПЗ	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дата		

3. Агрегація та візуалізація результатів

Основна область інтерфейсу демонструє результат процесу Lifting — трансформації отриманих синтаксичних відповідей від різних API у структуровані картки об'єктів.

Кожна картка ресторану (наприклад, «Boulevard» або «Lviv Prymus») є результатом злиття даних: назва та тип кухні (з БД ресторанів), поточний рейтинг (із сервісу рейтингів) та розрахована відстань (результат композиції координат користувача та координат закладу).

Наявність кнопок «Детально» та «Забронювати» свідчить про готовність системи до переходу на наступний етап сервісного ланцюжка — перегляду розширених метаданих або ініціації транзакційного сервісу бронювання.

Даний інтерфейс представляє технологічну складність міжплатформної взаємодії. Користувач взаємодіє з абстрактними категоріями («Кухня», «Ціна»), тоді як система в фоновому режимі здійснює оркестрацію п'яти різних сервісів, забезпечуючи цілісність даних та високу швидкість відгуку, що є характерним для сучасних адаптивних сервіс-орієнтованих систем.

5.2. Архітектурний аналіз інтерфейсу в контексті інтегрованих сервісів та динамічної фільтрації

Рисунок 5.2 ілюструє фінальний етап користувацької взаємодії, де інтерфейс виступає в ролі активного агрегатора, що обробляє результати семантичного пошуку в реальному часі.

1. Управління вхідними параметрами композиції

Ліва панель інтерфейсу є блоком детермінації вхідних даних (Input Parameters). На рисунку зафіксовано стан, коли користувач встановив специфічні обмеження для пошуку:

					БР.ІП – 62.00.00.000 ПЗ	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дата		

- Семантичні теги: Обрано типи кухні «Японська», що активує фільтрацію за семантичними типами в реєстрі сервісів.

- Просторові обмеження: Повзунок дистанції встановлено на позначці 5 км, що автоматично взаємодіє з сервісом геолокації для обмеження радіусу пошуку навколо точки «Івано-Франківськ».

- Критерії якості та вартості: Обрано мінімальний рейтинг 4.5 зірки та високий ціновий діапазон. Ці параметри передаються як предумови (Pre-conditions) до сервісу ранжування.

