

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: Моделювання та інтеграція знань у графових структурах для виявлення зв'язків між гетерогенними сутностями

Виконав: студент VI курсу, групи СНнм-61
спеціальності 122 Комп'ютерні науки
(шифр і назва спеціальності)

Гашинський Р.І.
(підпис) (прізвище та ініціали)

Керівник Фриз М. Є.
(підпис) (прізвище та ініціали)

Нормоконтроль Боднарчук І.О.
(підпис) (прізвище та ініціали)

Завідувач кафедри Боднарчук І.О.
(підпис) (прізвище та ініціали)

Рецензент Боднарчук І.О.
(підпис) (прізвище та ініціали)

Тернопіль
2026

АНОТАЦІЯ

Моделювання та інтеграція знань у графових структурах для виявлення зв'язків між гетерогенними сутностями // Кваліфікаційна робота освітнього ступеня «Магістр» // Гашинський Роман Ігорович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНм-61 // Тернопіль, 2026 // С. 73, рис. – 17, табл. – 11, кресл. – 0, додат. – 2, бібліогр. – 55.

Ключові слова: графи знань, гетерогенні сутності, RDF, JSON-LD, графова структура, інтеграція знань.

Кваліфікаційна робота присвячена дослідженню та розробці підходу до моделювання й інтеграції знань у графових структурах для виявлення зв'язків між гетерогенними сутностями.

У першому розділі проаналізовано теоретичні основи представлення знань, графових структур, гетерогенних даних, RDF, JSON-LD та методів інтеграції знань.

У другому розділі описано предметну область, визначено типи сутностей і зв'язків, сформовано модель JSON-LD даних, архітектуру системи та алгоритм інтеграції знань у графову модель.

У третьому розділі наведено реалізацію програмної системи, результати побудови RDF-графа, візуалізації графової моделі, аналізу центральності та пошуку прямих і непрямих зв'язків між сутностями.

Об'єкт дослідження: процеси моделювання та інтеграції знань у графових структурах.

Предмет дослідження: методи, моделі та програмні засоби представлення, інтеграції й аналізу зв'язків між гетерогенними сутностями у графових структурах.

ANNOTATION

Modeling and Integration of Knowledge in Graph Structures for Identifying Relationships Between Heterogeneous Entities // Qualification work of the educational level "Master" // Roman Gashynskiy // Ternopil Ivan Pulyu National Technical University, Computer and Information Systems and Software Engineering Faculty, Computer Sciences Department, group SNnm-61 // Ternopil, 2026 // P. 73, fig. - 17, tabl. - 11, chair. - 0, annexes. – 2, references - 55.

Keywords: knowledge graphs, heterogeneous entities, RDF, JSON-LD, graph structure, knowledge integration.

The qualification work is devoted to the study and development of an approach to knowledge modeling and integration in graph structures for identifying relationships between heterogeneous entities.

The first chapter analyzes the theoretical foundations of knowledge representation, graph structures, heterogeneous data, RDF, JSON-LD, and methods of knowledge integration.

The second chapter describes the subject area, defines the types of entities and relationships, and presents the JSON-LD data model, the system architecture, and the algorithm for integrating knowledge into a graph model.

The third chapter presents the implementation of the software system, the results of RDF graph construction, visualization of the graph model, centrality analysis, and the search for direct and indirect relationships between entities.

Object of the research: the processes of knowledge modeling and integration in graph structures.

Subject of the research: methods, models, and software tools for representing, integrating, and analyzing relationships between heterogeneous entities in graph structures.

ПЕРЕЛІК СКОРОЧЕНЬ І ТЕРМІНІВ

API (англ. Application Programming Interface) – прикладний програмний інтерфейс.

RDF (англ. Resource Description Framework) – модель подання даних у вигляді трійок “суб’єкт – предикат – об’єкт”.

JSON-LD (англ. JavaScript Object Notation for Linked Data) – формат подання пов’язаних даних на основі JSON.

URI (англ. Uniform Resource Identifier) – уніфікований ідентифікатор ресурсу.

