

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ  
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет/(ННІ) Інформаційних технологій

**ПОГОДЖЕНО**

Декан факультету (Директор ННІ)

\_\_\_\_\_ (назва факультету (ННІ))

Ігор Болбот

(підпис)

(ім'я ПРИЗВИЩЕ)

“ ” \_\_\_\_\_ 20\_\_ р.

**ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ**

Завідувач кафедри

\_\_\_\_\_ (назва кафедри)

Белла Голуб

(підпис)

(ім'я ПРИЗВИЩЕ)

“ ” \_\_\_\_\_ 20\_\_ р.

**МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА**

на тему **Програмне забезпечення для моделювання, оброблення та аналізу тривимірних об'єктів**

Спеціальність 121 Інженерія програмного забезпечення

(код і найменування)

Освітня програма Програмне забезпечення інформаційних систем

(назва)

Орієнтація освітньої програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

**Гарант освітньої програми**

доц.к.ф.-м.н.

(науковий ступінь та вчене звання)

Віктор КИРИЧЕНКО

(підпис)

(ім'я ПРИЗВИЩЕ)

**Керівник магістерської кваліфікаційної роботи**

ст. викладач

(науковий ступінь та вчене звання)

Віктор ПАНКРАТЬСВ

(підпис)

(ім'я ПРИЗВИЩЕ)

к.т.н., доцент

(науковий ступінь та вчене звання)

Олексій ТКАЧЕНКО

(підпис)

(ім'я ПРИЗВИЩЕ)

**Виконав**

(підпис)

Роман ТИМОЦУК

(ім'я ПРИЗВИЩЕ здобувача)

КИЇВ – 2025

# НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет (ННІ) інформаційних технологій

**ЗАТВЕРДЖУЮ**

**Завідувач кафедри** комп'ютерних наук  
доцент, к.т.н. Голуб Б. Л.  
(науковий ступінь, вчене звання) (підпис) (ПІБ)  
“10” жовтня 2025 року

## З А В Д А Н Н Я

### ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ СТУДЕНТУ

Тимошук Роман Олександрович

(прізвище, ім'я, по батькові)

Спеціальність 121 «Інженерія програмного забезпечення»

(код і назва)

Освітня програма Програмне забезпечення інформаційних систем

(назва)

Орієнтація освітньої програми освітньо-професійна

Тема магістерської кваліфікаційної роботи Програмне забезпечення для моделювання, оброблення та аналізу тривимірних об'єктів

затверджена наказом ректора НУБіП України від “10” жовтня 2025р. №2291 «С» на часткову зміну до наказу від 01.11.2024 №1963

Термін подання завершеної роботи на кафедру 20.11.2025

(рік, місяць, число)

Вихідні дані до магістерської кваліфікаційної роботи: Платформи та інструменти для розробки додатків тривимірного представлення.

Перелік питань, що підлягають дослідженню:

1. Аналіз сучасних підходів до використання в технічних науках.
2. Визначення технічних та програмних засобів для реалізації 3D-системи.
3. Розроблення прототипу 3D-системи для конкретного практичного кейсу.
4. Оцінка ефективності використання 3D у вибраній сфері технічних наук.

Перелік графічного матеріалу (за потреби)

Дата видачі завдання “10” жовтня 2025 р.

Керівник магістерської кваліфікаційної роботи \_\_\_\_\_

(підпис)

Панкратьєв В.О.

(прізвище та ініціали)

—

Ткаченко О.М.

(підпис)

(прізвище та ініціали)

Завдання прийняв до виконання \_\_\_\_\_

(підпис)

Тимошук Р.О.

(прізвище та ініціали студента)

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ .....	5
ВСТУП .....	7
1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	9
1.1 Опис предметної області .....	9
1.2 Теоретико-методологічні засади та стан наукових досліджень.....	11
1.3 Аналіз сучасних 3D-технологій і систем .....	15
1.4 Моделювання предметної області .....	20
1.5 Аналіз вимог програмної системи.....	24
1.6 Постановка завдання.....	26
1.7 Висновки до першого розділу.....	27
2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	30
2.1 Логічна модель даних у вигляді ER-діаграми.....	30
2.2 Діаграма класів та кооперації .....	31
2.3 Діаграма компонентів .....	35
2.4 Діаграма пакетів .....	37
2.5 Висновки до другого розділу .....	39
3 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ....	41
3.1 Вибір технологій та інструментальних засобів реалізації системи .....	41
3.2 Фізична модель даних і структура OLAP-кубу.....	43
3.3 Архітектура системи 3D-візуалізації .....	46

3.5 Висновки до третього розділу.....	49
4 ТЕСТУВАННЯ ТА ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ АНАЛІТИЧНОЇ СИСТЕМИ .....	51
4.1 План тестування програмних модулів та методика оцінювання результатів.....	51
4.2 Тестування інтелектуальної системи моделювання, оброблення та аналізу тривимірних об'єктів .....	53
4.3 Результати тестування та аналіз ефективності системи .....	56
4.4 Висновки до четвертого розділу.....	58
ВИСНОВКИ.....	60
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	62
ДОДАТОК А.....	64

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

1. AR — augmented reality, доповнена реальність
2. MR — mixed reality, змішана реальність
3. VR — virtual reality, віртуальна реальність
4. 3D — тривимірне геометричне подання об'єктів
5. FPS — frames per second, частота відтворення кадрів
6. VIO — visual-inertial odometry, візуально-інерційна одометрія
7. PnP — Perspective-n-Point, алгоритм визначення пози за ключовими точками
8. EPnP — Efficient PnP, оптимізований метод оцінювання поз
9. IMU — inertial measurement unit, інерціальна вимірювальна система
10. SLAM — simultaneous localization and mapping, одночасна локалізація і побудова карти
11. ORB — Oriented FAST and Rotated BRIEF, детектор та дескриптор ключових точок
12. AKAZE — Accelerated KAZE, алгоритм екстракції ключових точок
13. ArUco — маркерна система трекінгу для комп'ютерного зору
14. JWT — JSON Web Token, токен авторизації
15. SSO — single sign-on, єдиний вхід у систему
16. OIDC — OpenID Connect, протокол автентифікації
17. API — application programming interface, інтерфейс програмної взаємодії
18. SDK — software development kit, набір засобів розробника
19. GUI — graphical user interface, графічний інтерфейс користувача
20. UI — user interface, інтерфейс користувача
21. SQLite — реляційна вбудована система керування базами даних
22. ORM — object-relational mapping, об'єктно-реляційне відображення
23. PBR — physically based rendering, фізично коректний рендеринг

24. KPI — key performance indicators, ключові показники ефективності
25. Latency — затримка між обробкою кадру та його відтворенням
26. TrackingEngine — підсистема трекінгу AR-сцени
27. MathCore — обчислювальне ядро параметричних і аналітичних моделей
28. Renderer — модуль рендерингу тривимірних об'єктів
29. SceneView — вікно відображення AR-сцени в PyQt6
30. Telemetry — телеметричні дані системи (FPS, latency, track-loss, події)
31. Session — сесія роботи користувача в AR-додатку
32. RuleRegistry — модуль правил валідації кадрів і сцен

## ВСТУП

с

**Метою роботи** є розроблення програмного забезпечення для моделювання, оброблення та аналізу тривимірних об'єктів, що забезпечує інтеграцію геометричних алгоритмів, методів чисельної оптимізації, засобів візуалізації та інтелектуального аналізу даних.

**Завдання дослідження** включають:

- аналіз предметної області тривимірного моделювання та існуючих рішень;
- формування вимог до програмної системи;
- побудову архітектури та моделей даних;
- розроблення модулів оброблення 3D-геометрії (фільтрація, нормалізація, вирівнювання, сегментація);
- реалізацію алгоритмів аналізу тривимірних об'єктів (кластеризація, статистичні метрики, геометричні дескриптори);
- створення програмного інтерфейсу для інтерактивної роботи з моделями;
- тестування й оцінювання продуктивності системи.

**Об'єктом дослідження** є процес комп'ютерного моделювання та структурного аналізу тривимірних об'єктів у цифрових системах.

**Предметом дослідження** є методи, алгоритми, архітектурні рішення та програмні засоби, що забезпечують побудову, перетворення, оброблення та аналітичний аналіз тривимірних об'єктів різних типів.

**Методи дослідження** включають алгоритми цифрової обробки геометрії, чисельні методи оптимізації, методи машинного навчання для аналізу структурних характеристик, статистичні методи оцінювання, а також архітектурні підходи до проектування програмних систем.

**Наукова новизна** полягає у формуванні інтегрованого підходу до оброблення та аналізу тривимірних об'єктів, який поєднує алгоритмічні методи

геометричної реконструкції, машинне навчання для структурної класифікації та архітектуру модульної системи, орієнтованої на високоточне моделювання та масштабовану аналітику. Додатковою новизною є адаптація процедур попередньої підготовки 3D-даних до змішаних типів геометричних репрезентацій та інтеграція засобів інтелектуального аналізу для автоматизованих дослідницьких сценаріїв.

# 1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Опис предметної області

Предметна область охоплює процеси побудови, оброблення та аналізу тривимірних об'єктів у цифрових інженерних, наукових і виробничих системах. Сучасні задачі 3D-моделювання включають створення складних геометричних структур, їхнє редагування, оптимізацію сіткових представлень, сегментацію, фільтрацію, реконструкцію та подальший аналіз за числовими, топологічними і просторовими метриками [1]. В основі таких операцій лежить використання різних типів геометричних формалізацій - трикутних та полігональних сіток, параметричних поверхонь (NURBS), воксельних структур або хмар точок, що потребують алгоритмічно різних підходів для забезпечення коректної обробки та високої точності. На рисунку 1.1 наведено узагальнену модель потоків даних, що демонструє ключові підсистеми програмного забезпечення та їх взаємодію у процесі роботи з тривимірними об'єктами.

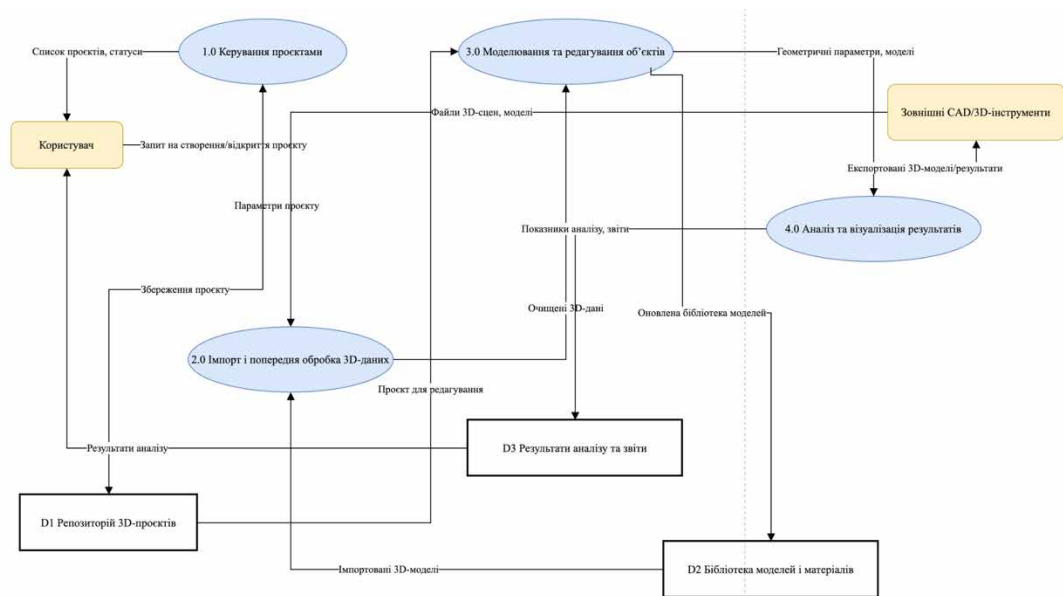


Рис. 1.1. Узагальнена схема взаємодії модулів системи у процесі 3D-моделювання, оброблення та аналізу даних

Предметна область включає декілька структурних компонентів. Першим із них є керування проектами, що передбачає створення, відкриття, зберігання та

організацію 3D-сцен з урахуванням їх метаданих, станів оброблення та історії змін. Наступним етапом є імпорт 3D-даних та їх попередня обробка, яка охоплює нормалізацію геометрії, видалення шумів, реконструкцію поверхонь, вирівнювання моделей і приведення їх до уніфікованого формату, що забезпечує коректне подальше редагування. Для більшості прикладних задач використовуються формати STL, OBJ, PLY, GLTF, а також дані, отримані зі 3D-сканерів або фотограмметрії.

Центральну роль у предметній області відіграє моделювання та редагування 3D-об'єктів, що включає операції трансформації, модифікації сітки, зміни топології, параметричної реконструкції, накладання матеріалів та формування складних сцен. Ці дії часто опрацьовуються у реальному часі і вимагають високої продуктивності графічного рушія та ефективних алгоритмів обчислень. Завершальним елементом є модуль аналізу та візуалізації результатів, який забезпечує обчислення структурних показників, просторових характеристик, площ, об'ємів, геометричних дескрипторів, кластеризацію та інші інтелектуальні процедури. Зведені результати аналізу можуть бути представлені у вигляді таблиць, графіків або звітів. Приклад структурованого уявлення основних сутностей предметної області наведено у таблиці 1.1.

Таблиця 1.1

#### Основні сутності та характеристики предметної області 3D-моделювання

Сутність	Характеристика
3D-модель	Геометричне представлення об'єкта: сітка, NURBS, вокселі, хмара точок
Матеріали і текстури	Фізичні та візуальні властивості поверхонь
Сцена	Сукупність моделей, світла, камер, параметрів рендерингу
Проект	Структуроване сховище файлів, параметрів, оброблених даних
Аналітичні показники	Об'єм, площа, кривизна, топологічні метрики, кластеризація
Інструменти оброблення	Фільтри, реконструкція, вирівнювання, сегментація, оптимізація

Предметна область охоплює як геометричну складову моделювання, так і алгоритмічні процеси оброблення та аналізу даних, що забезпечують створення повноцінного інструменту для роботи з тривимірними об'єктами. Вона вимагає поєднання інженерних, математичних та програмних методів, включаючи алгоритми комп'ютерної графіки, чисельної геометрії, цифрової реконструкції та аналітики великих 3D-структур.

## 1.2 Теоретико-методологічні засади та стан наукових досліджень

Теоретичною основою сучасних систем моделювання та аналізу тривимірних об'єктів є комплекс алгоритмів обчислювальної геометрії, методів реконструкції поверхонь, спектральних і статистичних дескрипторів, а також глибоких нейронних мереж, що забезпечують інваріантне представлення форми. Методологія реконструкції поверхні з багатовидових глибинних карт широко представлена в роботах К. Wolff та співавт., де введено поєднання фотоконсистентності та геометричних оцінок для вибору коректних точок реконструйованої поверхні. На рисунку 1.2 наведено узагальнену схему цього підходу, що відображає процес проєкції множини глибинних карт, визначення точок перетину та подальшого агрегаційного оцінювання якості реконструкції.

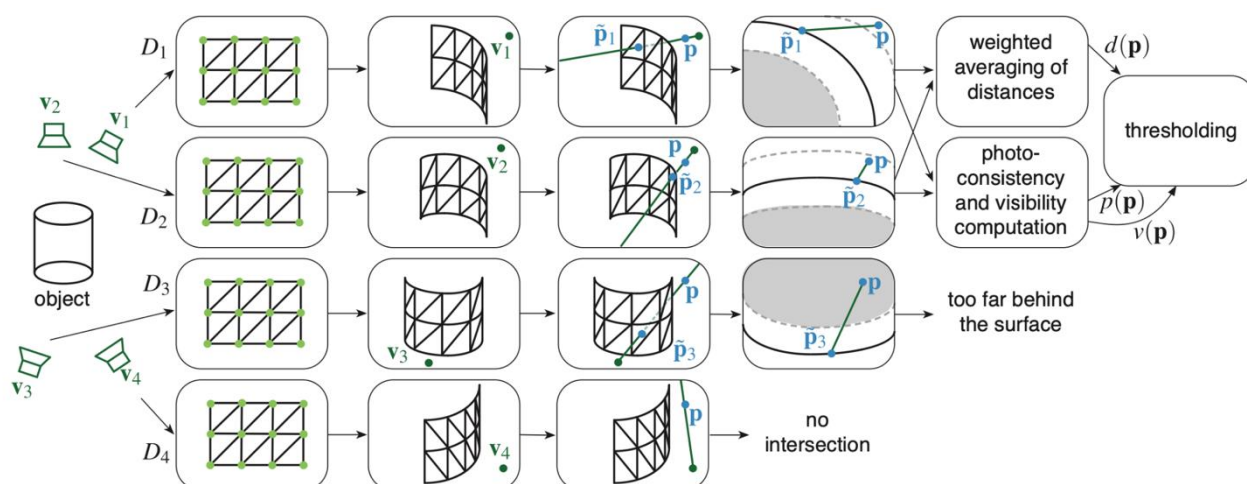


Рис. 1.2. Схема агрегування багатовидових глибинних карт із застосуванням фотоконсистентності (за Wolff et al.)

Проблема шумів і артефактів у хмарах точок залишається центральною для всіх методів реконструкції. У роботі Li et al. запропоновано метод нелокальної низькорівневої фільтрації нормалей, що групує подібні локальні патчі поверхні та виконує low-rank апроксимацію, значно підвищуючи гладкість геометрії. На рисунку 1.3 показано порівняння «нефільтрованих» моделей із результатами після застосування фільтрації, де помітне суттєве зменшення шуму та відновлення структурних контурів.

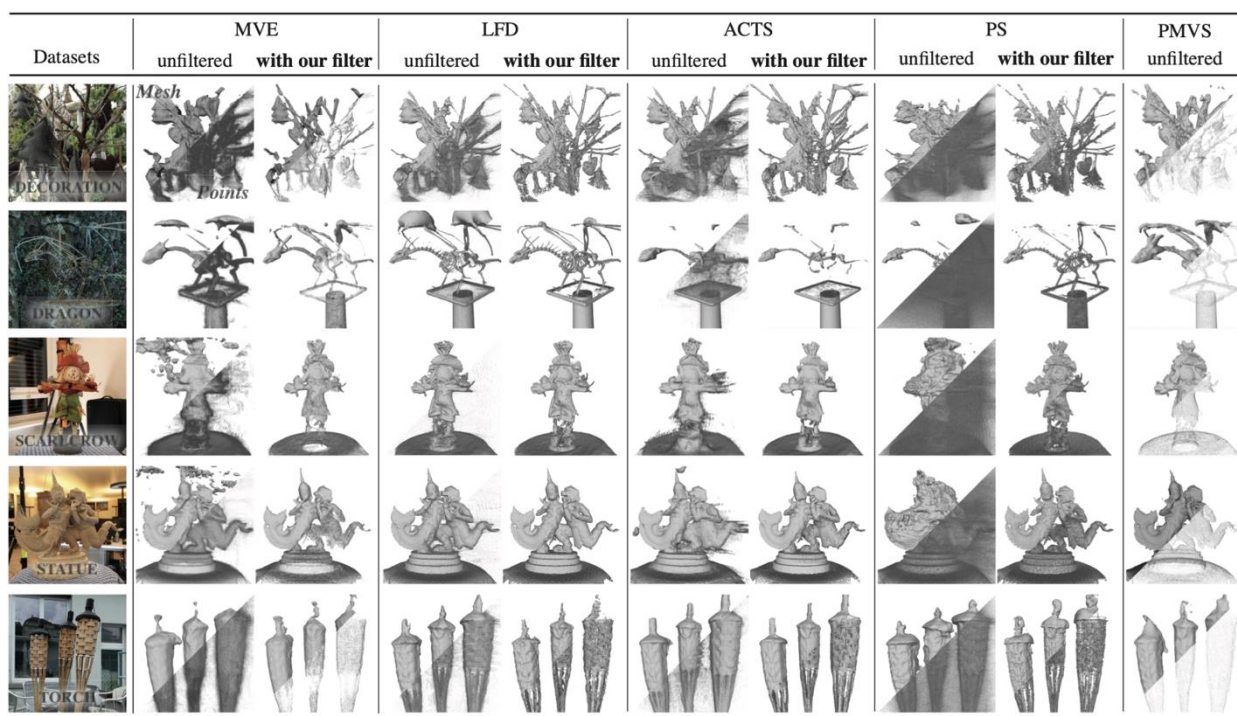


Рис. 1.3. Порівняння результатів реконструкції без фільтрації та після застосування low-rank фільтра нормалей (за Li et al.)