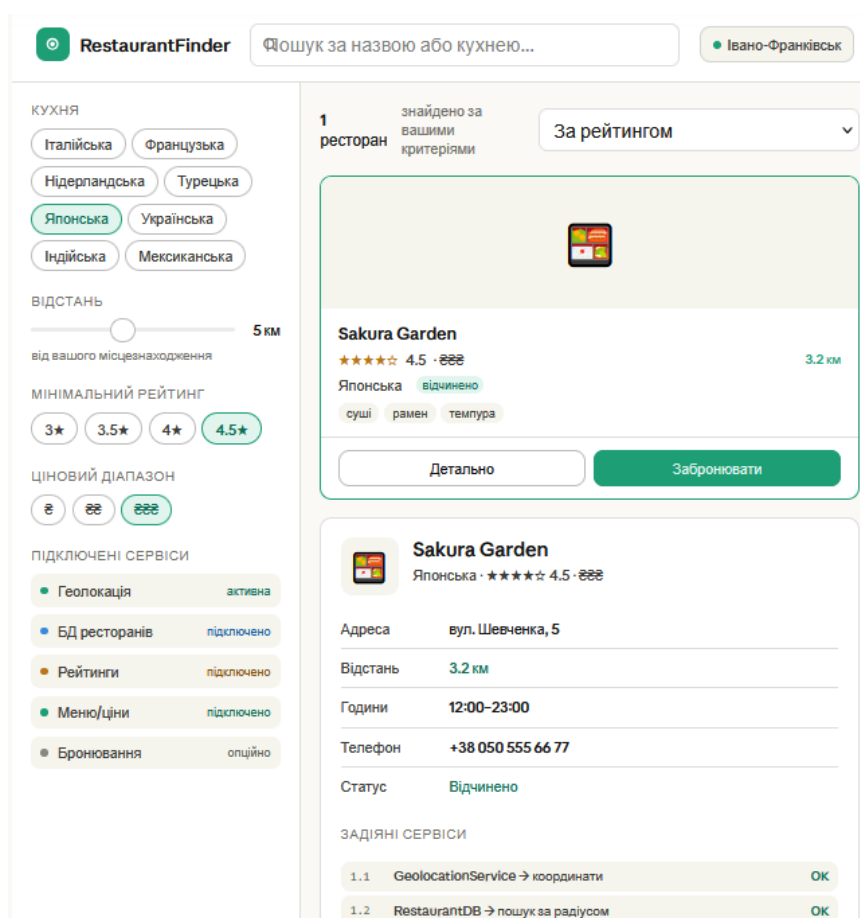


Рисунок 5.2 – Інтерфейс результату роботи системи композиції сервісів

2. Моніторинг статусу оркестрації

Особливу наукову цінність має блок «Підключені сервіси», який відображає архітектурну цілісність поточної композиції. На рисунку чітко видно розподіл ролей:

					БР.ІП – 62.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		74

- Геолокація (статус: активна): Постійний потік даних про місцезнаходження користувача.

- БД ресторанів, Рейтинги та Меню/ціни (статус: підключено): Свідчить про успішну інтеграцію трьох незалежних джерел даних у єдиний інформаційний простір.

3. Синтез та представлення результатів (Lifting)

Центральна частина екрана відображає результат виконання композиції — «1 ресторан знайдено».

Об'єкт «Sakura Garden» є продуктом агрегації даних: назва, категорія та статус («відчинено») отримані з бази даних, рейтинг (4.5) — із сервісу відгуків, а точна відстань (3.2 км) розрахована динамічно шляхом зіставлення координат закладу та користувача.