SPARQL – мова запитів до RDF-графів.

RDFLib – бібліотека Python для роботи з RDF-графами та JSON-LD даними.

NetworkX – бібліотека Python для створення, аналізу та обробки графових структур.

Streamlit – фреймворк Python для створення інтерактивних вебінтерфейсів.

Граф знань – графова модель, у якій сутності та зв’язки між ними мають семантичне значення.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД МЕТОДІВ МОДЕЛЮВАННЯ ТА ІНТЕГРАЦІЇ ЗНАНЬ У ГРАФОВИХ СТРУКТУРАХ.....	11
1.1 Представлення знань у сучасних інформаційних системах	11
1.2 Графові структури та графи знань як засіб моделювання знань	13
1.3 Гетерогенні дані та рівні гетерогенності у графових структурах.....	15
1.4 RDF, JSON-LD та семантичне подання даних.....	17
1.5 Методи інтеграції знань у графових моделях.....	20
1.6 Методи виявлення зв'язків між гетерогенними сутностями у графових моделях.....	21
1.7 Постановка проблеми та обґрунтування напряму дослідження	23
1.8 Висновки до першого розділу	23
РОЗДІЛ 2. МЕТОДИ ТА ЗАСОБИ МОДЕЛЮВАННЯ Й ІНТЕГРАЦІЇ ЗНАНЬ У ГРАФОВІЙ СТРУКТУРІ.....	25
2.1 Опис предметної області графової структури	25
2.2 Визначення типів гетерогенних сутностей у графовій структурі.....	27
2.3 Визначення типів зв'язків між сутностями у графовій структурі	28
2.4 Модель вхідних JSON-LD даних у графовій структурі.....	30
2.5 Моделювання пояснювальної записки як складеної сутності	32
2.6 Архітектура програмної системи інтеграції знань	34
2.7 Алгоритм інтеграції JSON-LD описів у графову структуру	36
2.8 Методика виявлення зв'язків між гетерогенними сутностями у графовій структурі	39
2.9 Висновки до другого розділу.....	41
РОЗДІЛ 3. РЕАЛІЗАЦІЯ СИСТЕМИ ТА АНАЛІЗ РЕЗУЛЬТАТІВ ІНТЕГРАЦІЇ ЗНАНЬ.....	42
3.1 Засоби реалізації програмної системи інтеграції знань	42
3.2 Структура програмного проєкту інтеграції знань	43

	7
3.3 Реалізація завантаження та обробки JSON-LD файлів	44
3.4 Побудова RDF-графа та графової моделі NetworkX	46
3.5 Візуалізація графової моделі знань	48
3.6 Формування таблиць сутностей і зв'язків графової структури	49
3.7 Аналіз складу графа знань	50
3.8 Виявлення прямих і непрямих зв'язків між сутностями графа знань .	52
3.9 Аналіз центральності вузлів графа знань	54
3.10 Оцінювання результатів реалізації системи інтеграції знань	56
3.11 Висновки до третього розділу	58
РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	59
4.1 Принципи ергономіки робочого місця	59
4.2 Правила електробезпеки при роботі з електронними пристроями	64
4.3 Висновки до четвертого розділу	67
ВИСНОВКИ.....	68
ПЕРЕЛІК ДЖЕРЕЛ	70
ДОДАТКИ	

ВСТУП

Актуальність теми. У сучасному інформаційному суспільстві обсяги даних постійно зростають, а їх структура та зміст стають дедалі складнішими. Дані надходять із різних джерел, описують об'єкти різної природи та відображають різні аспекти предметних областей. У таких умовах важливого значення набуває не лише накопичення інформації, а й її змістове впорядкування, інтеграція та подання у формі, придатній для подальшого аналізу [1].

Однією з ключових проблем сучасних інформаційних систем є робота з гетерогенними сутностями. Такі сутності можуть відрізнятися за структурою, властивостями, роллю в предметній області та характером зв'язків з іншими об'єктами. Через це традиційні підходи до зберігання та обробки даних не завжди забезпечують достатній рівень гнучкості для виявлення складних взаємозалежностей між об'єктами. Особливо актуальною є проблема встановлення не лише прямих, а й опосередкованих зв'язків, які можуть проявлятися через проміжні сутності [2].