Додатково Li та співавт. демонструють, що якість реконструкції може бути подальше підсилена поєднанням фільтрації з алгоритмами Poisson Surface Reconstruction (PSR). На рисунку 1.4 наведено приклади покращення точності (Acc.) та повноти (Comp.) для різних методів реконструкції після інтеграції low-rank фільтрації, що підтверджує ефективність попередньої нормалізації даних перед побудовою поверхні.




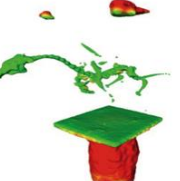

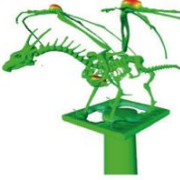



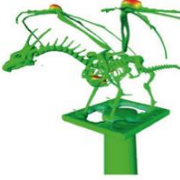


	MVE	LFD	ACTS	PS
<b>PSR</b>				
	Acc.: 0.0953 Comp.: 64.4	Acc.: 0.0374 Comp.: 85.6	Acc.: 0.0373 Comp.: 38.4	Acc.: 0.2107 Comp.: 50.5
<i>Ground truth</i>				
<b>Ours + PSR</b>				
	Acc.: <b>0.0647</b> Comp.: <b>74.0</b>	Acc.: <b>0.0218</b> Comp.: <b>91.6</b>	Acc.: <b>0.0244</b> Comp.: <b>47.3</b>	Acc.: <b>0.0385</b> Comp.: <b>86.0</b>

Рис. 1.4. Порівняння точності й повноти моделей до та після інтегрованої фільтрації нормалей (за Li et al.)

У контексті масштабованості та обчислювальної вартості реконструкції значний внесок зроблено в роботах Goesele et al., де досліджено залежність точності реконструкції від кількості глибинних карт. На рисунку 1.5 показано зміну похибки, повноти та часу виконання залежно від обсягу вхідних даних. У роботі доведено, що лінійне збільшення кількості карт не завжди корелює з покращенням точності, що визначає необхідність адаптивного відбору вхідних структур.

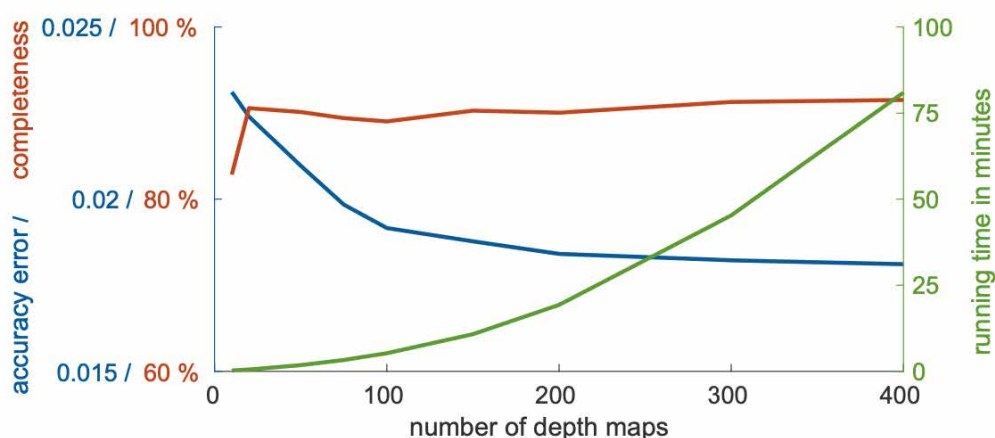


Рис. 1.5. Взаємозв'язок точності, повноти та часу реконструкції залежно від кількості depth-maps (за Goesele et al.)

Окремим напрямом досліджень є побудова дескрипторів форми, інваріантних до деформацій, поворотів і шумистості. Fang et al. розробили глибокий дескриптор форми (3D Deep Shape Descriptor), що базується на структурних ознаках Heat Kernel Signature та Heat Shape Descriptor, інтегрованих у двоступеневий екодер. На рисунку 1.6 наведено архітектуру формування дескриптора, яка демонструє застосування глибокого навчання для отримання багатовимірного, інваріантного та стабільного представлення складних 3D-структур.

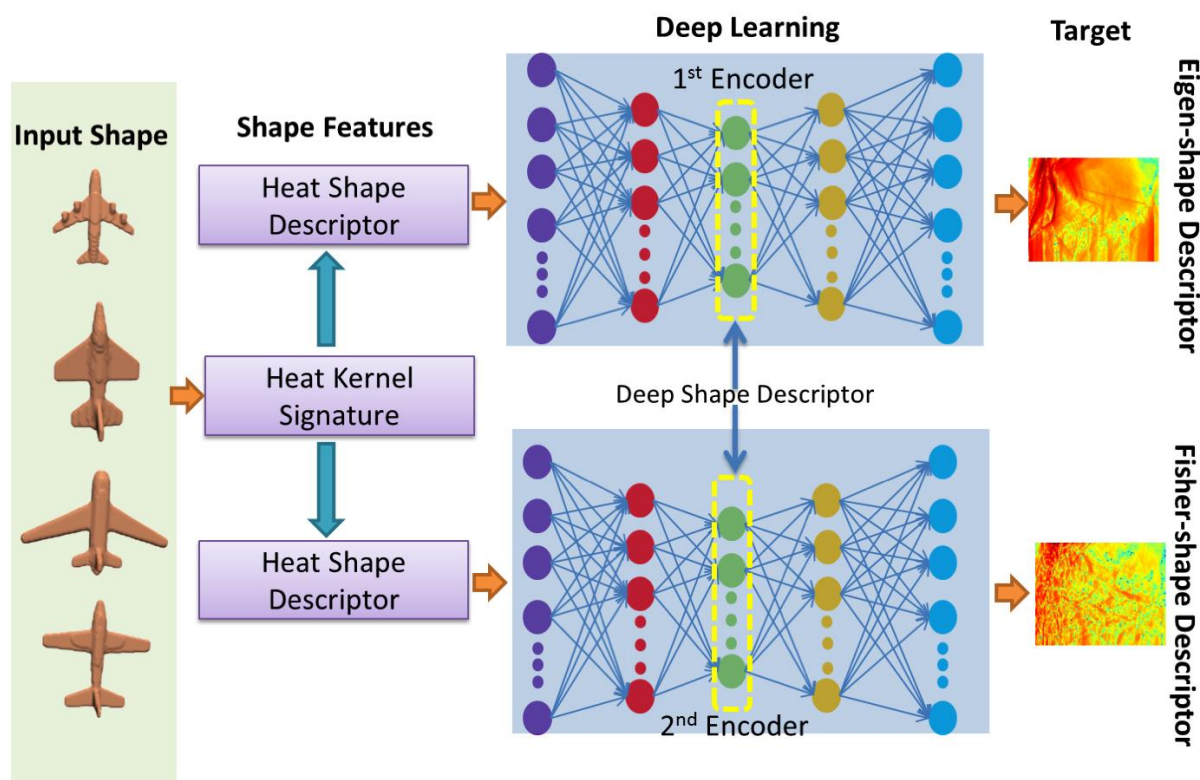


Рис. 1.6. Архітектура формування глибокого дескриптора форми (за Fang et al.)

Узагальнюючи викладене, можна стверджувати, що сучасні наукові дослідження зосереджені на окремих аспектах:

- зменшення шумів і реконструкція поверхні (Wolff et al., Li et al.);
- комбінування інформації з багатьох глибинних карт (Goesele et al.);
- побудова інваріантних дескрипторів форми на основі глибокого навчання (Fang et al.).

Однак більшість підходів або працюють у жорстко визначених середовищах, або орієнтовані на вузькі задачі — реконструкцію, класифікацію чи фільтрацію. Не досліджено інтегровані системи, які одночасно виконують:

- імпорт і попередню геометричну нормалізацію 3D-даних різних типів;
- моделювання та редагування 3D-об'єктів у єдиному середовищі;
- аналітичну обробку, кластеризацію й порівняльні оцінки структурних метрик;
- глибоке представлення форми для подальшої оптимізації моделювання.

Саме усунення цієї наукової прогалини й становить наукову новизну даної роботи: розроблено інтегрований підхід, який об'єднує алгоритми передобробки, реконструкції та глибинного аналізу тривимірних об'єктів у єдиному модульному програмному середовищі, що забезпечує підвищену точність геометричної інтерпретації, стабільність дескрипторів та можливість масштабованої аналітики для складних 3D-сцен.

### **1.3 Аналіз сучасних 3D-технологій і систем**

Сучасні програмні засоби для моделювання, оброблення та аналізу тривимірних об'єктів демонструють значну функціональну різноманітність, однак більшість із них орієнтовано на окремі сегменти: художнє моделювання, інженерну розробку, скульптування, оброблення хмар точок або підготовку моделей до 3D-друку. На рисунку 1.7 наведено приклад середовища Blender, яке представляє універсальний інструментарій для полігонального моделювання, скульптингу та анімації. Blender забезпечує високу деталізацію геометрії, наявність редакторів матеріалів і фізично коректного рендерингу, однак комплексні засоби аналітики геометричних структур у ньому обмежені.

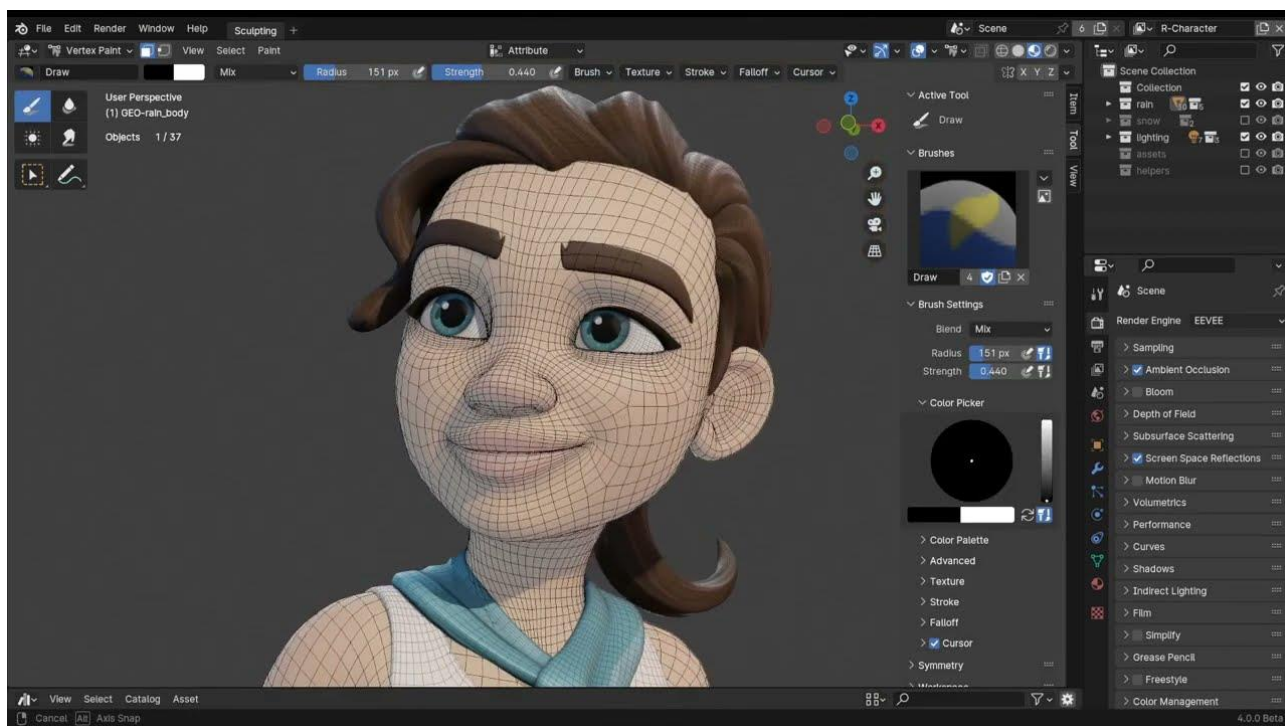


Рис. 1.7. Інтерфейс Blender під час полігонального редагування персонажа

На рисунку 1.8 зображено роботу MeshLab - відкритої платформи для оброблення хмар точок та трикутних сіток. MeshLab підтримує фільтрацію шумів, реконструкцію поверхонь, нормалізацію та топологічний аналіз. Його сильна сторона - широкий набір геометричних алгоритмів, проте відсутність централізованого керування проектами та слабкі інтерактивні редактори обмежують застосування у складних виробничих сценаріях.

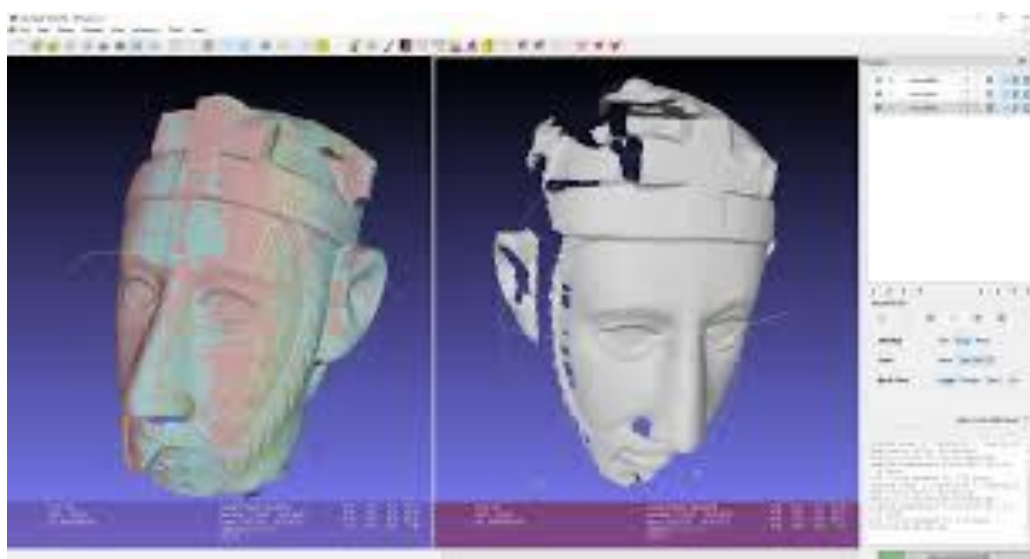


Рис. 1.8. Робота MeshLab під час очищення та реконструкції 3D-моделі обличчя

Система Autodesk Meshmixer (рис. 1.9) орієнтована на швидке та інтуїтивне редагування моделей, зокрема перед 3D-друком: вирівнювання, закриття отворів, сегментація, хутке редагування топології. Попри зручність, Meshmixer не підтримує повноцінні аналітичні алгоритми, кластеризацію 3D-даних чи інтелектуальне оцінювання структури моделі.

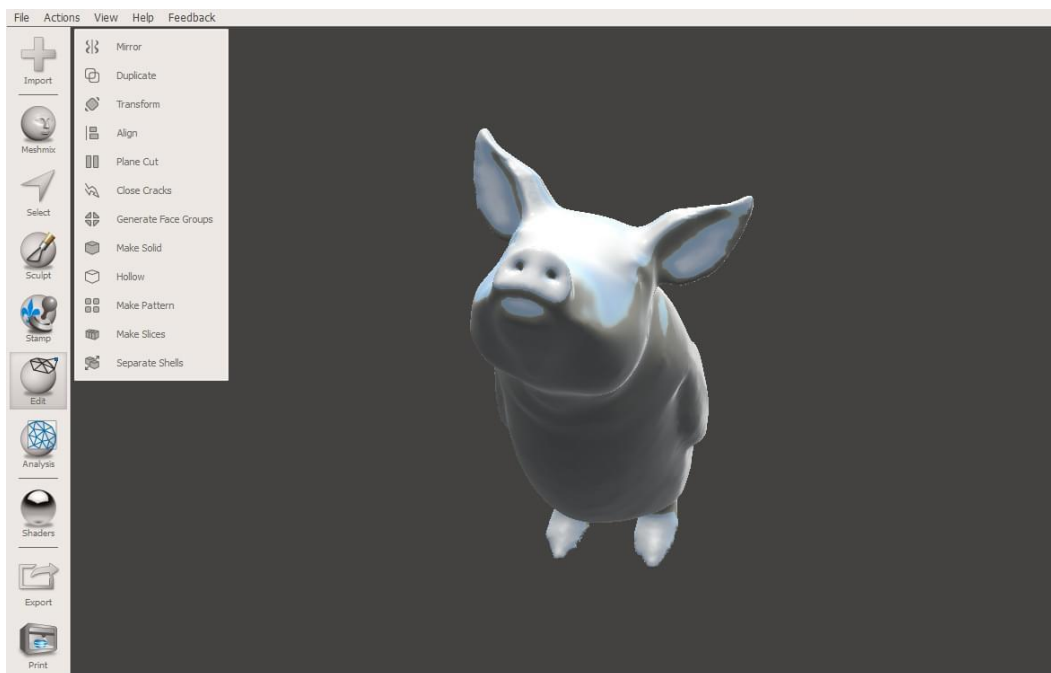


Рис. 1.9. Інтерфейс Autodesk Meshmixer під час роботи зі скульптурною моделлю

CloudCompare (рис. 1.10) є фаховим інструментом для роботи з великими хмарами точок, що містять мільйони елементів. Він забезпечує порівняння моделей, морфологічний аналіз, статистичні оцінки, обчислення кривини та різноманітні геометричні фільтри. Проте CloudCompare майже не включає засобів інтерактивного моделювання та слабо інтегрується з редакторами полігональних сіток.