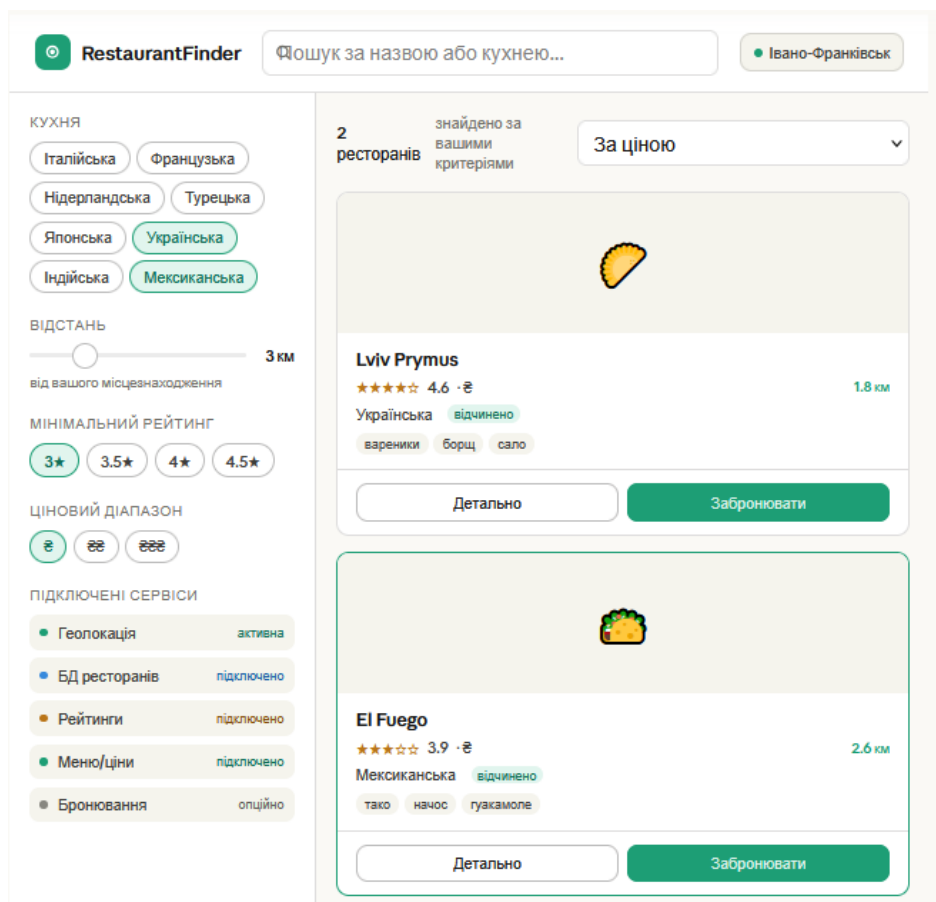


Рисунок 5.3 – Інтерфейс результату роботи системи композиції сервісів (змінений набір умов для сервісів)

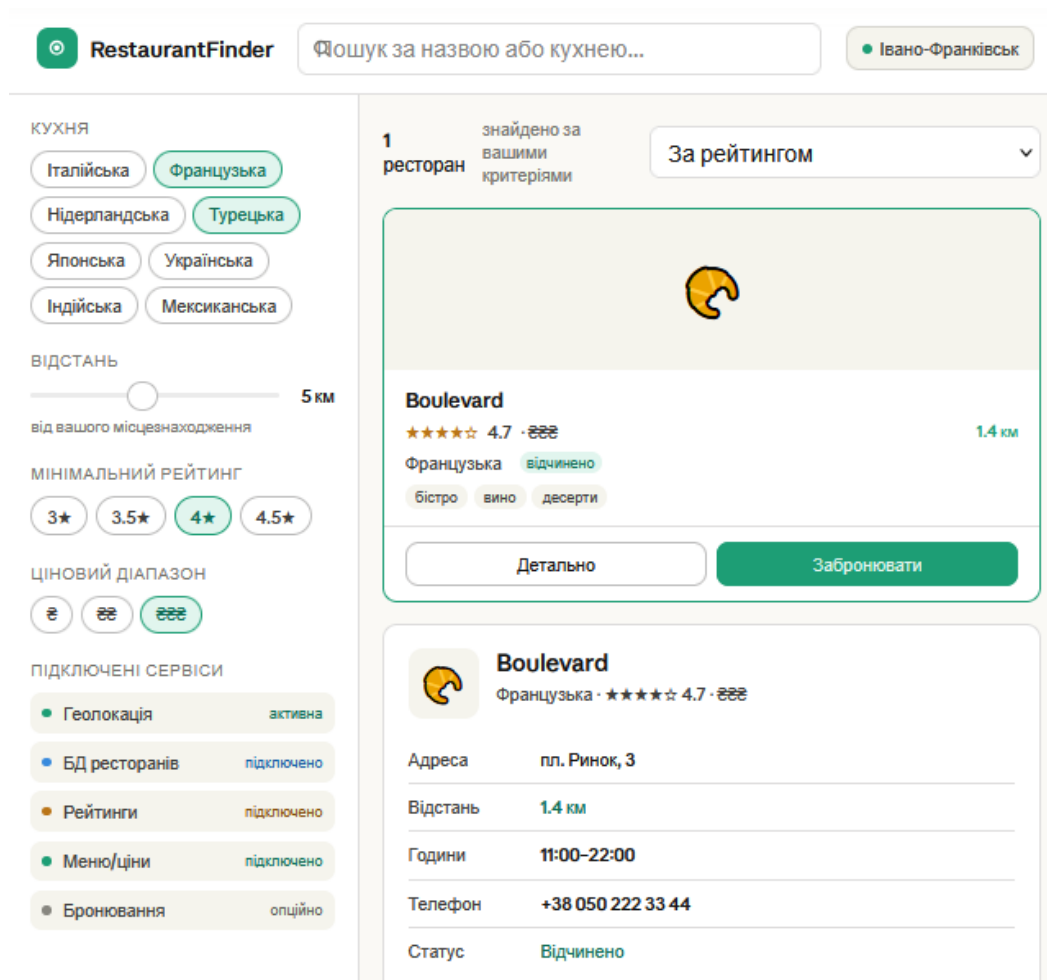


Рисунок 5.4 – Інтерфейс результату роботи системи композиції сервісів
(змінений набір умов для сервісів)

Наявність кнопки «Забронювати» підтверджує, що система пройшла валідацію вхідних даних і готова до виконання транзакційного сервісу бронювання.