Перспективним напрямком розв'язання цієї проблеми є використання графових структур для моделювання та інтеграції знань. Графове подання дозволяє відображати предметну область як сукупність сутностей і зв'язків між ними. Такий підхід забезпечує наочність, гнучкість і можливість аналізу складних структур знань, у яких важливими є не лише окремі об'єкти, а й система їх взаємодії [3].

Моделювання знань у графових структурах є особливо доцільним у випадках, коли необхідно об'єднати різноманітні фрагменти інформації в єдину цілісну модель. Це дозволяє подолати фрагментарність даних, узгодити їх у межах спільної структури та створити основу для подальшого аналізу зв'язків між сутностями. Завдяки цьому графові моделі можна використовувати для підтримки прийняття рішень, інтелектуального аналізу даних, побудови інформаційних систем і формалізації складних предметних областей.

Тому дослідження методів моделювання та інтеграції знань у графових структурах для виявлення зв'язків між гетерогенними сутностями є актуальним напрямком у галузі комп'ютерних наук.

Мета і задачі дослідження. Метою даної кваліфікаційної роботи освітнього ступеня «Магістр» є дослідження та розробка підходу до моделювання й інтеграції знань у графових структурах для виявлення зв'язків між гетерогенними сутностями.

Для досягнення поставленої мети потрібно виконати ряд завдань, зокрема:

- Проаналізувати підходи до представлення знань, гетерогенних даних і графових структур.
- Дослідити використання RDF і JSON-LD для семантичного подання та інтеграції знань.
- Розробити модель подання гетерогенних сутностей і зв'язків у єдиній графовій структурі.
- Спроекувати та реалізувати програмну систему для побудови, візуалізації й аналізу графа знань.
- Провести аналіз отриманих результатів і оцінити ефективність запропонованого підходу для виявлення зв'язків між гетерогенними сутностями.

Виконання зазначених завдань сприятиме розробленню підходу, який дозволяє представляти різноманітні знання у вигляді єдиної графової структури та використовувати її для аналізу зв'язків між об'єктами предметної області.

Об'єкт дослідження – процеси моделювання та інтеграції знань у графових структурах.

Предмет дослідження – методи, моделі та програмні засоби представлення, інтеграції й аналізу зв'язків між гетерогенними сутностями у графових структурах.

Методи дослідження. У роботі використано методи аналізу та узагальнення наукових джерел, методи теорії графів, методи моделювання знань,

методи структурного аналізу предметної області, а також методи програмної реалізації та експериментальної перевірки отриманих результатів.

Практичне значення одержаних результатів. Практичне значення одержаних результатів полягає у розробленні програмних засобів, які дозволяють формувати графову модель знань, інтегрувати в ній різноманітні сутності та аналізувати зв'язки між ними. Запропонований підхід може бути використаний для моделювання складних предметних областей, у яких важливо відображати не лише окремі об'єкти, а й систему взаємозалежностей між ними.

Розроблена система дає змогу представити знання у вигляді взаємопов'язаної структури, здійснювати її візуальний аналіз, визначати ключові сутності та знаходити прямі й опосередковані зв'язки між об'єктами. Це підвищує зручність аналізу складних інформаційних середовищ і може бути основою для подальшого розвитку інтелектуальних інформаційних систем.

Наукова новизна одержаних результатів полягає в удосконаленні підходу до моделювання гетерогенних сутностей у графових структурах та інтеграції різноманітних знань у межах єдиної моделі. У роботі запропоновано покращений спосіб подання сутностей і зв'язків, який дозволяє враховувати різну природу об'єктів, різні типи відношень між ними та можливість виявлення опосередкованих залежностей.

Публікації. Основні результати кваліфікаційної роботи опубліковано у тезах. Здобувачу освітнього рівня магістр належить аналіз використання JSON-LD для семантичного подання та інтеграції знань [4] та у співавторстві дослідження підходів до моделювання та інтеграції знань у графових структурах [5].