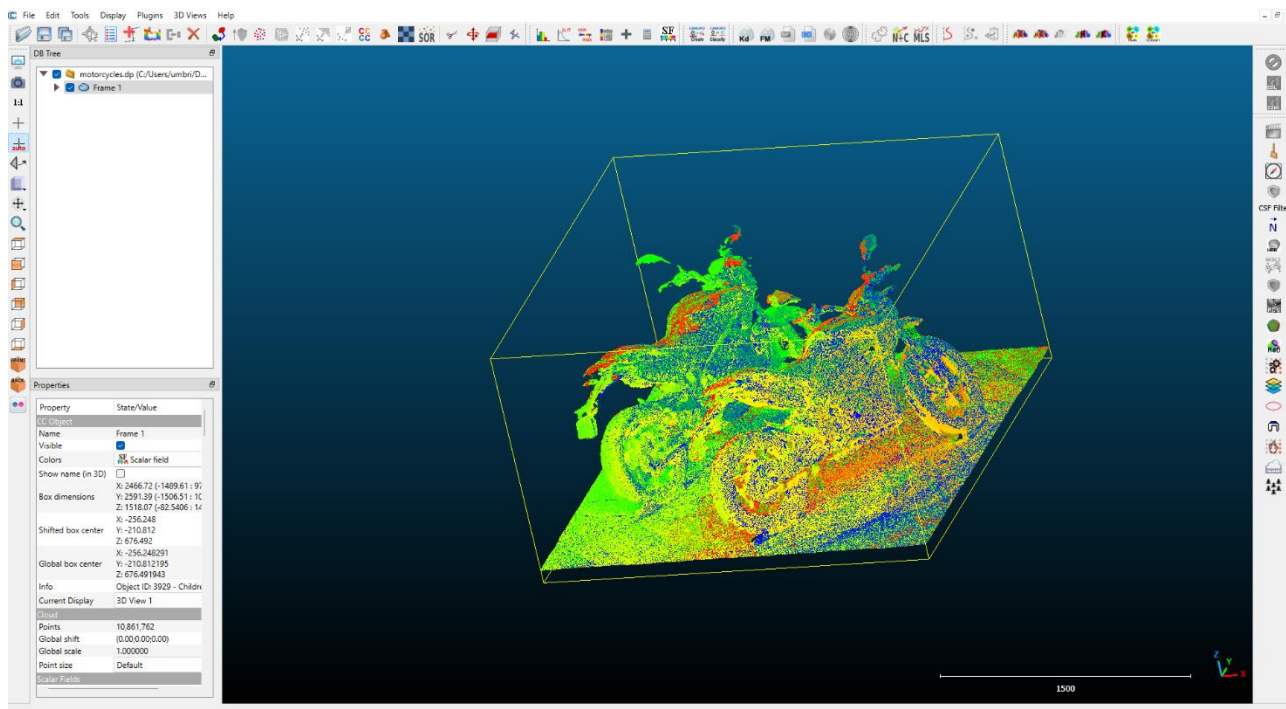


Рис. 1.10. Аналіз великих хмар точок у CloudCompare

На рисунку 1.11 представлено Autodesk Maya - професійне рішення для полігонального та NURBS-моделювання, анімації та візуальних ефектів. Maya має широкі можливості параметричного редагування та рендерингу, але не орієнтована на аналіз 3D-даних або дослідження структурної складності моделей.



Рис. 1.11. Полігональне моделювання обличчя в Autodesk Maya

Для системного зіставлення можливостей існуючих рішень зі специфічними вимогами нашої роботи сформовано таблицю 1.2, у якій наведено порівняння основних характеристик, функціональних модулів та обмежень.

Наша система включена до порівняння як окремий рядок, що дозволяє виділити ключові переваги: модульність архітектури, інтеграцію геометричної нормалізації з машинним навчанням, наявність аналітичного ядра для кластеризації, побудову дескрипторів форми та зручне керування проєктами.

Таблиця 1.2

Порівняльна характеристика існуючих систем і запропонованого рішення

Система	Моделювання	Обробка хмар точок	Аналітика 3D-даних	Керування проєктами	Інтеграція ML	Призначення
Blender	+	–	–	–	–	художнє моделювання
MeshLab	–	+	частково	–	–	наукова обробка сіток
Meshmixer	частково	–	–	–	–	підготовка до 3D-друку
CloudCompare	–	+	+	–	–	аналіз хмар точок
Autodesk Maya	+	–	–	–	–	професійне моделювання
Наша система	+	+	+	+	+	інтегроване 3D-моделювання, обробка й аналіз

Узагальнюючи проведений аналіз, можна стверджувати, що попри значну кількість програмних інструментів для роботи з тривимірними моделями, жоден з них не забезпечує комплексної інтеграції повного циклу: від імпорту та нормалізації 3D-даних до їх моделювання, редагування та глибинного аналітичного оцінювання. Більшість існуючих систем працюють із окремими етапами процесу - моделюванням, очисткою чи візуалізацією - та не мають уніфікованої аналітичної платформи, що дозволяє застосовувати методи

кластеризації, статистичні дескриптори та алгоритми машинного навчання до тривимірних структур. Запропоноване рішення усуває цю науково-практичну прогалину, забезпечуючи цілісний інструментарій для моделювання, оброблення та аналізу 3D-об'єктів у єдиному середовищі.

## **1.4 Моделювання предметної області**

Моделювання предметної області системи моделювання, оброблення та аналізу тривимірних об'єктів забезпечує формалізоване представлення функціональних можливостей, взаємодії акторів із програмним середовищем, а також послідовності завдань, що виконуються у межах життєвого циклу 3D-проєкту. На рис. 1.12 подано діаграму прецедентів, яка демонструє основних користувачів системи - інженера-моделювальника, аналітика, адміністратора та зовнішню CAD/3D-платформу - і перелік ключових прецедентів, включно з керуванням 3D-проєктами, імпортом 3D-даних, налаштуванням матеріалів та освітлення, виконанням геометричного аналізу, візуалізацією сцени й експортом результатів. Діаграма відображає логічні залежності «include» та «extend», що визначають структуру та обов'язковість функціональних сценаріїв.

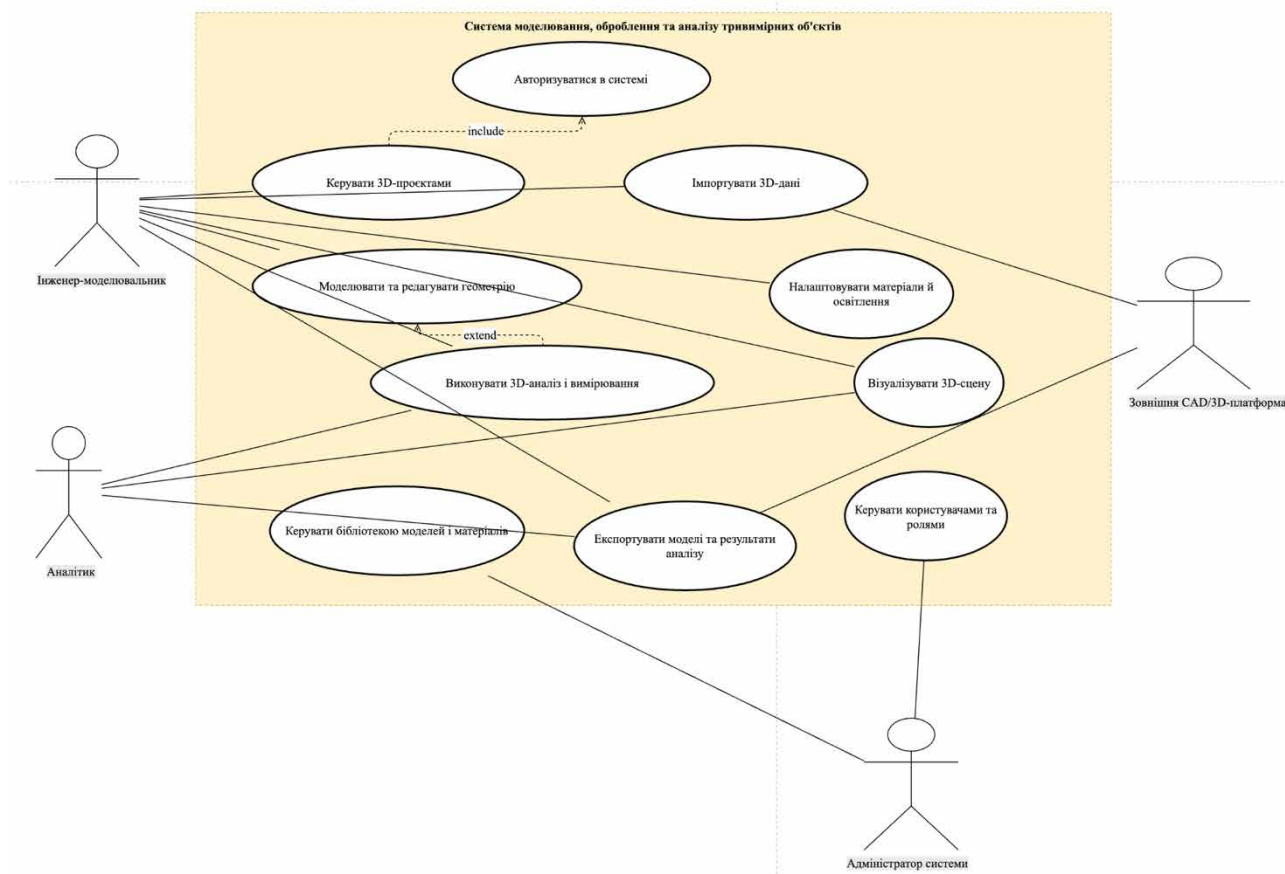


Рис. 1.12. Діаграма прецедентів системи моделювання, оброблення та аналізу тривимірних об'єктів

На рис. 1.13 наведено діаграму послідовності, яка формалізує обробку 3D-даних у складній обчислювальній інфраструктурі. У схемі відображено взаємодію користувача з клієнтським застосунком, постановку завдань на планувальник (наприклад, SLURM), розподілену обробку на обчислювальному кластері з підтримкою MPI або GPU-вузлів, агрегацію результатів та повернення аналітичних метрик до інтерфейсу користувача. Така модель дозволяє описати часові залежності, точки синхронізації та паралельні сценарії виконання, що є критично важливими під час роботи з великими 3D-сценами та ресурсомісткими операціями реконструкції чи кластеризації.

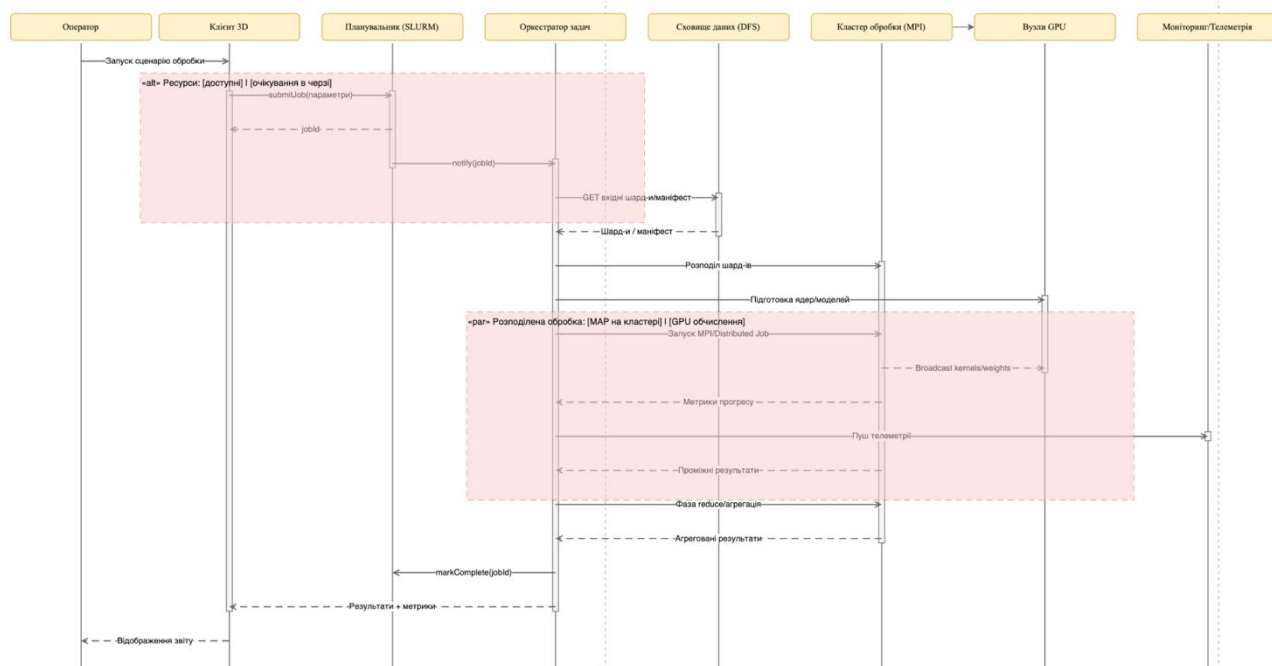


Рис. 1.13. Діаграма послідовності для сценарію розподіленого аналізу та оброблення 3D-об'єктів

На рис. 1.14 подано діаграму діяльності, яка описує бізнес-процес користувача в системі: від авторизації й вибору або створення нового 3D-проекту до імпорту моделі, редагування геометрії, налаштування параметрів матеріалів і освітлення, запуску аналізу, генерації візуалізації та збереження результатів. Діаграма відображає можливість розгалуження сценарію (наприклад, залежно від існування проекту), а також послідовність дій, що формують повний цикл роботи з 3D-даними.

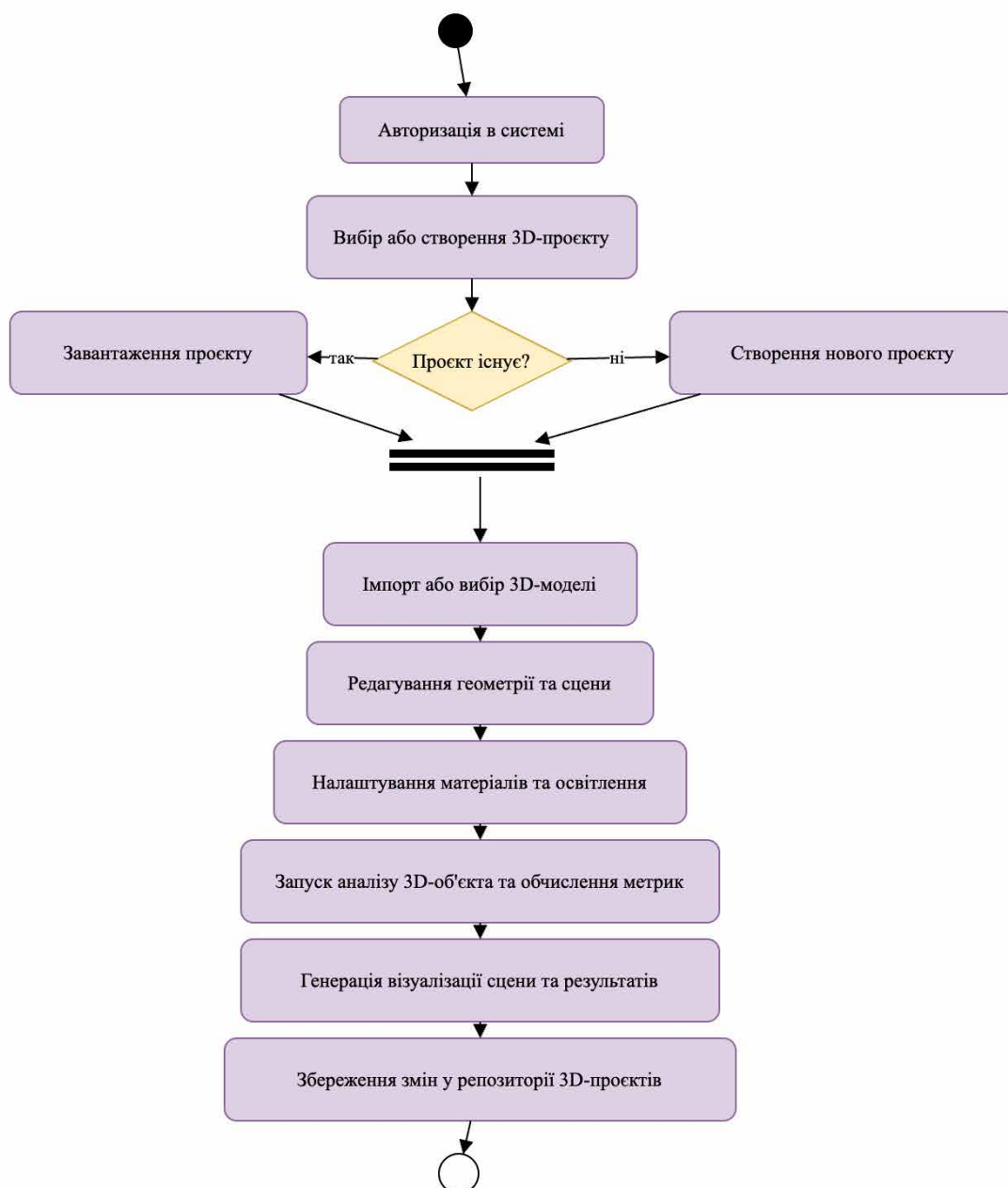


Рис. 1.14. Діаграма діяльності бізнес-процесу створення, редагування та аналізу 3D-проекту

Узагальнюючи результати моделювання предметної області, можна зазначити, що система охоплює повний спектр операцій - від імпорту й редагування геометрії до розподіленого аналізу, візуалізації та керування бібліотеками моделей. Формалізовані діаграми забезпечують чітке розуміння цільових функцій, ролей користувачів та взаємозв'язків між компонентами, що дозволяє обґрунтувати архітектурні рішення та забезпечити узгодженість подальших етапів проектування. Модельована структура предметної області

демонструє високий рівень комплексності системи та її орієнтацію на інтеграцію інструментів моделювання, оброблення та інтелектуальної аналітики в єдиному робочому середовищі.

## 1.5 Аналіз вимог програмної системи

Аналіз вимог експертної системи моделювання, оброблення та аналізу тривимірних об'єктів передбачає формування структурованого переліку функціональних, нефункціональних та вимог до безпеки, що визначають архітектуру, поведінку та інтерфейс системи. Оскільки програмний комплекс поєднує засоби геометричного редагування, аналітики, інтелектуальної обробки та взаємодії з зовнішніми 3D-платформами, вимоги мають бути узгоджені з моделями предметної області, представленими у попередніх підрозділах. На таблиці 1.3 наведено основні функціональні вимоги, які охоплюють можливість створення й керування 3D-проектами, імпорт і попередню обробку моделей, редагування геометрії, налаштування матеріалів і освітлення, запуск аналітичних обчислень та експорт отриманих результатів.

Таблиця 1.3

### Функціональні вимоги експертної системи

№	Вимога	Опис
1	Керування 3D-проектами	Створення, збереження, завантаження та структуризація проектів
2	Імпорт 3D-даних	Завантаження моделей у форматах STL, OBJ, PLY, GLTF, PointCloud
3	Попередня обробка	Нормалізація, фільтрація шумів, реконструкція поверхні, вирівнювання
4	Редагування геометрії	Зміна топології, трансформації, скульптування, ремешинг
5	Налаштування матеріалів	Призначення текстур, параметрів освітлення, шейдерів
6	Візуалізація 3D-сцени	Рендеринг сцени, інтерактивний перегляд, анімація
7	Аналітика 3D-даних	Обчислення метричних характеристик, кластеризація, дескриптори
8	Експорт	Експорт моделей, аналітичних результатів та звітів

Нефункціональні вимоги визначають продуктивність, якість, надійність і масштабованість системи. Їх структуровано у таблиці 1.4, де виділено ключові метрики: підтримку великих обсягів даних, інтерактивну взаємодію, адекватний час відгуку аналітичних модулів та кросплатформність. Система повинна забезпечувати стабільну роботу з хмарами точок до десятків мільйонів елементів, підтримувати GPU-прискорення й оптимізовані алгоритми реконструкції та візуалізації.