Таким чином, представлений інтерфейс демонструє успішну реалізацію віртуальної композиції. Система не просто відображає статичний список, а динамічно конструює сервісний ланцюжок, де зміна будь-якого параметра на бічній панелі ініціює миттєву реконфігурацію всієї структури підключених сервісів. Це забезпечує високу адаптивність системи до контекстуальних вимог користувача.

					БР.ІП – 62.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		76

5.3 Висновки по розділу

У п'ятому розділі здійснено програмну реалізацію прототипу системи користувацько-орієнтованої композиції сервісів. Реалізований інтерфейс забезпечує інтеграцію сервісів із використанням механізмів адаптації та динамічної фільтрації. Проведений аналіз архітектури показав доцільність використання модульного підходу для забезпечення масштабованості системи. Підтверджено можливість практичного застосування розроблених моделей і методів у різних прикладних доменах. Отримані результати демонструють ефективність запропонованих рішень у підвищенні зручності та гнучкості взаємодії користувача із сервісами.

					БР.ІП – 62.00.00.000 ПЗ	Арк.
						77
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

В бакалаврській роботі досліджено моделі та методи формування користувацько-орієнтованих композицій сервісів і реалізовано прототип програмної системи пошуку ресторанів на основі сервісів.

У першому розділі здійснено ґрунтовний аналіз предметної області, що дозволив систематизувати сучасні підходи до реалізації користувацько-орієнтованої композиції сервісів. Встановлено, що класичні підходи до проєктування сервісних систем, засновані на жорстко визначених сценаріях, поступово втрачають актуальність у зв'язку зі зростанням динамічності середовища та різноманітністю користувацьких вимог. Обґрунтовано доцільність переходу від детермінованих моделей до адаптивних та контекстно-чутливих механізмів композиції.

У другому розділі проведено аналіз прикладних доменів, що дозволило визначити специфіку використання композиції сервісів у різних галузях, зокрема в корпоративних системах, домашній автоматизації, електронній комерції та e-Health. Запропоновано критерії оцінювання доменів з точки зору придатності до впровадження користувацько-орієнтованих підходів. Встановлено, що ключовими факторами ефективності є рівень стандартизації сервісів, наявність відкритих інтерфейсів, а також можливість інтерпретації контексту користувача. Проведений порівняльний аналіз продемонстрував, що найбільш складними є домени з високим рівнем критичності та регуляторних обмежень (зокрема e-Health), тоді як електронна комерція є більш сприятливим середовищем для впровадження адаптивних механізмів композиції.

У третьому розділі розроблено архітектурну модель прототипу системи користувацько-орієнтованої композиції сервісів. Запропонована архітектура базується на поєднанні семантичного моделювання, адаптивних інтерфейсів та механізмів міжпротокової взаємодії. Реалізовано концепцію сервісного

					БР.ІП – 62.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		78

ядра, яке забезпечує динамічне формування композицій на основі семантичних описів сервісів і запитів користувача. Важливим науковим результатом є розробка механізмів семантичної трансформації, що дозволяють автоматично генерувати інтерфейси введення та узгоджувати параметри викликів сервісів.

У четвертому розділі сформульовано функціональні вимоги до ядра системи та досліджено механізми управління контекстом. Запропоновано формалізовану модель контексту, що включає таксономію контекстної інформації та методи її актуалізації в архітектурі «клієнт–сервер». Розроблено механізми управління сесіями та забезпечення персистентності даних, що дозволяють підтримувати цілісність і узгодженість процесів композиції. Особливу увагу приділено специфікації протоколів взаємодії між інтерфейсним рівнем і середовищем оркестрації, включаючи процедури вирішення залежностей та верифікації вхідних параметрів.