Структура та обсяг роботи. Загальний обсяг роботи становить 73 сторінки. Робота містить 17 рисунків, 14 таблиць, 2 додатки. Перелік використаних джерел містить 55 найменувань.

РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД МЕТОДІВ МОДЕЛЮВАННЯ ТА ІНТЕГРАЦІЇ ЗНАНЬ У ГРАФОВИХ СТРУКТУРАХ

1.1 Представлення знань у сучасних інформаційних системах

Представлення знань є важливим напрямом розвитку сучасних інформаційних систем, оскільки воно забезпечує не лише збереження даних, а й формалізований опис об'єктів, їх властивостей і зв'язків у межах певної предметної області. На відміну від звичайного накопичення інформації, представлення знань орієнтоване на структурування, інтерпретацію та використання даних для подальшого аналізу, пошуку залежностей і підтримки прийняття рішень [6].

У сучасних інформаційних системах дані часто надходять із різних джерел, мають неоднорідну структуру та описують об'єкти різної природи. Тому важливими стають моделі, які дозволяють не лише зберігати записи, а й відображати змістові зв'язки між ними.

Одним із найпоширеніших способів організації даних є реляційна модель. Вона ефективна для зберігання структурованої інформації у вигляді таблиць, де кожна сутність має наперед визначений набір атрибутів. Проте в задачах, де необхідно моделювати різнотипні сутності та складні багаторівневі зв'язки, таблична модель має певні обмеження. Зокрема, зв'язки між об'єктами часто подаються через зовнішні ключі або проміжні таблиці, що ускладнює їхню інтерпретацію та аналіз [7].

Крім реляційного підходу, у сфері представлення знань використовуються логічні, фреймові, онтологічні та графові моделі. Логічні моделі дозволяють формально описувати твердження й правила виведення [8]. Однак вони можуть бути складними для великих і динамічних предметних областей [9]. Фреймові моделі описують об'єкти через набір слотів і значень, але не завжди достатньо гнучкі для відображення різнотипних зв'язків [10]. Онтологічні моделі дають змогу формалізувати поняття предметної області, їх ієрархію та відношення [11].

Під час побудови онтології важливо визначити основні класи, властивості та обмеження, які описують досліджувану предметну область [12].

Для узагальнення основних підходів до представлення знань у таблиці 1.1 наведено їх порівняння за короткою характеристикою, перевагами та обмеженнями.

Таблиця 1.1 – Порівняння підходів до представлення знань

Підхід	Коротка характеристика	Переваги	Обмеження
Реляційна модель	Подання даних у вигляді таблиць	Чітка структура, зручність зберігання	Менш зручна для складних багаторівневих зв'язків
Логічна модель	Опис знань через формальні твердження та правила	Висока формалізація, можливість логічного виведення	Складність реалізації для великих предметних областей
Фреймова модель	Опис об'єктів через слоти й атрибути	Зручна для типових об'єктів	Обмежена гнучкість для динамічних зв'язків
Онтологічна модель	Формалізація понять, класів і відношень	Семантична узгодженість	Потребує ретельного проєктування
Графова модель	Подання сутностей як вузлів, а зв'язків як ребер	Наочність, гнучкість, зручність аналізу зв'язків	Потребує коректного моделювання сутностей і відношень

Серед наведених підходів особливе місце займають графові моделі. У таких моделях сутності подаються як вершини графа, а зв'язки між ними – як ребра. Це дозволяє безпосередньо відображати структуру предметної області та аналізувати не лише окремі об'єкти, а й їхнє місце в системі взаємозв'язків.

У задачах інтеграції знань графові моделі є особливо корисними, оскільки дозволяють поєднувати фрагменти інформації з різних джерел у межах єдиної структури. Якщо окремі джерела описують різні частини предметної області, графова модель дає змогу об'єднати ці фрагменти через спільні сутності, ідентифікатори або семантичні зв'язки. У результаті формується цілісна модель знань, придатна для подальшого аналізу.