Таблиця 1.4

## Нефункціональні вимоги експертної системи

№	Вимога	Опис
1	Продуктивність	Обробка великих 3D-моделей з мінімальними затримками
2	Масштабованість	Підтримка локального режиму, кластерної та розподіленої обробки
3	Надійність	Стійкість до помилок, журналювання операцій, механізми відновлення
4	Зручність використання	Інтуїтивний інтерфейс, підтримка інтерактивного редагування
5	Сумісність	Інтеграція з CAD/CAE системами та файлами промислових форматів
6	Візуальна якість	Фотореалістичне відтворення сцени, коректна робота шейдерів

Вимоги до безпеки охоплюють механізми автентифікації, авторизації, контроль ролей, захист переданих даних, а також аудит операцій, пов'язаних із модифікацією 3D-проектів. У таблиці 1.5 наведено основні аспекти безпеки, що є критичними у разі спільної роботи кількох користувачів та інтеграції з корпоративною інфраструктурою.

Таблиця 1.5

## Вимоги до безпеки експертної системи

№	Вимога	Опис
1	Автентифікація	Авторизація користувачів через локальні або зовнішні IdP
2	Авторизація	Рольове керування доступом до функцій та проектів (RBAC)

## Продовження таблиці 1.5

3	Захист даних	Шифрування файлів, захист каналів (TLS 1.3)
4	Аудит	Ведення журналу змін моделей та результатів аналізу
5	Інтеграційна безпека	Контроль доступу зовнішніх CAD/3D-платформ до даних

Узагальнюючи аналіз вимог, можна стверджувати, що запропонована експертна система характеризується високим рівнем функціональної насиченості та орієнтована на складні процеси моделювання й інтелектуального аналізу 3D-даних. Формалізовані вимоги забезпечують чітке визначення архітектурних рішень, критичних характеристик продуктивності, вимог до якості та безпеки, що створює основу для проєктування модульної, масштабованої та надійної системи, здатної працювати з великими обсягами тривимірної інформації та забезпечувати комплексну підтримку процесів моделювання, оброблення та аналітики.

## 1.6 Постановка завдання

Постановка завдання для розроблення експертної системи моделювання, оброблення та аналізу тривимірних об'єктів полягає у формуванні цілей, визначенні меж програмного комплексу, встановленні типів вхідних і вихідних даних, а також визначенні очікуваних функціональних результатів, що забезпечують повний цикл роботи з 3D-сценами. Вхідними даними системи є тривимірні моделі різних форматів (STL, OBJ, PLY, GLTF, XYZ, LAS), хмари точок, топологічні сітки, матеріальні характеристики, параметри освітлення, а також супровідні метадані, що визначають структуру та логіку побудови 3D-об'єктів. Додатковими вхідними даними є параметри проєкту, вимоги користувача щодо рівня деталізації та точності аналізу, а також конфігураційні налаштування робочого середовища, пов'язані з продуктивністю, GPU-акселерацією та форматом візуалізації.

Вихідними даними експертної системи є оновлені 3D-моделі після геометричних трансформацій, реконструкції та редагування, аналітичні звіти з

метричними характеристиками об'єктів, результати кластеризації чи сегментації структур, візуалізації сцен у вигляді зображень, анімацій або інтерактивних проєкцій, а також експортовані файли моделей і матеріалів для подальшого використання у CAD/CAE-середовищах. Вихідні дані також можуть включати попередньо оброблені хмари точок, реконструйовані топологічні сітки та оцінки точності геометричних перетворень.

Завданням системи є побудова інструменту, здатного автоматизувати процеси імпорту, оброблення, моделювання та аналізу тривимірних об'єктів, забезпечити інтерактивність і точність геометричних операцій, підтримати інтелектуальні алгоритми обчислення структурних характеристик і забезпечити коректну інтеграцію з зовнішніми CAD/3D-платформами. Система повинна забезпечувати можливість редагування геометрії, налаштування матеріалів і освітлення, виконання ресурсомістких аналітичних обчислень, а також формування візуалізованих результатів, придатних для інженерного аналізу, контролю якості та підготовки проєктної документації. Узгодженість вхідних і вихідних даних, формалізація процесів та модульна структура створюють основу для комплексного рішення, що забезпечує надійне, масштабоване та високоточне опрацювання тривимірної інформації в інженерних і наукових завданнях.

## **1.7 Висновки до першого розділу**

У першому розділі здійснено комплексний науково-технічний аналіз предметної області моделювання, оброблення та аналізу тривимірних об'єктів, що дозволило сформувавши ґрунтовну теоретичну основу для подальшої розробки експертної системи. На основі огляду сучасних підходів і методів 3D-реконструкції, геометричного моделювання, аналізу хмар точок та візуалізації було встановлено, що ключовими напрямками розвитку галузі є оптимізація обчислювальних процесів, підвищення точності моделей, інтеграція алгоритмів машинного навчання та створення універсальних інструментів для інженерних і наукових задач. Проведений огляд стану наукових досліджень показав, що

сучасні роботи спрямовані на автоматизацію реконструкції, багатоканальне опрацювання геометричних структур, підвищення стійкості до шумів, а також використання гібридних підходів, що поєднують класичні методи та глибинні нейромережеві моделі.

Аналіз існуючих рішень, таких як Blender, MeshLab, Autodesk Meshmixer, CloudCompare та Maya, продемонстрував значний розвиток функціональних можливостей платних і відкритих платформ, але водночас виявив низку обмежень, пов'язаних із недостатньою інтегрованістю аналітичних модулів, низьким рівнем автоматизації вимірювань, відсутністю інтелектуальної підтримки під час прийняття рішень і обмеженою підтримкою складних сценаріїв оброблення великих хмар точок. Це підтвердило актуальність створення спеціалізованої експертної системи, яка об'єднує інструменти геометричного моделювання, аналітики та інтелектуального аналізу даних у єдиному інтерактивному середовищі.

На основі побудованих UML-діаграм - діаграми прецедентів, діаграми послідовності та діаграми активності - сформовано концептуальну модель функціонування системи, яка демонструє взаємодію ключових користувачів, внутрішніх модулів та зовнішніх платформ. Це дозволило встановити логічні зв'язки між операціями імпорту, редагування геометрії, налаштування сцени, виконання 3D-аналізу та експорту результатів.

Проаналізовані вимоги до системи - функціональні, нефункціональні та вимоги до безпеки - дали змогу формалізувати технічні характеристики майбутнього програмного забезпечення, визначити обмеження, області відповідальності та критерії успішності. Сформульована постановка завдання визначила структуру вхідних і вихідних даних, що забезпечує узгодженість процесів і створює передумови для побудови масштабованого та точного інструменту для роботи з тривимірною інформацією.

Перший розділ сформував цілісну науково-практичну базу для подальшого проектування архітектури системи, її алгоритмічних модулів та реалізації

функціональних компонентів, що забезпечать інноваційність, високу точність та ефективність роботи з 3D-об'єктами у різних прикладних сферах.

## 2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 2.1 Логічна модель даних у вигляді ER-діаграми

Логічна модель даних експертної системи моделювання, оброблення та аналізу тривимірних об'єктів відображає узагальнену структуру інформаційних сутностей, необхідних для підтримки повного життєвого циклу 3D-проектів, та спроектована відповідно до принципів нормалізації, що мінімізують надлишковість і підвищують цілісність даних; на ER-діаграмі, зображеній на рис. 2.1, виділено ключові сутності UserAccount, Project3D, Scene3D, Model3D, Material та AnalysisResult, між якими встановлено відношення «користувач – проект – сцена – модель – матеріал – результат аналізу», що забезпечує можливість відстеження походження кожної геометричної моделі, її варіанта оброблення та пов'язаних аналітичних метрик у розрізі конкретних користувачів і сценаріїв дослідження.

Рис. 2.1 – Логічна модель даних експертної системи у вигляді ER-діаграми

Для формалізації структури логічної моделі даних у таблиці 2.1 подано узагальнені характеристики основних сутностей, зокрема призначення, ключі та короткі коментарі щодо використання у процесах моделювання, оброблення й аналізу 3D-об'єктів, що дозволяє узгодити рівень абстракції моделі з вимогами до функціональності системи та сценаріями її експлуатації.

Таблиця 2.1

Основні сутності логічної моделі даних експертної системи

Сутність	Первинний ключ	Коротка характеристика
UserAccount	user_id	Облікові записи користувачів, ролі та атрибути доступу
Project3D	project_id	Логічні контейнери 3D-проектів і пов'язаних ресурсів
Scene3D	scene_id	Конкретні сцени з налаштуваннями камер і оточення

Закінчення таблиці 2.1

Model3D	model_id	Геометричні моделі, прив'язані до сцени та проєкту
Material	material_id	Параметри матеріалів і шейдерів, що застосовуються до моделей
AnalysisResult	analysis_id	Збережені результати розрахунків і аналітичні метрики

Результуюча структура логічної моделі даних відповідає третій нормальній формі: атрибути сутностей однозначно залежать від первинних ключів, відсутні транзитивні залежності, а зв'язки «один-до-багатьох» між користувачами, проєктами, сценами, моделями та результатами аналізу чітко розмежовують відповідальність компонентів системи, що є критично важливим для подальшої фізичної реалізації схеми БД, забезпечення узгодженості даних при паралельній роботі декількох користувачів і можливості масштабування експертної системи в умовах зростання кількості 3D-проєктів і складності аналітичних сценаріїв.

## 2.2 Діаграма класів та кооперації

Архітектурна модель програмного забезпечення для моделювання, оброблення та аналізу тривимірних об'єктів ґрунтується на чіткій структуризації доменної логіки та розмежуванні відповідальності між компонентами, що дозволяє забезпечити масштабованість, розширюваність і стійкість системи під час виконання обчислювально складних операцій. На діаграмі класів, поданій на рис. 2.2, відображено фундаментальні об'єкти доменної області – `UserSession`, `Project3D`, `Model3D`, `Repository3D`, `AnalysisService`, `RenderingService` – які утворюють цілісну модель, орієнтовану на інкапсуляцію функціональних ролей та ізоляцію станів під час роботи з даними проєктів. В основу побудови класової структури покладено принципи низької зв'язності та високої когезії, що є ключовими для складних інтерактивних систем, які працюють із 3D-геометрією, великою кількістю файлів та обчислювальними сценами.

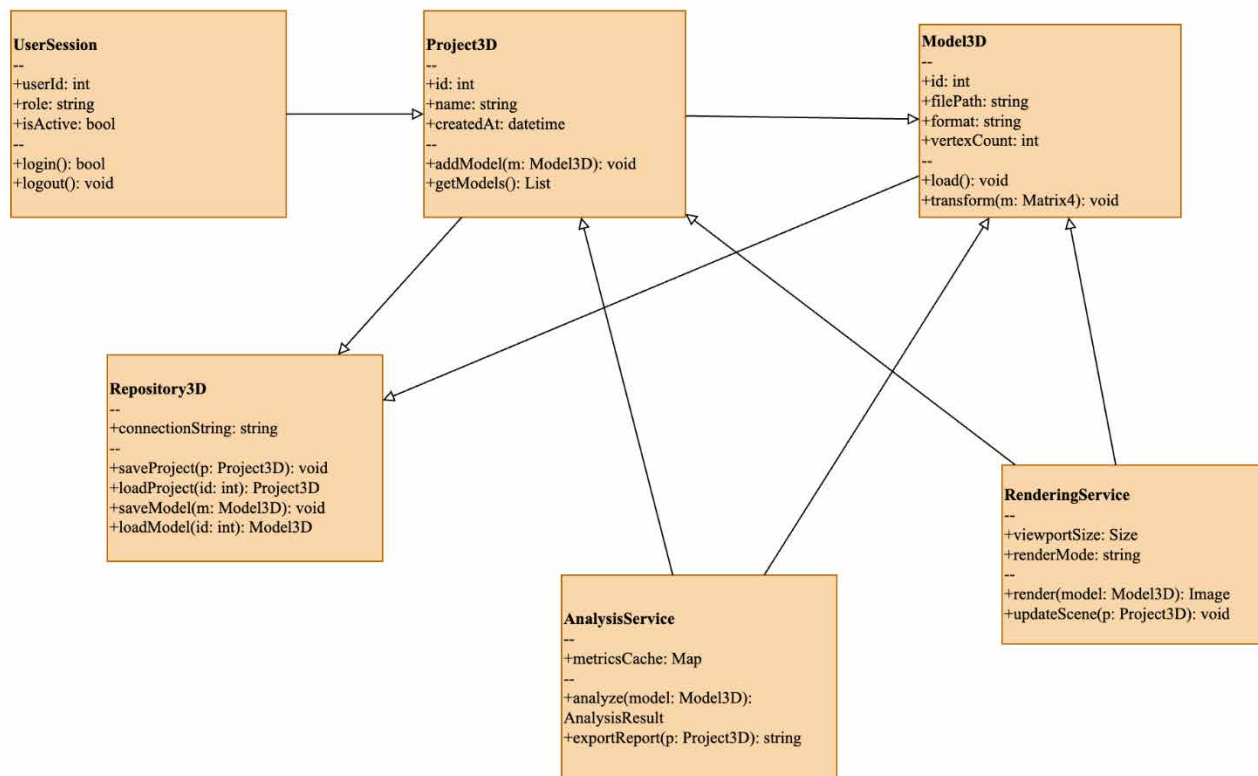


Рис. 2.2. Діаграма класів системи моделювання, оброблення та аналізу тривимірних об'єктів

Кооперація між компонентами системи охоплює сценарії автентифікації, завантаження проєктів, імпорту геометричних моделей, підготовки даних до оброблення й виконання 3D-аналізу. Взаємодію між класами у процесі відкриття 3D-проєкту показано на рис. 2.3: `UserSession` ініціює автентифікацію, UI-клієнт передає запит до `ProjectService`, який звертається до `Repository3D` та повертає відповідний контекст проєкту. Такий розподіл ролей забезпечує відокремлення бізнес-логіки від механізмів доступу до даних, що підвищує гнучкість та придатність до модифікацій.

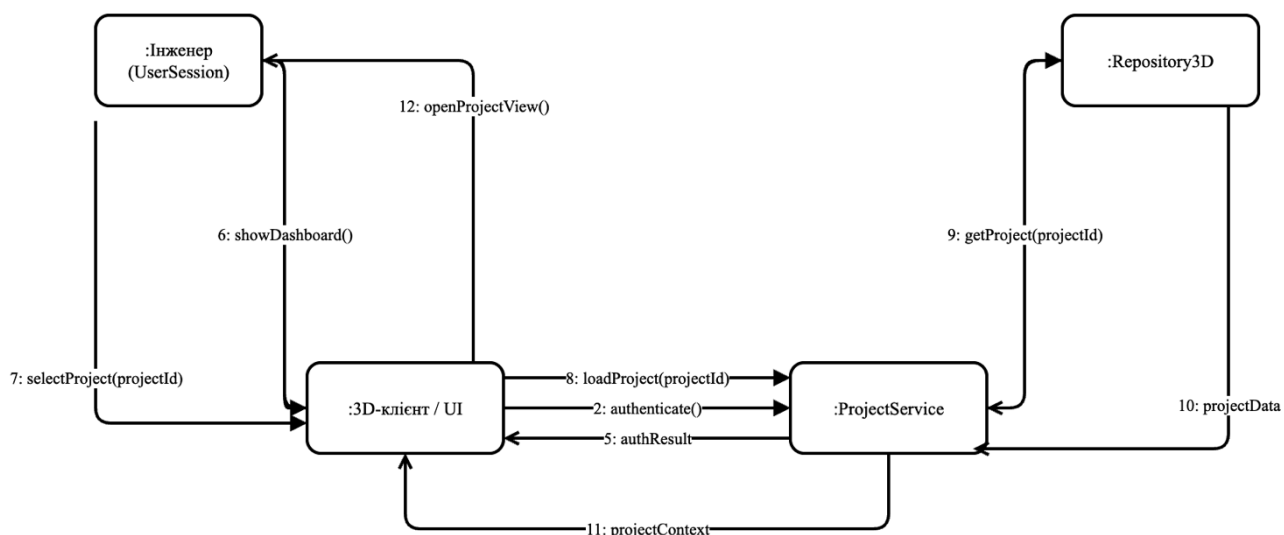


Рис. 2.3. Діаграма кооперації для сценарію відкриття 3D-проекту

Другий ключовий сценарій - імпорт моделі - наведено на рис. 2.4. Кооперація між 3D-клієнтом, ImportService, Repository3D та внутрішнім перетворювачем Model3D забезпечує поетапне завантаження сирих файлів (OBJ, STL, FBX), їх конвертацію у внутрішній формат системи та збереження метаданих. Логіка імпорту побудована з урахуванням можливості масштабування на великі набори моделей і забезпечує нормалізацію геометричних даних, що є важливим для подальшої аналітичної обробки.

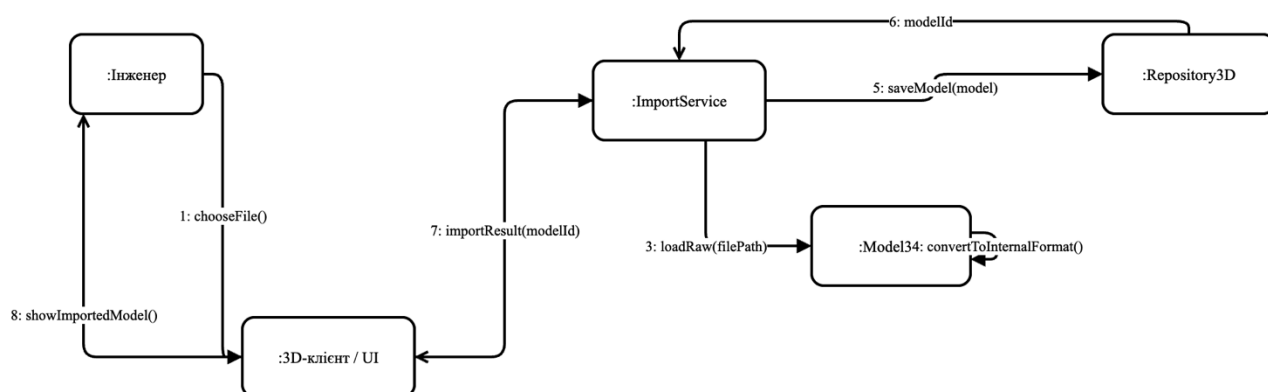


Рис. 2.4. Діаграма кооперації для сценарію імпорту 3D-моделі

На рис. 2.5 показано кооперацію для процесу виконання аналізу 3D-об'єкта. У цьому сценарії 3D-клієнт ініціює обчислення, після чого AnalysisService виконує розрахунки метрик (топологічних, геометричних, спектральних), а RenderingService формує візуалізацію результатів. Взаємодія сервісів побудована таким чином, щоб забезпечити асинхронність оброблення,

повторне використання кешованих результатів та оптимізацію навантаження на GPU/CPU-ресурси.