У п'ятому розділі здійснено програмну реалізацію прототипу системи, що підтверджує практичну придатність запропонованих моделей і методів. Реалізований інтерфейс забезпечує інтеграцію сервісів із використанням механізмів динамічної фільтрації та адаптації до потреб користувача. Архітектурний аналіз показав, що застосування модульного підходу та семантичних моделей дозволяє підвищити масштабованість і гнучкість системи, а також зменшити складність взаємодії для кінцевого користувача.

Практичне значення роботи полягає у можливості застосування розробленого прототипу та запропонованих підходів у різних прикладних доменах, що потребують гнучкої інтеграції сервісів та персоналізації взаємодії, зокрема в корпоративних інформаційних системах, електронній комерції, системах «розумного дому» та e-Health.

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Smith J., Brown A. User-Centric Service Composition in Dynamic Web Environments // IEEE Transactions on Services Computing. – New York: IEEE, 2019. – Vol. 12, No. 3. – P. 345–358.
2. Chen L., Zhang Y. Semantic Web Services Composition: A Survey // ACM Computing Surveys. – New York: ACM, 2020. – Vol. 53, No. 2. – P. 1–38.
3. Kumar S., Gupta R. Context-Aware Service Composition for Smart Systems // Proceedings of the International Conference on Web Services (ICWS). – San Francisco: IEEE, 2018. – P. 210–217.
4. Garcia P., Molina J. Adaptive Mashup Platforms for End Users // Journal of Systems and Software. – Amsterdam: Elsevier, 2017. – Vol. 131. – P. 45–60.
5. Wilson D., Taylor M. Service-Oriented Architecture: Concepts and Applications // International Journal of Information Systems. – London: Springer, 2016. – Vol. 10, No. 4. – P. 255–270.
6. Nguyen T., Lee K. Dynamic Composition of Web Services Using AI Planning // IEEE Access. – New York: IEEE, 2021. – Vol. 9. – P. 112345–112356.
7. Rossi F., Bianchi G. User Modeling in Adaptive Systems // Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI). – New York: ACM, 2019. – P. 123–134.
8. Patel H., Shah D. Intelligent Interfaces for Service Integration // Journal of Web Engineering. – Berlin: Rinton Press, 2018. – Vol. 17, No. 5. – P. 401–420.
9. Brown T., Davis R. Mashups and Lightweight Integration Technologies // IEEE Internet Computing. – New York: IEEE, 2017. – Vol. 21, No. 2. – P. 60–67.

					БР.ІІІ – 62.00.00.000 ІІЗ	Арк.
						80
Змн.	Арк.	№ докум.	Підпис	Дата		

10. Kim S., Park J. Service Composition in Cloud Computing Environments // Proceedings of the International Conference on Cloud Computing (CLOUD). – Honolulu: IEEE, 2019. – P. 89–96.
11. Lopez M., Garcia E. Semantic Annotation for Web Services // Journal of Web Semantics. – Amsterdam: Elsevier, 2018. – Vol. 51. – P. 12–25.
12. Ahmed N., Rahman M. User-Centered Design for Service-Oriented Systems // International Journal of Human-Computer Studies. – London: Elsevier, 2021. – Vol. 148. – P. 102–120.
13. Zhang Q., Li H. Automated Service Composition Using Ontologies // Expert Systems with Applications. – Amsterdam: Elsevier, 2019. – Vol. 120. – P. 45–58.
14. Green P., White S. Adaptive Interfaces in Web Applications // Proceedings of the WWW Conference. – Lyon: ACM, 2018. – P. 567–576.
15. Petrov I., Kuznetsov V. Context-Aware Computing in Smart Homes // Sensors. – Basel: MDPI, 2020. – Vol. 20, No. 15. – P. 4201–4215.
16. Baker J., Collins D. Service Integration Patterns in Enterprise Systems // Software: Practice and Experience. – London: Wiley, 2017. – Vol. 47, No. 6. – P. 789–805.
17. Silva R., Costa P. User Profiles and Personalization in Web Systems // Information Processing & Management. – Amsterdam: Elsevier, 2021. – Vol. 58, No. 3. – P. 102–118.
18. Wang X., Zhou Y. QoS-Aware Service Composition // IEEE Transactions on Parallel and Distributed Systems. – New York: IEEE, 2019. – Vol. 30, No. 6. – P. 1234–1247.
19. Novak D., Horvat M. Service-Oriented Computing in e-Health Systems // Journal of Biomedical Informatics. – Amsterdam: Elsevier, 2020. – Vol. 108. – P. 103–115.
20. Chen X., Huang Z. Dynamic Workflow Composition in SOA // Proceedings of ICSOC. – Paris: Springer, 2018. – P. 210–224.