Отже, представлення знань є основою для структурованого та семантично узгодженого опису предметної області. Для задач, у яких важливо враховувати різнотипні об'єкти, складні зв'язки та інтеграцію даних із кількох джерел, доцільним є використання графових моделей.

1.2 Графові структури та графи знань як засіб моделювання знань

Графові структури є одним із найбільш зручних способів подання складних взаємозв'язків між об'єктами в інформаційних системах. На відміну від традиційних табличних моделей, у яких основний акцент робиться на збереженні даних у вигляді рядків і стовпців, графова модель орієнтована насамперед на відображення зв'язків. У такій структурі об'єкти подаються у вигляді вершин, а відношення між ними – у вигляді ребер. Завдяки цьому граф дозволяє не лише зберігати інформацію про окремі сутності, а й показувати, як саме вони пов'язані між собою [13].

У загальному вигляді граф можна визначити як сукупність вершин і ребер, де кожна вершина відповідає певному об'єкту, поняттю або сутності, а кожне ребро описує відношення між двома вершинами. Якщо напрям зв'язку має значення, використовується орієнтований граф. Якщо ж важливим є лише факт наявності зв'язку, може застосовуватися неорієнтований граф. У задачах

представлення знань частіше використовуються орієнтовані графи, оскільки відношення між сутностями зазвичай мають певну логіку та напрям, наприклад: об'єкт належить до класу, документ створений автором, ресурс пов'язаний з темою, поняття є частиною ширшої категорії [13].

Як зображено на рисунку 1.1, базовим елементом графової моделі є структура “сутність – відношення – сутність”, що дозволяє представляти знання у вигляді мережі взаємопов'язаних об'єктів. Розвитком такого підходу є графи знань – моделі, у яких вершини та зв'язки мають семантичне значення.