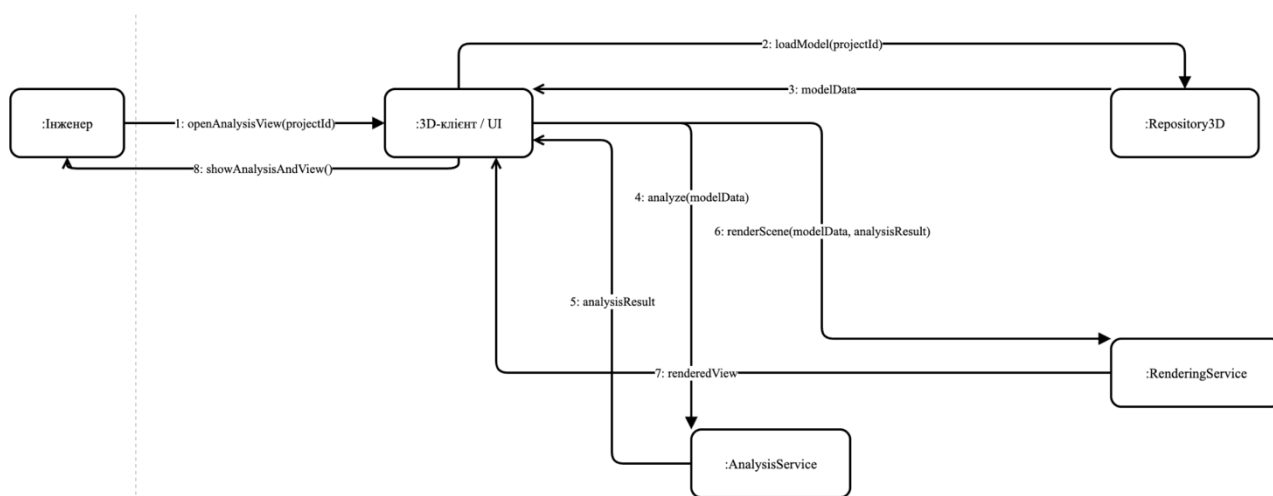


Рис. 2.5. Діаграма кооперації для сценарію аналізу 3D-моделі

Узагальнені характеристики ключових класів та їхніх ролей наведено в таблиці 2.2, що дозволяє сформувати системне уявлення про архітектурну модель і підтвердити відповідність її структури вимогам логічної декомпозиції та підтримки масштабування.

Таблиця 2.2

#### Основні класи системи та їх функціональне призначення

Клас	Призначення
UserSession	Керування сесією користувача, автентифікація, ролі
Project3D	Модель проекту, агрегація моделей та сцени
Model3D	Представлення 3D-геометрії та операцій трансформації
Repository3D	Доступ до сховища проектів і моделей, операції збереження/завантаження
AnalysisService	Розрахунок аналітичних метрик, оцінок, дескрипторів
RenderingService	Генерація візуалізацій, відтворення сцени та формування рендерів

Результуюча архітектурна модель на рівні класів та кооперацій демонструє системну інтеграцію функціональних компонентів, підтримує повну трасованість операцій над 3D-моделями та забезпечує сталу основу для подальшої реалізації модульної структури програмного забезпечення. Такий підхід дозволяє зберегти узгодженість між моделями даних, сценаріями

взаємодії та технічними вимогами, що є критично важливим для коректної роботи системи в умовах високої складності вхідних даних та інтенсивного аналітичного навантаження.

## 2.3 Діаграма компонентів

Архітектурна декомпозиція системи моделювання, оброблення та аналізу тривимірних об'єктів передбачає чітке структурування функціональних модулів, що забезпечують взаємодію користувача з даними, виконання геометричних перетворень, обчислення 3D-метрик та формування візуалізацій. На рис. 2.6 наведено діаграму компонентів, яка відображає логічну організацію системи та інтерфейси між її ключовими підсистемами. Кожний компонент реалізує окремий набір функцій, інкапсулює внутрішню логіку та взаємодіє з іншими модулями через стандартизовані API-контракти, що підвищує гнучкість і масштабованість системи під час інтеграції нових функціональних можливостей.

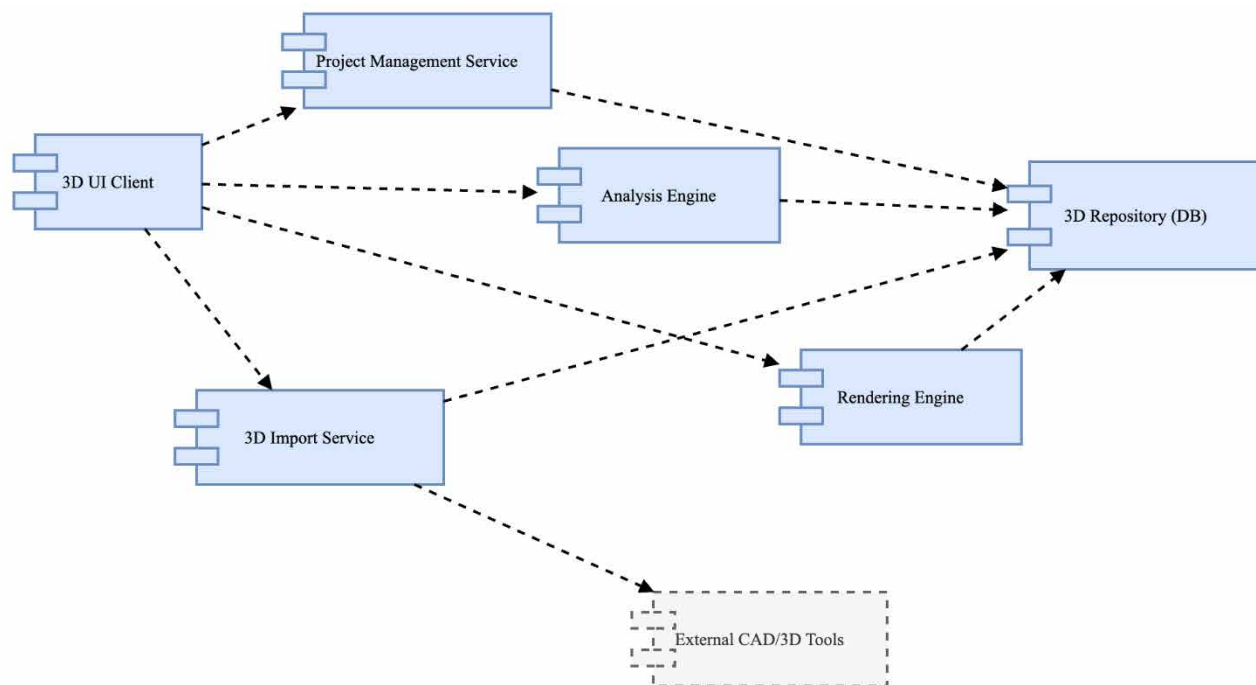


Рис. 2.6. Діаграма компонентів системи моделювання, оброблення та аналізу тривимірних об'єктів

У структурі компонентів важливе місце займають сервіси Project Management Service, 3D Import Service, Analysis Engine та Rendering Engine, які

формують ядро системи й забезпечують оркестрацію операцій над проектами та 3D-моделями. Компонент 3D Repository виступає як централізоване сховище геометричних даних, матеріалів, сцен та результатів аналізу, підтримуючи транзакційність та узгодженість даних при одночасній роботі множини користувачів. Взаємодія з зовнішніми CAD/3D-платформами відбувається через окремий інтеграційний модуль, що дозволяє імпортувати моделі з різних форматів і забезпечує розширення функціональності системи без модифікації базових компонентів.

Узагальнена характеристика компонентів наведена в таблиці 2.3, що дозволяє концептуально оцінити їх роль у загальній архітектурі та визначити ключові точки взаємодії між ними.

Таблиця 2.3

#### Основні компоненти системи та їх функціональне призначення

Компонент	Призначення
3D UI Client	Інтерактивна взаємодія з користувачем, керування сценами та моделями
Project Management Service	Оброблення операцій над 3D-проектами, керування метаданими
3D Import Service	Імпорт геометрії, конвертація форматів, первинна нормалізація моделей
Analysis Engine	Виконання аналітичних обчислень, генерація метрик і структур дескрипторів
Rendering Engine	Формування візуалізацій, рендеринг сцен, підготовка графічних результатів
3D Repository (DB)	Збереження моделей, сцен, матеріалів, результатів аналізу та журналів
External CAD/3D Tools	Інтеграція сторонніх систем моделювання та експорту/імпорту даних

Результуюча архітектурна модель компонентів формує цілісне середовище для підтримки повного циклу роботи з тривимірними об'єктами — від формування проєкту до виконання комплексного аналізу та генерації візуальних результатів. Така структуризація забезпечує не лише логічну завершеність

системи, а й високу гнучкість її подальшого розширення за рахунок модульності та стандартизованих інтерфейсів.

## 2.4 Діаграма пакетів

Для забезпечення масштабованості, розширюваності та чіткої модульності програмного забезпечення системи моделювання, оброблення та аналізу тривимірних об'єктів використано пакетно-орієнтований підхід до організації коду. На рис. 2.7 наведено діаграму пакетів, яка відображає логічний поділ системи на функціональні домени, кожен з яких інкапсулює власний набір класів, сервісів та обчислювальних процедур. Така структуризація відповідає вимогам ДСТУ щодо формування багаторівневих архітектур та забезпечує мінімальний рівень зв'язності між підсистемами при максимальній концентрації відповідальності всередині кожного пакета.

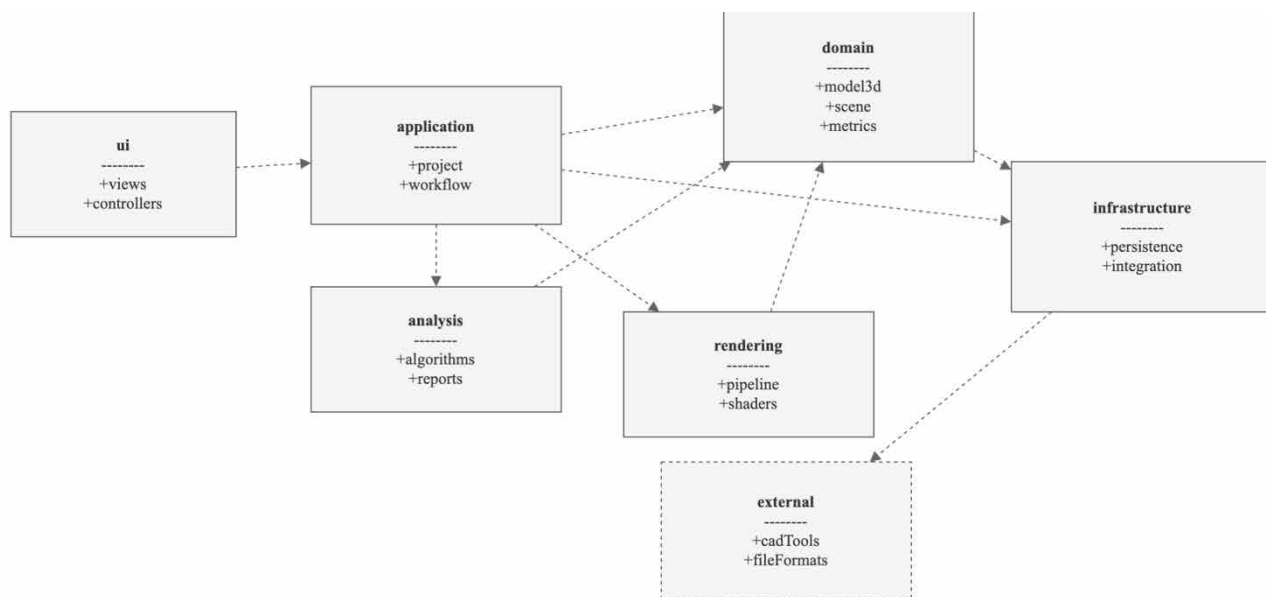


Рис. 2.7. Діаграма пакетів системи моделювання, оброблення та аналізу тривимірних об'єктів

Архітектурний поділ на пакети дозволяє розмежувати логіку представлення (ui), бізнес-правила та керування робочими процесами (application), предметну область (domain), рендеринг та графічний конвеєр (rendering), аналітичні алгоритми (analysis), а також інфраструктурні сервіси,

включно з persistence-рівнем і каналами інтеграції (infrastructure). Окремий пакет external забезпечує взаємодію з CAD/3D-інструментами й підтримку різних форматів файлів, що дає змогу розширювати межі системи без зміни внутрішнього ядра. Така побудова відповідає принципам Domain-Driven Design і Clean Architecture, забезпечує незалежність модулів та суттєво спрощує супровід програмного забезпечення на всіх етапах його життєвого циклу.

У таблиці 2.4 наведено узагальнену характеристику ключових пакетів системи, що дозволяє оцінити їх функціональне навантаження та визначити їх місце у загальній архітектурі.

Таблиця 2.4

#### Основні пакети системи та їх функціональне призначення

Пакет	Призначення
ui	Відповідає за представлення інформації, інтерфейс, контролери, інтерактивну взаємодію
application	Координує робочі процеси, керує проєктами, реалізує бізнес-логіку
domain	Містить сутності предметної області: 3D-моделі, сцени, метрики
analysis	Реалізує аналітичні алгоритми, обчислення, формування структурних звітів
rendering	Містить графічний конвеєр (pipeline), системи шейдерів, механізми рендерингу
infrastructure	Забезпечує доступ до БД, інтеграцію, системні сервіси, драйвери й обгортки над зовнішніми API
external	Взаємодія зі сторонніми 3D/CAD-платформами та конвертація форматів файлів

Результуюча пакетно-орієнтована структура забезпечує стабільне функціонування системи навіть за умови інтенсивного розширення обчислювальних можливостей або зміни алгоритмів аналізу. Відокремлення інтерфейсної, логічної, аналітичної та інфраструктурної частин знижує ризики перехресного впливу модулів і підвищує рівень керованості приєднання нових функцій, що є необхідним для складного ПЗ, орієнтованого на роботу з тривимірними даними.

## 2.5 Висновки до другого розділу

У другому розділі було сформовано цілісну архітектурно-структурну модель програмного забезпечення для моделювання, оброблення та аналізу тривимірних об'єктів, що відповідає вимогам сучасних інтелектуальних систем і принципам побудови високонадійних програмних рішень. Побудовано логічну модель даних, яка забезпечує повну нормалізацію сутностей предметної області та уніфіковану структуру зберігання 3D-моделей, сцен, аналітичних характеристик та метаданих про проєкти. На основі ER-підходу визначено стійкі зв'язки між сутностями, що дозволяє гарантувати цілісність даних і консистентність операцій незалежно від складності обчислювальних задач.

Діаграма класів та діаграми кооперації продемонстрували функціональну взаємодію між компонентами системи, включно з модулями імпорту, візуалізації, аналізу та управління робочими процесами. Такий підхід забезпечує відокремлення бізнес-логіки від інфраструктурних механізмів і дозволяє ефективно масштабувати систему, впроваджуючи нові алгоритми та методи опрацювання 3D-даних без порушення базових механізмів. Коопераційні схеми детально показали послідовність обміну повідомленнями між сервісами та ролями користувачів, що є ключовим для підтвердження коректності виконання робочих сценаріїв у режимі реального часу.

Створена діаграма компонентів відобразила незалежність функціональних модулів і визначила чіткі межі між підсистемами UI, логічного ядра, аналітичних алгоритмів, рендерингового конвеєра та сховища даних, що відповідає архітектурним принципам низької зв'язності та високої когезії. У свою чергу, діаграма пакетів продемонструвала завершену модульну структуру, що підтримує розширюваність, переносимість і супровідність ПЗ, забезпечуючи можливість інтеграції з зовнішніми CAD/3D-інструментами та новими форматами файлів.

У сукупності результати другого розділу формують комплексне архітектурне підґрунтя для подальшої реалізації програмної системи,

забезпечують концептуальну цілісність її побудови та підтверджують відповідність вимогам до сучасних інтелектуальних систем 3D-оброблення. Структурні, логічні та взаємодіючі моделі, наведені в цьому розділі, виступають базисом для розроблення алгоритмів, механізмів рендерингу та обчислювальних компонентів, що будуть розкриті у наступних частинах роботи.

## 3 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 3.1 Вибір технологій та інструментальних засобів реалізації системи

Реалізація програмного забезпечення для моделювання, оброблення та аналізу тривимірних об'єктів потребує комплексного поєднання високорівневих мов програмування, графічних API, аналітичних бібліотек та інструментів управління даними. Архітектура системи, сформована у попередньому розділі, вимагає використання засобів, здатних забезпечити високу точність геометричних операцій, ефективний рендеринг, масштабовану обробку даних та інтеграцію з сучасними форматами 3D-моделей. Вибір технологій здійснюється з урахуванням таких критеріїв, як продуктивність, портативність, підтримка апаратного прискорення, надійність і можливість подальшого розширення системи.

Для реалізації ядра 3D-обробки обрано C++ з бібліотеками OpenGL/GLM, що забезпечують низькорівневий контроль над геометричними операціями, трансформаціями та буферизацією вершин. Рендерингова частина використовує OpenGL 4.x з підтримкою шейдерів GLSL, що дозволяє виконувати апаратно прискорені операції освітлення, нормалізації поверхонь та обчислення векторної графіки. Модуль аналітики, який виконує кластеризацію, метричні перетворення та дескрипторні обчислення, реалізовано на Python з використанням NumPy, SciPy, scikit-learn, що надає доступ до широкого спектра алгоритмів машинного навчання та швидкого чисельного обчислення.

Підсистема управління проєктами та зберіганням метаданих базується на PostgreSQL, що забезпечує транзакційну цілісність даних, підтримку JSON-полів (для параметрів моделей і сцен) та можливість масштабування. Зовнішні 3D-файли, зокрема формати \*.obj, \*.fbx, .ply, обробляються через Assimp, що

забезпечує уніфікований доступ до імпорту геометрії незалежно від CAD-середовища. Інтерактивний графічний інтерфейс клієнта реалізовано на Qt 6, що забезпечує кросплатформність, підтримку OpenGL-віджетів і оптимізовані інструменти UI.

Узагальнена характеристика обраних технологій наведена у таблиці 3.1.

Таблиця 3.1

## Технології та інструментальні засоби системи

Компонент системи	Обрана технологія	Основне призначення
Ядро 3D-обробки	C++, OpenGL, GLM	Геометричні операції, сцени, матричні перетворення
Імпорт моделей	Assimp	Завантаження різних форматів 3D-файлів
Аналітичний модуль	Python, NumPy, SciPy, scikit-learn	Аналіз геометрії, кластеризація, дескриптори
Рендеринг	OpenGL 4.x, GLSL	Високопродуктивна візуалізація 3D-об'єктів
Інтерфейс користувача	Qt 6	Кросплатформний UI та інтеграція з OpenGL
Сховище даних	PostgreSQL	Зберігання проєктів, сцен, метаданих і параметрів
Інтеграція з CAD	Assimp / сторонні API	Взаємодія з зовнішніми CAD-інструментами
Сервісна логіка	C++/Python	Реалізація бізнес-правил і комунікацій між модулями

Вибір зазначених технологій забезпечує оптимальний баланс між продуктивністю рендерингу, точністю чисельних обчислень і зручністю розробки. Такий підхід дозволяє інтегрувати у систему складні алгоритми аналізу моделей, забезпечити плавну роботу графічного інтерфейсу та гарантувати коректне зберігання й оброблення великих обсягів 3D-даних. Комбінація C++ і Python забезпечує гібридну архітектуру, у якій обчислювально важкі процеси виконуються на нативному рівні, а аналітичні – у високорівневому середовищі, що істотно підвищує ефективність розроблення і подальшої експлуатації системи.

## 3.2 Фізична модель даних і структура OLAP-кубу

Інформаційна база системи моделювання, оброблення та аналізу тривимірних об'єктів формує багаторівневий комплекс даних, що охоплює операційне 3D-сховище, нормалізоване предметно-орієнтоване сховище даних та багатовимірний аналітичний OLAP-куб. На рис. 3.1 подано узагальнену структуру потоків даних, що визначає взаємодію між ETL-процесами, DWH-рівнем та аналітичною підсистемою.