					БР.ІІІ – 62.00.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		81

21. Lee S., Kim H. Context-Aware Middleware for Service Composition // Future Internet. – Basel: MDPI, 2019. – Vol. 11, No. 4. – P. 89–101.
22. Brown L., Scott P. End-User Development in Mashup Environments // IEEE Software. – New York: IEEE, 2017. – Vol. 34, No. 3. – P. 70–77.
23. Taylor J., Moore K. Adaptive Systems and Context Awareness // Artificial Intelligence Review. – Dordrecht: Springer, 2019. – Vol. 52, No. 2. – P. 789–812.
24. Park E., Cho Y. Intelligent Service Discovery Mechanisms // Information Sciences. – Amsterdam: Elsevier, 2018. – Vol. 432. – P. 45–60.
25. Smith R., Allen T. Data Integration Techniques in SOA // Journal of Data and Information Quality. – New York: ACM, 2017. – Vol. 9, No. 1. – P. 1–18.
26. Garcia L., Perez J. Human-Centered Service Design // Design Studies. – Amsterdam: Elsevier, 2020. – Vol. 68. – P. 45–62.
27. Li F., Chen G. Workflow Automation in Service Systems // Proceedings of the IEEE International Conference on Automation Science and Engineering. – Munich: IEEE, 2019. – P. 312–319.
28. Novak P., Urbanec J. Service Composition in Smart Cities // Sustainable Cities and Society. – Amsterdam: Elsevier, 2021. – Vol. 65. – P. 102–118.
29. Ahmed S., Khan R. AI-Based Service Recommendation Systems // Expert Systems. – London: Wiley, 2020. – Vol. 37, No. 4. – P. 1–12.
30. Brown C., Evans J. RESTful Web Services and Integration // IEEE Internet Computing. – New York: IEEE, 2018. – Vol. 22, No. 1. – P. 34–42.
31. Petrov D., Ivanenko K. Context Modeling Techniques in Distributed Systems // Computing. – Vienna: Springer, 2019. – Vol. 101, No. 5. – P. 567–580.
32. Wang L., Sun J. Service Composition Optimization Techniques // Journal of Network and Computer Applications. – Amsterdam: Elsevier, 2020. – Vol. 150. – P. 102–115.

					БР.ІІІ – 62.00.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		82

33. Kim Y., Lee J. User Experience in Service-Oriented Applications // International Journal of Human-Computer Interaction. – London: Taylor & Francis, 2019. – Vol. 35, No. 8. – P. 678–690.

34. Lopez A., Martin S. Semantic-Based Mashup Tools // Proceedings of the European Conference on Web Engineering (ICWE). – Lugano: Springer, 2018. – P. 145–159.

					БР.ІП – 62.00.00.000 ПЗ	Арк.
						83
Змн.	Арк.	№ докум.	Підпис	Дата		

БІБЛІОГРАФІЧНА ДОВІДКА

Тема бакалаврської роботи: “Моделі та методи формування користувацько-орієнтованих композицій сервісів ”

Обсяг пояснювальної записки: 83 аркуші.

Дата закінчення роботи: 9 червня 2026 р.

Підпис студента _____