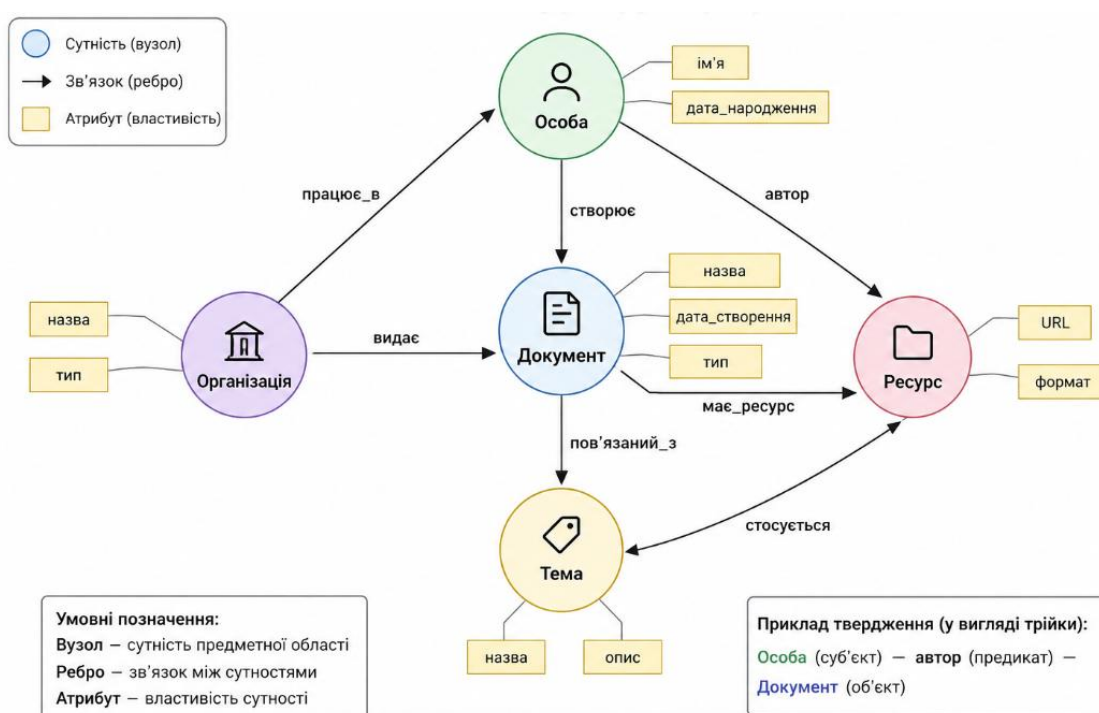


Рисунок 1.1 – Узагальнена структура графа знань

У графах знань сутності та зв'язки можуть бути типізованими: вершини представляють об'єкти предметної області, а ребра – різні типи відношень між ними. Одним із найбільш поширених підходів до реалізації графів знань є RDF-модель, у якій дані подаються у вигляді трійок “суб'єкт – предикат – об'єкт” [14]. Семантичний підхід до подання даних дає змогу пов'язувати ресурси не лише технічно, а й змістово [15]. Графи знань розглядаються як моделі, у яких поєднуються структурні зв'язки та семантична інтерпретація сутностей [16].

Іншим поширеним підходом є *property graph*, де вершини та ребра можуть містити додаткові властивості [17]. Такий підхід активно використовується в графових базах даних для роботи зі складними зв'язками між об'єктами [18]. Семантичний веб розвиває ідею пов'язаних даних і створює основу для інтеграції різномірних інформаційних ресурсів [19].

Отже, графові структури та графи знань є ефективним засобом подання, інтеграції та аналізу різномірних даних у сучасних інформаційних системах.

1.3 Гетерогенні дані та рівні гетерогенності у графових структурах

Однією з характерних ознак сучасних інформаційних систем є робота з гетерогенними даними. Під гетерогенними даними розуміють дані, які відрізняються за структурою, форматом подання, семантичним змістом, джерелами походження або способом інтерпретації [20]. Така неоднорідність є типовою для складних інформаційних середовищ, у яких дані формуються різними системами, користувачами, сервісами, документами або програмними засобами.

На відміну від однорідних наборів даних, де всі записи мають однакову структуру та описуються за спільною схемою, гетерогенні дані можуть містити різні типи об'єктів, атрибутів і зв'язків. У таких умовах виникає проблема не лише збереження даних, а й їх узгодження, інтерпретації та об'єднання в єдину модель знань [20]. Для інтеграції таких даних можуть застосовуватися підходи, що ґрунтуються на зіставленні запитів і представлень даних [21]. Узгодження схем і відповідностей між елементами різних джерел є окремою складною задачею інтеграції [22].

Гетерогенність даних проявляється на кількох рівнях. Найчастіше виділяють структурну, семантичну, джерельну та форматну гетерогенність. Структурна гетерогенність пов'язана з тим, що дані можуть мати різну організацію: табличну, ієрархічну, документну або графову. Семантична гетерогенність виникає тоді, коли однакові або подібні за змістом поняття

описуються різними назвами, атрибутами чи правилами інтерпретації. Джерельна гетерогенність зумовлена тим, що інформація надходить із різних джерел, які можуть відрізнятися рівнем повноти, достовірності та актуальності. Форматна гетерогенність проявляється у використанні різних способів запису одних і тих самих або близьких за змістом даних [20].

Основні прояви гетерогенності даних подано в таблиці 1.2.

Таблиця 1.2 – Прояви гетерогенності даних

Вид гетерогенності	Характеристика	Приклад
Структурна	Дані мають різну внутрішню організацію та можуть подаватися за різними моделями	Таблиця, JSON-документ, текстовий опис, графова структура
Семантична	Однакові або близькі за змістом об'єкти можуть описуватися різними термінами чи атрибутами	“Користувач”, “клієнт”, “замовник”
Джерельна	Дані надходять із різних джерел, які мають різний рівень повноти, деталізації та достовірності	База даних, API, вебресурс, файл
Форматна	Однакові або подібні значення подаються у різних формах запису	id, identifier, user_id, code
Сутнісна	В одній моделі поєднуються об'єкти різної природи	Особа, документ, подія, організація, процес
Реляційна	Між об'єктами існують різні типи зв'язків	“створює”, “належить до”, “перевіряє”, “пов'язаний з”

Особливе значення в межах задач моделювання знань має поняття гетерогенних сутностей. Це об'єкти різних типів, які можуть співіснувати в межах однієї інформаційної моделі та бути пов'язаними між собою різними відношеннями.