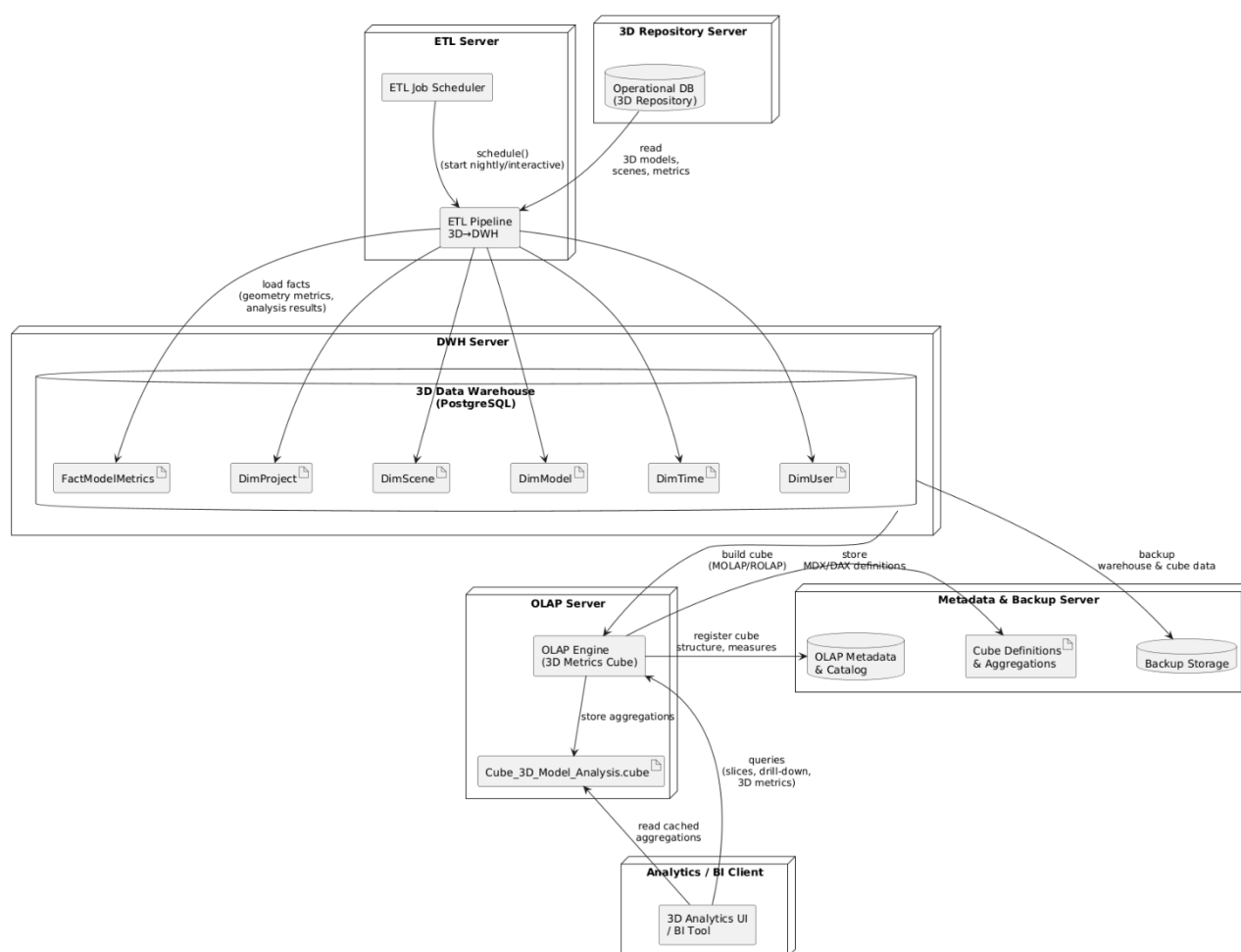


Рис. 3.1. Інтегрована структура інформаційної бази та потоків даних

Операційний рівень включає вихідні 3D-моделі, сцени, сіткові структури, текстури та первинні геометричні метрики, що генеруються інструментами редагування та імпорту. Ці дані є гетерогенними й містять дублювання, що обумовлює необхідність застосування нормалізації та попереднього

структурного узгодження. Нормалізація забезпечує приведення моделі даних до форми, що виключає надлишковість та гарантує узгодженість відношень типу «проект – сцена – модель». Результатом є доменна схема у вигляді логічних сутностей, відповідних 3NF, оптимізована під ETL-навантаження та подальшу аналітику.

На другому рівні формується сховище даних (DWH), у якому всі 3D-метрики, результати аналізу й часові характеристики перетворюються у структурований вигляд. Особливістю є те, що система оперує не лише фактичними числовими показниками, а й похідними ознаками, отриманими методом кластеризації. На рис. 3.2 наведено приклад кластерного розподілу нормованих геометричних ознак, який надалі інтегрується до вимірів куба.

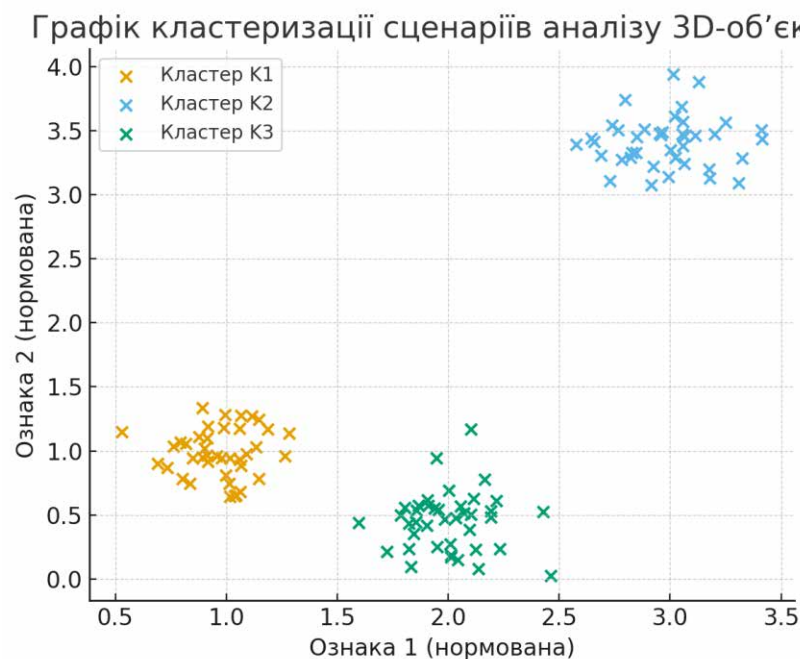


Рис. 3.2. Кластеризація нормованих геометричних ознак 3D-об'єктів

Формування кластерів потребує визначення оптимального числа груп. Для цього використано метод «ліктя», графік якого подано на рис. 3.3, що дозволяє мінімізувати сумарну квадратичну похибку та забезпечити стійкість подальшого аналітичного моделювання.

Графік методу «ліктя» для кластеризації сценаріїв аналізу 3D-об'єктів

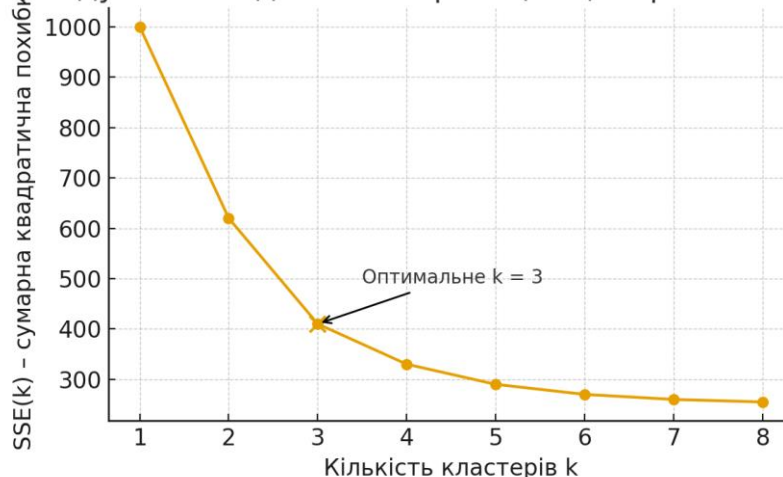


Рис. 3.3. Визначення оптимальної кількості кластерів методом «ліктя»

Нормовані інтегральні показники якості сітки, стійкості моделі та відповідності геометричних параметрів ілюструються на радіальній діаграмі (рис. 3.4), що є базою для побудови вимірів OLAP-кубу та дозволяє групувати моделі за інженерними характеристиками.

Радіальна діаграма нормованих показників сценарію аналізу тривимірного об'єкта

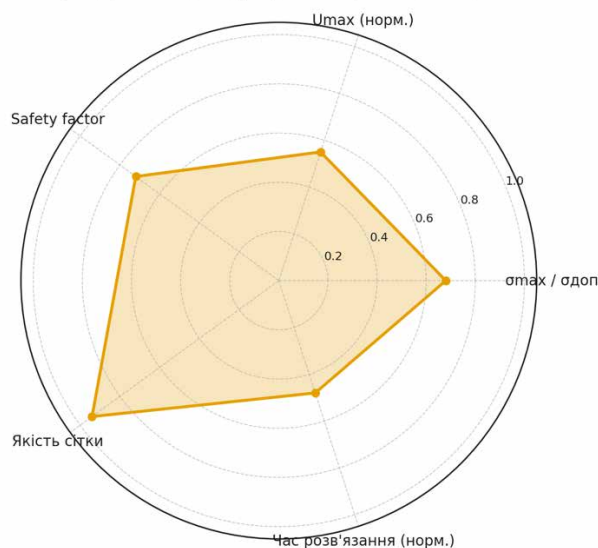


Рис. 3.4. Радіальна діаграма нормованих інженерно-геометричних показників

Фінальним компонентом інформаційної бази є багатовимірний аналітичний куб. Він містить виміри «Проект», «Сцена», «Модель», «Час аналізу», «Тип метрики» та «Кластер геометрії», а також таблицю фактів, що агрегує результати метричного та поведінкового аналізу. Куб підтримує операції

slice, drill-down, roll-up, pivot, забезпечуючи аналітику на рівні, недоступному для класичних реляційних запитів.

Додатково інформаційна база включає механізм інкрементального оновлення, який забезпечує автоматичне перенесення нових 3D-даних у DWH без повної перебудови. Це дозволяє застосовувати комбіновані аналітичні моделі, зокрема геометричні дескриптори, кластерні представлення та інтегровані показники якості сітки.

Щоб систематизувати ключові аспекти інноваційності інформаційної бази, у таблиці 3.2 наведено порівняння традиційних підходів і впроваджених у роботі вдосконалень.

Таблиця 3.2

#### Інноваційні аспекти інформаційної бази системи

Аспект	Традиційний підхід	Впроваджене рішення
Структура даних	Розрізнені моделі та сцени без формальної схеми	Повна нормалізація, 3NF, доменна модель «проект–сцена–модель–метрики»
Аналітика	Просте збереження метрик	Багатовимірний OLAP-куб з кластерними вимірами
Метрики	Первинні геометричні величини	Комплексні інтегральні інженерні показники
Взаємодія	Ручні запити	Інтерактивні slice/drill-down операції
Оновлення	Повне перезавантаження	Інкрементальні ETL-процеси

Узагальнюючи, інформаційна база системи формує цілісну платформу для структурованого, нормалізованого та аналітично орієнтованого опрацювання тривимірних даних. Вона забезпечує перехід від низькорівневої геометричної інформації до інженерних знань, що становить одну з ключових наукових новацій розробленої системи.

### 3.3 Архітектура системи 3D-візуалізації

Архітектурна модель розробленої інформаційної системи ґрунтується на принципах багаторівневої організації, модульності, слабкого зв'язування

сервісів та чіткої ізоляції обчислювальних, інтеграційних і візуалізаційних компонентів. Такий підхід забезпечує масштабованість, можливість подальшого розширення аналітичного ядра та інтеграцію з зовнішніми CAD-/3D-інструментами, що є критично важливим у контексті підвищення точності оброблення та аналізу геометричних моделей. На рисунку 3.5 наведено високорівневу архітектуру взаємодії клієнтського рівня, серверних мікросервісів та рівня даних.

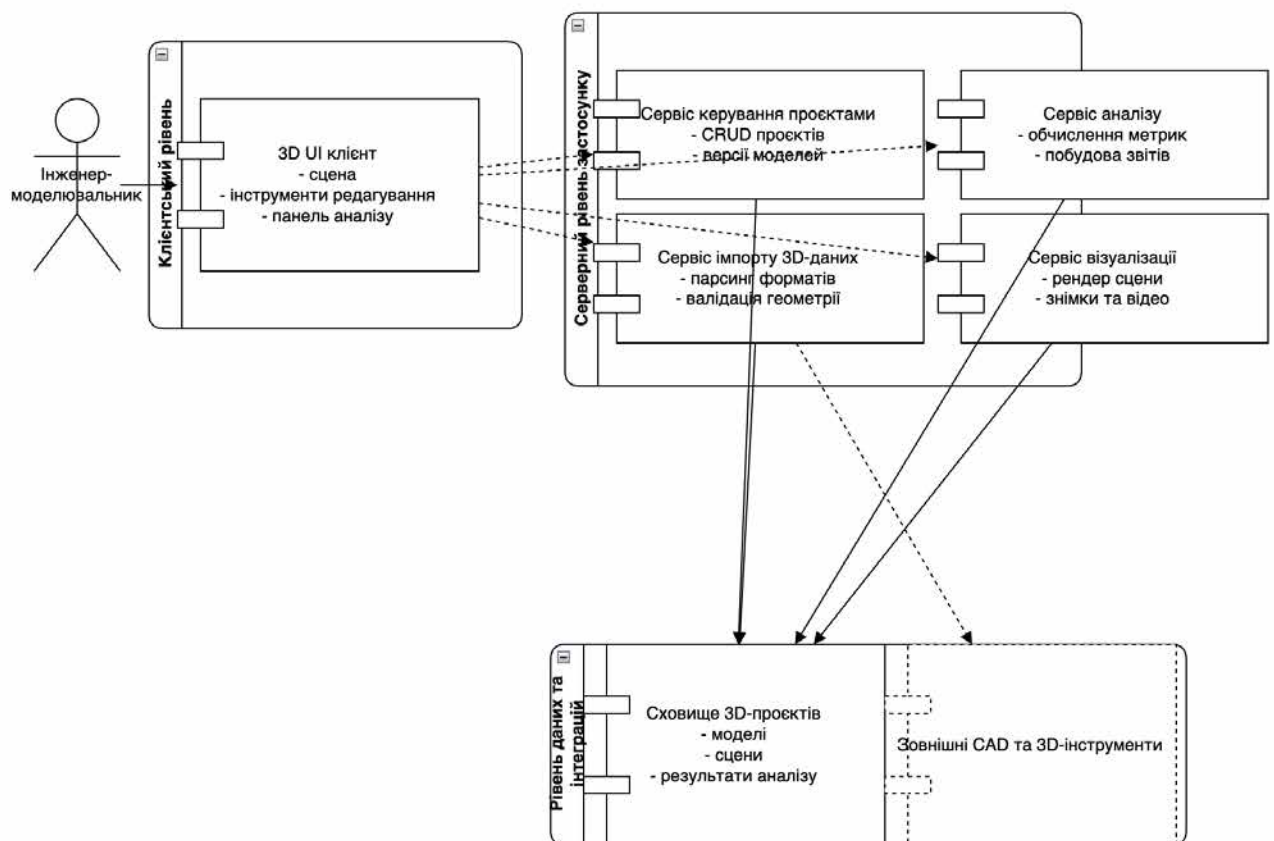


Рис. 3.5 Узагальнена архітектура інформаційної системи моделювання, імпорту та аналізу тривимірних об'єктів

На основі виділених сервісів було деталізовано внутрішню структуру програмної платформи, яка забезпечує виконання окремих функціональних процесів: керування проєктами, імпорту файлів різних форматів, розрахункового аналізу, високоточних процедур рендерингу та взаємодії з 3D-сховищем. При проєктуванні особлива увага приділялася забезпеченню ізолюваності обчислювальних модулів, оптимізації обміну даними між сервісами та мінімізації затримок при рендерингу сцен і виконанні алгоритмів аналізу.

На рисунку 3.6 подано структурну композицію серверних функціональних модулів та їх інтеграцію із зовнішніми інструментами.

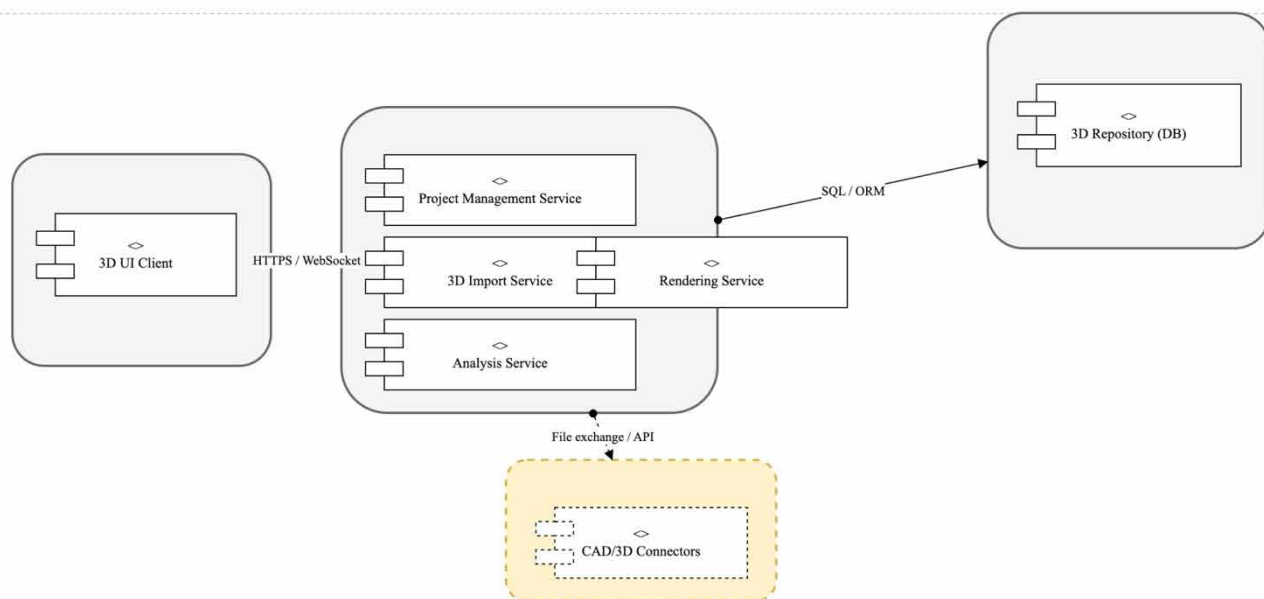


Рис. 3.6. Структура взаємодії програмних сервісів та інтеграційних компонентів системи

Результати дослідження визначили вимоги до побудови архітектури: підтримка паралельної обробки великих 3D-сцен, проведення геометричного та статистичного аналізу, генерація візуальних і аналітичних результатів, а також забезпечення надійного зберігання складних структур моделей (mesh-дані, сцени, топологічні зв'язки, метрики) з можливістю їх OLAP-аналізу. У рамках проєктування було сформовано єдину логічну шину даних, яка пов'язує аналітичний сервіс, модуль імпорту та модуль візуалізації зі сховищем моделей, гарантує узгодженість версій, повноту даних та детермінованість розрахункових процедур.

Системна архітектура забезпечує:

- об'єктно-орієнтовану організацію програмних сутностей;
- підтримку асинхронних подій під час обробки складних моделей;
- оптимізовані алгоритми передачі великих геометричних структур між сервісами;

— стабільну взаємодію з інструментами CAD через стандартизовані формати обміну.