Окремою проблемою є семантичне узгодження даних. У різних джерелах одна й та сама сутність може мати різні назви, ідентифікатори або атрибути, а один і той самий тип зв'язку може бути описаний різними словами чи конструкціями.

Для подолання цих проблем доцільно використовувати графові структури. Графова модель дозволяє явно подати різні типи сутностей у вигляді вершин, а різні типи відношень – у вигляді ребер між ними, не зводячи всі об'єкти до однакової табличної структури.

Гетерогенність також впливає на подальший аналіз даних. У таких моделях важливо враховувати не лише кількість об'єктів або їхні атрибути, а й типи зв'язків, напрям відношень, проміжні сутності та можливі непрямі залежності. Наприклад, дві сутності можуть не мати прямого зв'язку, але бути пов'язаними через одну або кілька проміжних вершин. Саме такі зв'язки часто є важливими для виявлення прихованих або неочевидних залежностей у складних інформаційних середовищах.

Отже, гетерогенні дані є типовим явищем для сучасних інформаційних систем і створюють проблеми структурної, семантичної, джерельної та форматної неоднорідності. Для ефективного моделювання таких даних доцільно використовувати графові структури, які поєднують різнорідні об'єкти в єдину модель знань.

1.4 RDF, JSON-LD та семантичне подання даних

Одним із основних підходів до подання знань у графових структурах є використання стандартів семантичного вебу, зокрема RDF та JSON-LD. Їх застосування дозволяє описувати дані не як ізольовані записи, а як систему

взаємопов'язаних сутностей, у якій кожен зв'язок має визначене смислове значення. Це є важливим для інтеграції інформації з різних джерел у межах єдиної моделі знань.

RDF є базовою моделлю семантичного подання даних. Її основна ідея полягає у представленні знань у вигляді трійок “суб’єкт – предикат – об’єкт”. Суб’єкт позначає сутність, про яку йдеться, предикат визначає властивість або тип зв’язку, а об’єкт указує на іншу сутність або конкретне значення. Сукупність таких трійок формує RDF-граф, у якому сутності можуть розглядатися як вершини, а предикати – як ребра між ними [23].

Наприклад, твердження про зв’язок між двома сутностями може бути подане так: “ex:entity_1 – ex:relatedTo – ex:entity_2”

У цьому прикладі ex:entity_1 є суб’єктом, ex:relatedTo – предикатом, а ex:entity_2 – об’єктом. Такий підхід забезпечує явне подання зв’язків між об’єктами та робить дані придатними для автоматизованої обробки, аналізу й пошуку взаємозв’язків.

JSON-LD є форматом серіалізації Linked Data, який дозволяє подавати семантично описані дані у звичній JSON-структурі. Її перевага полягає в тому, що він поєднує простоту JSON із можливістю формального опису сутностей, їх типів і зв’язків [24]. Завдяки цьому JSON-LD зручно використовувати як вхідний формат для систем, що працюють із графовими моделями знань.

Ключову роль у JSON-LD відіграє елемент @context, який визначає відповідність між локальними назвами властивостей і семантичними ідентифікаторами [23]. Наприклад, поле writes може бути пов’язане з формальним відношенням ex:writes, а поле name – з властивістю ex:name. Це дозволяє інтерпретувати звичайні поля JSON не просто як текстові ключі, а як елементи семантичної моделі [24]. Під час публікації даних у вебсередовищі важливо забезпечувати зрозумілість, доступність і повторне використання семантично описаних ресурсів [25]. RDFS використовується для опису базових класів і властивостей RDF-ресурсів [26]. OWL розширює можливості

семантичного опису даних завдяки формалізації класів, властивостей і логічних обмежень [27]. SPARQL забезпечує виконання запитів до RDF-графів і дає змогу отримувати потрібні фрагменти знань із семантичної моделі [28]. Turtle є компактним текстовим форматом запису RDF-трийок, який зручно використовувати для читання та перевірки семантичних даних [29].