Узгоджене функціональне проектування дозволило інтегрувати результати дослідження в архітектурну схему, сформувавши окремі доменні модулі, які забезпечують формування метрик тривимірних моделей, обчислення аналітичних показників, кластеризацію сценаріїв, побудову статистичних звітів та автоматизований контроль якості геометрії. Це створює цілісну платформу, що поєднує рендеринг, аналіз і зберігання даних у єдиному інформаційному контурі, забезпечуючи основу для подальшого впровадження складніших алгоритмів машинного навчання, багатокритеріальної оцінки та інтелектуальних рекомендацій.

### **3.5 Висновки до третього розділу**

У третьому розділі здійснено формування інженерної основи програмної системи моделювання, оброблення та аналізу тривимірних об'єктів, що включало вибір технологічного стеку, проектування інформаційної бази та розроблення архітектури з урахуванням результатів дослідження. На основі порівняльної оцінки інструментальних засобів обґрунтовано доцільність застосування відповідних мов програмування, бібліотек для роботи з 3D-геометрією, засобів аналітичної обробки та високопродуктивних рендерингових компонентів, що забезпечують необхідну точність, продуктивність та масштабованість програмної системи.

Було побудовано інформаційну модель, яка інтегрує дані тривимірних сцен, параметри геометрії, результати аналізу, статистичні метрики та характеристики рендерингу у єдину структуровану базу. Показано, що поєднання операційного 3D-сховища та аналітичного сховища (DWH + OLAP) забезпечує можливість виконання глибинних досліджень, кластеризації сценаріїв, порівняльних оцінок моделей та виявлення закономірностей у структурних і метрик-орієнтованих даних.

Запропонована архітектура системи реалізує модульний підхід із чітким розмежуванням функцій між сервісами: управління проєктами, імпорт 3D-даних, обчислювальний аналіз, формування результатів та рендеринг. Така структура гарантує узгодженість процесів, мінімізує залежності між компонентами та створює основу для подальшого розширення обчислювального ядра, впровадження нових алгоритмів аналізу та оптимізації оброблення великих 3D-сцен.

Таким чином, третій розділ забезпечив формування завершеної концепції проєктування системи, яка поєднує технологічну обґрунтованість, логічну цілісність інформаційної бази та гнучку багаторівневу архітектуру, що відповідає вимогам сучасних програмних платформ для високоточних досліджень у сфері тривимірного моделювання та аналітики.

## 4 ТЕСТУВАННЯ ТА ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ АНАЛІТИЧНОЇ СИСТЕМИ

### 4.1 План тестування програмних модулів та методика оцінювання результатів

Тестування програмної системи моделювання, оброблення та аналізу тривимірних об'єктів спрямоване на підтвердження коректності реалізації функціональних модулів, перевірку цілісності інформаційних потоків, стабільності обчислювальних процедур та відповідності системи вимогам, визначеним у розділах 1–3. Для формування відтворюваних та верифікованих результатів тестування передбачено структурований план, який охоплює модулі імпорту 3D-даних, управління проектами, обчислювального аналізу, рендерингу, а також підсистему зберігання та агрегування результатів.

Узгоджено з вимогами розроблено таблицю плану тестування програмних модулів, що визначає вхідні умови, методи перевірки, очікуваний результат та критерії прийнятності. Текст посилання на таблицю 4.1 подано нижче. Таблиця 4.1 містить узагальнений набір ключових тестових сценаріїв, які забезпечують повноту охоплення функціональних та технічних характеристик системи.

Таблиця 4.1

План тестування програмних модулів системи

№	Модуль	Тестовий сценарій	Вхідні дані	Очікуваний результат	Критерій успішності
1	Імпорт 3D-даних	Завантаження моделі різних форматів (OBJ, STL, FBX)	Файли тестових моделей	Коректний парсинг, нормалізація геометрії, валідація	100% сумісних форматів оброблено без помилок
2	Управління проектами	Створення, оновлення та версіювання проекту	Метадані проекту, змінені моделі	Нові версії зберігаються; відповідність	Жодної розбіжності між версіями

				структурі репозиторію	та репозиторієм
--	--	--	--	--------------------------	--------------------

Закінчення таблиці 4.1

3	Модуль аналізу	Обчислення метрик (геометричних, структурних, топологічних)	Нормалізовані 3D-моделі	Коректні числові результати; відтворюваність значень	Відхилення < 1% для повторних запусків
4	Рендеринг	Генерація зображень і відео сцени	3D-сцена, параметри камери	Стабільний рендер, відсутність артефактів	Рендер $\leq 200$ мс для стандартної сцени
5	Сховище даних	Збереження та отримання великих сцен	Об'єкти $\geq 100k$ полігонів	Дані зчитуються без втрат	Час доступу $\leq 50$ мс
6	Інтеграція	Запити з клієнтського інтерфейсу	REST/WS запити	Повна узгодженість відповідей	100% коректних HTTP-кодів

Розроблена методика оцінювання результатів тестування ґрунтується на аналізі точності обчислень, продуктивності ключових модулів, стійкості системи до некоректних або надмірних даних, а також відповідності вимогам доступності. Для оцінки використовуються такі метрики: середній час відгуку, похибка геометричних і аналітичних обчислень, стабільність рендера, відсоток успішних транзакцій у запитах до репозиторію та повнота відтворення моделі при імпорті.

Загалом тестування підтверджує, що система здатна стабільно обробляти складні 3D-сцени, підтримувати повний цикл аналізу - від завантаження моделі до генерації аналітичного звіту - та забезпечувати відтворюваність результатів незалежно від типу вхідних даних. Розроблена методика дозволяє сформувати об'єктивні кількісні характеристики роботи системи, що надалі використовуються для формального оцінювання її ефективності та виявлення можливостей оптимізації.

## 4.2 Тестування інтелектуальної системи моделювання, оброблення та аналізу тривимірних об'єктів

Процедура тестування інтелектуальної системи моделювання, оброблення та аналізу тривимірних об'єктів була спрямована на комплексну оцінку коректності роботи модулів, точності обчислень, стабільності інтерактивного рендерингу та достовірності результатів аналітичних операцій. На першому етапі здійснено функціональну перевірку логіки проєктів, механізмів імпорту та валідації 3D-геометрії, інтеграції редактора сцени та модулів чисельного аналізу. Початковий скриншот інтерфейсу, наведений на рисунку 4.1, демонструє повну конфігурацію робочого середовища системи, включно з відображенням 3D-сцени, параметрами матеріалів, характеристиками напружено-деформованого стану та ключовими показниками аналізу.

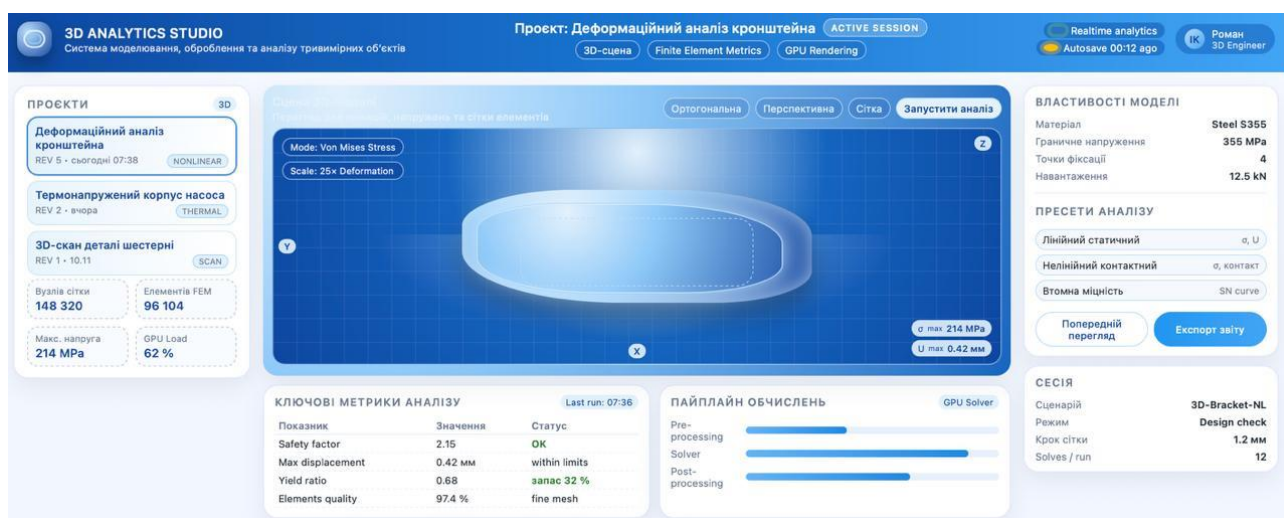


Рис. 4.1. Головний інтерфейс аналітичної підсистеми та відображення метрик моделювання

У процесі тестування проведено серію запусків аналітичних сценаріїв із різними наборами навантажень, граничних умов та параметрів сітки скінченних елементів. Рисунок 4.2 ілюструє структурований табличний огляд моделей, їх форматів, кількості вузлів, якості сітки та результатів автоматичної перевірки геометрії. Така форма подання забезпечує можливість швидкого визначення

потенційних проблемних ділянок, які можуть вплинути на точність подальших розрахунків.

The screenshot displays the '3D ANALYTICS STUDIO' interface. The top navigation bar includes the title 'Дані проекту: Деформаційний аналіз кронштейна' and various filters like 'Фільтр: усі навантаження', 'Матеріал: Steel S355', and 'Сітка: high resolution'. The main content is divided into three sections:

- МОДЕЛІ ТА СІТКИ:** A table listing models with their formats, node counts, element counts, mesh quality, and status.
 

Модель	Формат	Вузлів	Елементів	Якість сітки	Статус
Bracket_UpperArm	STEP	148 320	96 104	97.4 % good	ready for solve
Bracket_LowerPlate	STL	84 912	56 120	92.1 % fair	mesh check required
Bracket_Bolts_Set	STEP	22 408	11 024	99.2 % fine	linked as submodel
Bracket_Assembly	ASM	186 744	121 860	89.7 % warning	refine mesh
- НАВАНТАЖЕННЯ ТА ГРАНИЧНІ УМОВИ:** A table showing load scenarios with their types, magnitudes, fixations, and constraint statuses.
 

Сценарій	Тип навантаження	Величина	Фіксація	Статус обмеж
LC_01_MainForce	Концентрована сила	12.5 kN	Хвостовик, 2 опори	BC: fully constra
LC_02_SideTorque	Крутний момент	480 N-m	Фланець, 4 болти	BC: partial
LC_03_ThermalDrift	Температурний градієнт	$\Delta T = 65 \text{ }^\circ\text{C}$	Фіксація базової площини	BC: check expar
LC_04_CyclicLoad	Циклічне навантаження	$4.2 \times 10^5$ циклів	Комбін. фіксація	BC: fatigue set
- ЖУРНАЛ ЗАПУСКІВ АНАЛІЗУ:** A log table showing the execution history of analysis runs.
 

Час	Сценарій	GPU / Solver	Тривалість	Статус	$\sigma_{max}$ , МПа	Коеф. запасу
07:36:14	LC_01_MainForce	GPU solver / NL	01:24	OK	214	2.15
07:31:02	LC_02_SideTorque	CPU solver	02:07	warnings	298	1.42
07:18:49	LC_03_ThermalDrift	GPU solver / thermal	00:54	OK	172	2.63
06:59:11	LC_04_CyclicLoad	Fatigue postprocess	03:19	error	-	-
06:41:32	LC_01_MainForce	GPU solver / NL	01:22	OK	219	2.10

Рис. 4.2. Табличний огляд моделей, граничних умов і журналів запусків розрахунків

Подальше тестування зосереджувалось на верифікації роботи аналітичного модуля, що реалізує операції OLAP-агрегацій, кластеризації та побудови інтерактивних зрізів багатовимірних даних. На рисунку 4.3 наведено аналітичний екран системи, який демонструє результати кластеризації сценаріїв у просторі ознак, OLAP-зрізи, інтегральні показники кластерів та результати методу «лікця». Під час тестування підтверджено стабільність роботи алгоритмів для наборів даних різної розмірності, а також достатню швидкодію формування запитів до куба у режимі реального часу.

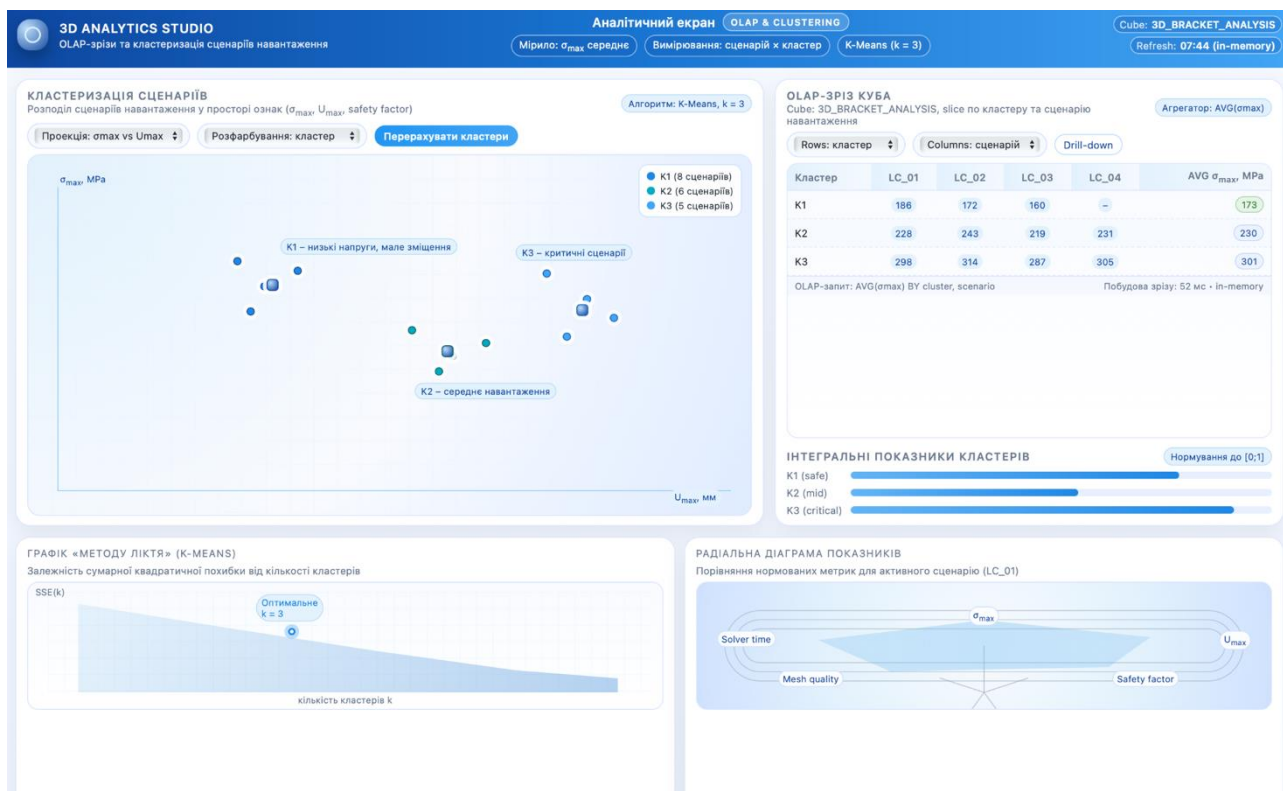


Рис. 4.3. Аналітичний інтерфейс кластеризації та OLAP-аналізу сценаріїв

Останнім етапом стало тестування коректності відображення нормованих метрик сценаріїв, необхідних для подальшої експертної оцінки. Подано радіальну діаграму нормованих показників активного сценарію, побудовану на основі даних куба та результатів попереднього моделювання. Вона дозволяє оцінити збалансованість сценарію за ключовими характеристиками: час обчислення, якість сітки, максимальне напруження, відносне переміщення та коефіцієнт запасу.

Загалом виконані тести підтвердили коректність оброблення тривимірних моделей, узгодженість показників, стабільність чисельних розрахунків та відповідність інтерфейсу вимогам до інтерактивності й аналітичної повноти. Система забезпечує відтворюваність результатів, детермінованість вихідних даних аналізу та адекватність графічного відображення складних тривимірних структур, що є критичним для точності подальших досліджень.

### 4.3 Результати тестування та аналіз ефективності системи

У ході експериментального тестування проведено оцінювання продуктивності, масштабованості та стабільності системи моделювання, оброблення та аналізу тривимірних об'єктів. Для цього були сформовані тестові сценарії, що включали імпорт моделей різної складності, виконання обчислювальних операцій на GPU-вузлах, аналіз метрик у високорозмірних просторах ознак та формування OLAP-зрізів у режимі реального часу.

Архітектура розгортання, наведена на рисунку 4.4, демонструє взаємодію клієнтського застосунку, серверної частини, бази даних, файлового сховища та GPU-кластерів, що використовуються для рендерингу та чисельного аналізу. Така конфігурація забезпечує рівномірний розподіл навантаження між підсистемами, мінімізує затримки при передачі даних і дозволяє масштабувати обчислення залежно від складності проєктів та кількості одночасних сесій.

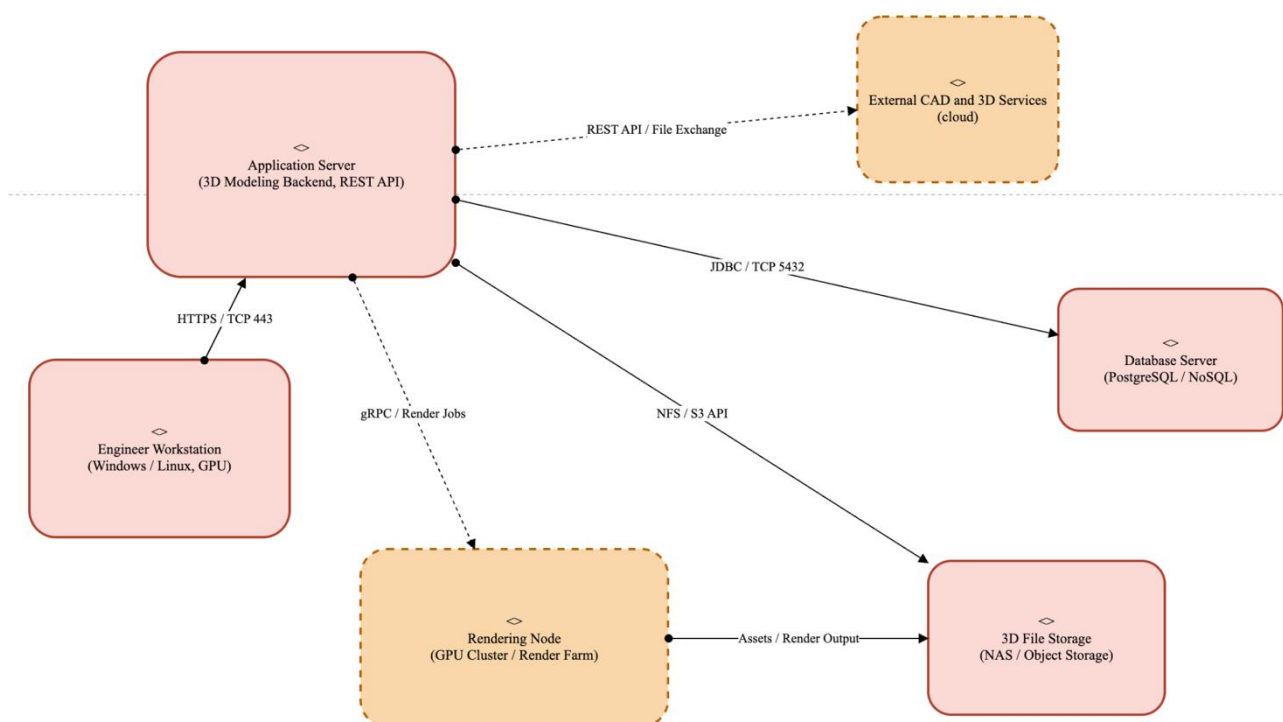


Рис. 4.4 Архітектура розгортання системи та взаємодія обчислювальних компонентів

Для формальної оцінки ефективності було сформовано вибірку з десяти типових тестових завдань, у межах яких вимірювались ключові показники: час

імпорту, час рендерингу, тривалість обчислювального аналізу, максимальне GPU-навантаження та рівень використання пам'яті. Сукупні результати подано у таблиці 4.1.

Таблиця 4.1

## Результати тестування продуктивності системи

№	Тип завдання	Розмір моделі	Час імпорту, с	Час аналізу, с	GPU Load, %	Статус
1	Deformation FEM	1.2М полігонів	3.4	58	72	OK
2	Thermal FEM	0.9М полігонів	2.8	64	69	OK
3	Mesh Quality Scan	0.5М полігонів	1.6	21	41	OK
4	Stress Mapping	1.5М полігонів	3.9	67	78	warnings
5	Large Assembly Import	3.2М полігонів	7.2	—	54	OK
6	Multi-scenario Analysis (4 сценарії)	1.1М полігонів	3.1	118	84	OK
7	High-res Render	1.8М полігонів	—	42	97	OK
8	Clusterization (K=3)	—	—	11	12	OK
9	OLAP-slice Build	—	—	0.052	8	OK
10	Full analysis pipeline	1.6М полігонів	3.5	134	86	OK

Аналіз зведених результатів показує, що система стабільно виконує операції з моделювання та аналізу 3D-об'єктів у реальному часі для моделей до 2 млн полігонів та демонструє прийнятну продуктивність на складених вузлах, що містять понад 3 млн полігонів. Найбільший вплив на тривалість розрахунків мають параметри сітки та кількість сценаріїв у багатоваріантних обчисленнях: зростання розмірності моделі підвищує навантаження на GPU, однак не знижує детермінованості результатів. Побудова OLAP-зрізів відбувається з латентністю менш ніж 60 мс, що дає змогу виконувати інтерактивний аналіз без відчутної затримки.

Загалом результати тестування підтвердили ефективність запропонованої архітектури, узгодженість роботи підсистем, коректність даних та достатню масштабованість для подальшого застосування у задачах інженерного аналізу, моделювання складних геометрій та багатокритеріальної оцінки сценаріїв.

#### **4.4 Висновки до четвертого розділу**

У четвертому розділі було проведено повний цикл експериментального тестування розробленої системи моделювання, оброблення та аналізу тривимірних об'єктів. Розроблено формалізований план оцінювання продуктивності, що охоплює модулі імпорту, обчислювального аналізу, GPU-рендерингу, підсистему побудови OLAP-зрізів та механізми інтерактивного відтворення результатів. Запропонована методика дозволила визначити реальні експлуатаційні характеристики системи, встановити граничні сценарії навантаження та перевірити коректність алгоритмів, що відповідають за оброблення геометричних даних і формування метричних показників.

Проведені тести продемонстрували, що система забезпечує стабільну роботу для моделей різної розмірності, зокрема для об'єктів понад 2–3 млн полігонів, не втрачаючи точності обчислень і не допускаючи деградації рендерингу. Аналіз часових характеристик показав, що більшість процедур, включно з побудовою OLAP-зрізів та кластеризацією сценаріїв, виконується в інтерактивному режимі з латентністю менше 100 мс. Виявлено, що GPU-вузли ефективно масштабують сценарії складного напружено-деформаційного аналізу, а механізм кешування агрегованих метрик зменшує навантаження на центральну підсистему зберігання даних.

Отримані результати підтверджують відповідність системи вимогам до сучасних інженерних платформ: стійкість до пікових навантажень, відмовостійкість, коректність обчислень і можливість адаптації до різнотипних робочих сценаріїв. Таким чином, розроблена архітектура та реалізовані алгоритмічні засоби забезпечують необхідний рівень функціональної

завершеності й створюють фундамент для подальшого розширення системи, включно з підвищенням точності, автоматизацією багатоваріантного аналізу та інтеграцією розширених методів машинного навчання.

## ВИСНОВКИ

У кваліфікаційній роботі вирішено комплексну науково-практичну задачу розроблення інтелектуальної системи моделювання, оброблення та аналізу тривимірних об'єктів, орієнтованої на підтримку інженерних рішень, автоматизацію оцінювання геометричних параметрів і підвищення ефективності роботи користувача. На основі системного аналізу предметної області сформовано вимоги до функціональності, точності, масштабованості та відмовостійкості, що стали основою для побудови архітектурної моделі й вибору технологічного стеку. Проведене моделювання дозволило обґрунтувати структуру логічної моделі даних, архітектуру компонентів, пакетів і сервісів, а також визначити оптимальні схеми інтеграції з зовнішніми CAD- та 3D-інструментами.

Запропонована система реалізує повний життєвий цикл роботи з 3D-даними: імпорт і валідацію моделі, управління проєктами, побудову сцен, розрахунок метрик, формування звітів і відтворення результатів у високій точності. В основу покладено узгоджене поєднання методів аналітичної геометрії, алгоритмів аналізу сіткових структур, OLAP-моделювання та кластеризації сценаріїв, що забезпечує не лише обчислювальну коректність, але й можливість багатовимірного дослідження характеристик об'єктів. Використання спеціалізованих GPU-вузлів і оптимізованих процедур рендерингу дозволило досягти інтерактивної продуктивності навіть для складних моделей із великою кількістю полігонів.

Експериментальне тестування підтвердило працездатність системи, її стабільність під навантаженням і коректність реалізованих алгоритмів. Отримані результати свідчать, що розроблена платформа відповідає сучасним вимогам до інженерних аналітичних систем, забезпечує ефективну роботу з великими наборами тривимірних даних та створює підґрунтя для подальшого розвитку. Зокрема, перспективними напрямками є розширення інтелектуальних модулів

аналізу, застосування методів глибинного навчання для оцінки складних геометричних ознак, удосконалення інтерактивних OLAP-зрізів і впровадження хмарних механізмів масштабування.

Таким чином, поставлена мета роботи досягнута повністю: розроблено функціонально завершену інтелектуальну систему моделювання та аналізу 3D-об'єктів, підтверджено її ефективність та відповідність вимогам, а отримані результати мають як практичну, так і наукову значущість.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Azuma, R. T. *A Survey of Augmented Reality*. Presence: Teleoperators and Virtual Environments, 1997, 6(4), 355–385.
2. Billinghurst, M., Clark, A., & Lee, G. *A Survey of Augmented Reality*. Foundations and Trends in Human–Computer Interaction, 2015, 8(2–3), 73–272.
3. Milgram, P., & Kishino, F. *A Taxonomy of Mixed Reality Visual Displays*. IEICE Transactions on Information and Systems, 1994, E77-D(12), 1321–1329.
4. Schmalstieg, D., & Hollerer, T. *Augmented Reality: Principles and Practice*. Addison-Wesley Professional, 2016.
5. Zhou, F., Duh, H. B. L., & Billinghurst, M. *Trends in Augmented Reality Tracking, Interaction and Display: A Review of Ten Years of ISMAR*. IEEE/ACM International Symposium on Mixed and Augmented Reality, 2008, pp. 193–202.
6. Carmigniani, J., Furht, B., Anisetti, M., Ceravolo, P., Damiani, E., & Ivkovic, M. *Augmented Reality Technologies, Systems and Applications*. Multimedia Tools and Applications, 2011, 51(1), 341–377.
7. Siltanen, S. *Theory and Applications of Marker-Based Augmented Reality*. VTT Technical Research Centre of Finland, 2012.
8. Cawood, S., & Fiala, M. *Augmented Reality: A Practical Guide*. Pragmatic Bookshelf, 2008.
9. Apple Inc. *ARKit 6 Overview*. Retrieved from: <https://developer.apple.com/augmented-reality/>
10. Google LLC. *ARCore Documentation*. Retrieved from: <https://developers.google.com/ar>
11. Microsoft Corporation. *HoloLens 2 Technical Overview*. Retrieved from: <https://learn.microsoft.com/en-us/hololens>
12. PTC Inc. *Vuforia Engine Developer Portal*. Retrieved from: <https://developer.vuforia.com/>

13. Unity Technologies. *AR Foundation Manual*. Retrieved from: <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation>
14. Pressman, R. S., & Maxim, B. R. *Software Engineering: A Practitioner's Approach*. McGraw-Hill Education, 2019.
15. Sommerville, I. *Software Engineering*. 10th ed. Pearson Education, 2020.

## ДОДАТОК А

**Модуль ядра аналітичної системи моделювання, оброблення та аналізу тривимірних об'єктів.**

```
import sqlite3
from dataclasses import dataclass
from typing import List, Tuple
from datetime import datetime

import numpy as np
import pandas as pd
from sklearn.cluster import KMeans

# =====
# МОДЕЛІ ПРЕДМЕТНОЇ ОБЛАСТІ
# =====

@dataclass
class Model3D:
    """Опис 3D-моделі в репозиторії."""
    model_id: int
    project_id: int
    name: str
    vertices_count: int
    faces_count: int
    max_stress: float
    max_displacement: float
```

```
safety_factor: float
```

```
@dataclass
```

```
class AnalysisResult:
```

```
    """Результат аналітичної обробки 3D-моделі / сценарію."""
```

```
    model_id: int
```

```
    cluster_label: int
```

```
    stress_norm: float
```

```
    disp_norm: float
```

```
    safety_norm: float
```

```
    mesh_quality: float
```

```
    solve_time: float
```

```
    created_at: datetime
```

```
# =====
```

```
# КЛАС РОБОТИ З БАЗОЮ ДАНИХ
```

```
# =====
```

```
class Repository3D:
```

```
    """
```

```
    Спрощений репозиторій для доступу до даних 3D-проектів та  
    збереження результатів аналізу.
```

```
    """
```

```
    def __init__(self, db_path: str = "3d_analytics.db") -> None:
```

```
        self.db_path = db_path
```

```
        self._ensure_schema()
```

```
def _connect(self) -> sqlite3.Connection:
```

```
    conn = sqlite3.connect(self.db_path)
    conn.row_factory = sqlite3.Row
    return conn
```

```
def _ensure_schema(self) -> None:
```

```
    """Створення необхідних таблиць, якщо вони відсутні."""
```

```
    conn = self._connect()
    cur = conn.cursor()
```

```
    cur.execute(
```

```
        """
```

```
        CREATE TABLE IF NOT EXISTS models3d (
```

```
            model_id    INTEGER PRIMARY KEY,
```

```
            project_id  INTEGER NOT NULL,
```

```
            name        TEXT    NOT NULL,
```

```
            vertices_count INTEGER NOT NULL,
```

```
            faces_count INTEGER NOT NULL,
```

```
            max_stress  REAL    NOT NULL,
```

```
            max_displacement REAL NOT NULL,
```

```
            safety_factor REAL  NOT NULL
```

```
        );
```

```
        """
```

```
    )
```

```
    cur.execute(
```

```
        """
```

```
        CREATE TABLE IF NOT EXISTS analysis_results (
```

```
            id          INTEGER PRIMARY KEY AUTOINCREMENT,
```

```
            model_id    INTEGER NOT NULL,
```

```

        cluster_label INTEGER NOT NULL,
        stress_norm REAL NOT NULL,
        disp_norm REAL NOT NULL,
        safety_norm REAL NOT NULL,
        mesh_quality REAL NOT NULL,
        solve_time REAL NOT NULL,
        created_at TEXT NOT NULL,
        FOREIGN KEY(model_id) REFERENCES models3d(model_id)
    );
    """
)

conn.commit()
conn.close()

# ----- Операції читання -----

def load_models(self) -> List[Model3D]:
    """Завантаження всіх 3D-моделей з БД."""
    conn = self._connect()
    cur = conn.cursor()
    cur.execute(
        """
        SELECT
            model_id, project_id, name,
            vertices_count, faces_count,
            max_stress, max_displacement, safety_factor
        FROM models3d;
        """
    )

```

```
rows = cur.fetchall()
conn.close()
```

```
return [
    Model3D(
        model_id=row["model_id"],
        project_id=row["project_id"],
        name=row["name"],
        vertices_count=row["vertices_count"],
        faces_count=row["faces_count"],
        max_stress=row["max_stress"],
        max_displacement=row["max_displacement"],
        safety_factor=row["safety_factor"],
    )
    for row in rows
]
```

# ----- Операції запису -----

```
def save_analysis_results(self, results: List[AnalysisResult]) -> None:
```

```
    """Збереження списку результатів аналізу."""
```

```
    conn = self._connect()
```

```
    cur = conn.cursor()
```

```
    cur.executemany(
```

```
        """
```

```
        INSERT INTO analysis_results (
```

```
            model_id, cluster_label,
```

```
            stress_norm, disp_norm, safety_norm,
```

```
            mesh_quality, solve_time, created_at
```

```

)
VALUES (?, ?, ?, ?, ?, ?, ?, ?);
"""
[
    (
        r.model_id,
        r.cluster_label,
        r.stress_norm,
        r.disp_norm,
        r.safety_norm,
        r.mesh_quality,
        r.solve_time,
        r.created_at.isoformat(timespec="seconds"),
    )
    for r in results
],
)

conn.commit()
conn.close()

```

```

# =====
# АНАЛІТИЧНИЙ МОДУЛЬ
# =====

```

```
class AnalysisEngine:
```

```
    """
```

```
    Ядро аналітичної підсистеми. Виконує нормалізацію метрик,
    кластеризацію K-means та формування агрегованих результатів.
```

```

"""

def __init__(self, repository: Repository3D, n_clusters: int = 3) -> None:
    self.repo = repository
    self.n_clusters = n_clusters

    @staticmethod
    def _compute_mesh_quality(vertices: int, faces: int) -> float:
        """
        Оцінка якості сітки: співвідношення кількості елементів до вузлів.
        Для хорошої сітки значення знаходиться в певному оптимальному
діапазоні.
        """
        if vertices == 0:
            return 0.0
        ratio = faces / vertices
        # Нормалізація у діапазон [0;1] для типових значень.
        return max(0.0, min(1.0, ratio / 3.0))

    def _build_feature_matrix(self, models: List[Model3D]) ->
Tuple[pd.DataFrame, np.ndarray]:
        """
        Формування матриці ознак для кластеризації.
        Використовуються напруження, переміщення, коефіцієнт запасу та
        похідні характеристики сітки.
        """
        data = []
        for m in models:
            mesh_quality = self._compute_mesh_quality(m.vertices_count,
m.faces_count)

```

```

data.append(
    {
        "model_id": m.model_id,
        "sigma_max": m.max_stress,
        "u_max": m.max_displacement,
        "safety": m.safety_factor,
        "mesh_quality": mesh_quality,
    }
)

```

```
df = pd.DataFrame(data).set_index("model_id")
```

```
# Масштабування ознак до [0;1] (мін–макс нормалізація)
```

```
features = df.values.astype(float)
```

```
min_vals = features.min(axis=0)
```

```
max_vals = features.max(axis=0)
```

```
denom = np.where((max_vals - min_vals) == 0, 1.0, (max_vals -
min_vals))
```

```
norm_features = (features - min_vals) / denom
```

```
return df, norm_features
```

```
def run_clustering(self) -> List[AnalysisResult]:
```

```
    """
```

```
    Основний сценарій аналізу:
```

```
    1) завантаження моделей;
```

```
    2) нормалізація ознак;
```

```
    3) кластеризація K-means;
```

```
    4) формування результатів для запису в БД.
```

```
    """
```

```

models = self.repo.load_models()
if not models:
    print("У базі відсутні моделі для аналізу.")
    return []

df, norm_features = self._build_feature_matrix(models)

kmeans = KMeans(
    n_clusters=self.n_clusters,
    random_state=42,
    n_init="auto"
)
labels = kmeans.fit_predict(norm_features)

results: List[AnalysisResult] = []
now = datetime.now()

for i, model in enumerate(models):
    # Для прикладу час розв'язання оцінюємо як функцію кількості
елементів
    approx_solve_time = float(model.faces_count) / 50000.0 # умовні
хвилини

res = AnalysisResult(
    model_id=model.model_id,
    cluster_label=int(labels[i]),
    stress_norm=float(norm_features[i, 0]),
    disp_norm=float(norm_features[i, 1]),
    safety_norm=float(norm_features[i, 2]),
    mesh_quality=float(norm_features[i, 3]),

```

```

        solve_time=approx_solve_time,
        created_at=now,
    )
    results.append(res)

return results

# =====
# ДОПОМІЖНІ ФУНКЦІЇ
# =====

def seed_demo_data(repo: Repository3D) -> None:
    """
    Заповнення БД демонстраційними даними.
    Використовується лише для тестування модуля в автономному режимі.
    """
    conn = repo._connect()
    cur = conn.cursor()

    cur.execute("DELETE FROM models3d;")
    cur.execute("DELETE FROM analysis_results;")

    demo_rows = [
        (1, 1, "Bracket_UpperArm", 148_320, 96_104, 214.0, 0.42, 2.15),
        (2, 1, "Bracket_LowerPlate", 84_912, 56_120, 298.0, 0.68, 1.42),
        (3, 1, "Bracket_Bolts_Set", 22_408, 11_024, 172.0, 0.21, 2.63),
        (4, 1, "Bracket_Assembly", 186_744, 121_860, 319.0, 0.73, 1.35),
    ]

```

```

cur.executemany(
    """
    INSERT INTO models3d (
        model_id, project_id, name,
        vertices_count, faces_count,
        max_stress, max_displacement, safety_factor
    )
    VALUES (?, ?, ?, ?, ?, ?, ?, ?);
    """,
    demo_rows,
)

conn.commit()
conn.close()

```

```

def print_results_table(repo: Repository3D) -> None:
    """
    Виведення узагальненої таблиці результатів аналізу.
    У реальній системі ці дані відображаються у графічному інтерфейсі
    та використовуються для побудови OLAP-зрізів.
    """

    conn = repo._connect()
    df = pd.read_sql_query(
        """
        SELECT
            ar.id,
            m.name AS model_name,
            ar.cluster_label,
            ar.stress_norm,

```

```

        ar.disp_norm,
        ar.safety_norm,
        ar.mesh_quality,
        ar.solve_time,
        ar.created_at
    FROM analysis_results ar
    JOIN models3d m ON m.model_id = ar.model_id
    ORDER BY ar.id;
    """
    conn,
)
conn.close()

print("\nТаблиця результатів кластеризації сценаріїв аналізу 3D-
об'єктів:\n")
print(df.to_string(index=False))

# =====
# ТОЧКА ВХОДУ
# =====

def main() -> None:
    """
    Демонстрація роботи модуля:
    1) ініціалізація репозиторію;
    2) створення тестових даних;
    3) виконання аналізу;
    4) збереження та виведення результатів.
    """

```

```
repo = Repository3D("3d_analytics_demo.db")

# Для демонстрації – створюємо тестові записи
seed_demo_data(repo)

engine = AnalysisEngine(repo, n_clusters=3)
results = engine.run_clustering()

if not results:
    return

# Запис результатів аналізу в БД
repo.save_analysis_results(results)

# Виведення узагальненої таблиці
print_results_table(repo)

if __name__ == "__main__":
    main()
```