Таким чином, JSON-LD виступає проміжною формою між звичайним JSON-представленням даних і формальною RDF-моделлю. Він дозволяє зберегти зручність текстового формату, але водночас забезпечує можливість автоматичного перетворення описів у набір RDF-трийок. Саме цей процес є основою подальшої побудови графової моделі знань.

Як зображено на рисунку 1.2, JSON-LD опис може бути перетворений у набір RDF-трийок, які надалі формують граф знань.

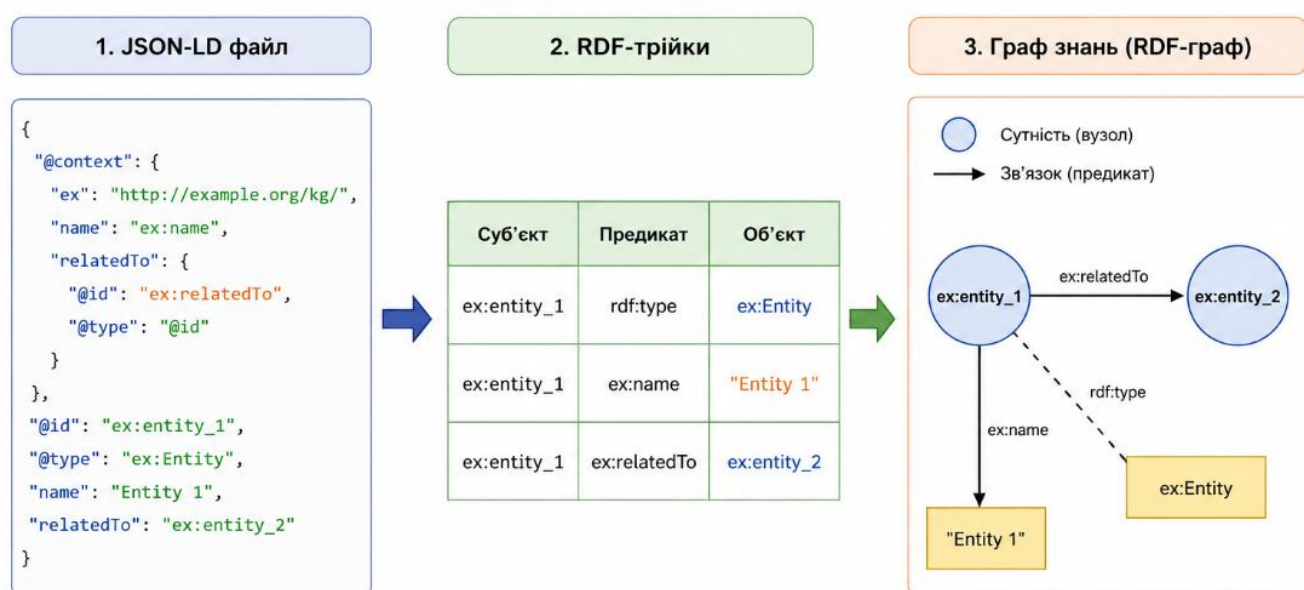


Рисунок 1.2 – Перетворення JSON-LD опису у RDF-трийки

Після перетворення JSON-LD у RDF сутності стають вершинами графа, а предикати – ребрами між ними. Якщо об'єктом трійки є інша сутність, між вершинами формується зв'язок; якщо об'єкт містить текстове або числове значення, воно розглядається як атрибут відповідної вершини. Саме така логіка

використовується у практичній частині роботи під час інтеграції кількох JSON-LD файлів у єдину графову модель знань.

Отже, RDF забезпечує формальне графове подання знань у вигляді трійок “суб’єкт – предикат – об’єкт”, а JSON-LD надає зручний спосіб опису таких даних у JSON-форматі. Їх поєднання дозволяє інтегрувати різноманітні дані, формувати графи знань та виконувати аналіз зв’язків між гетерогенними сутностями.

1.5 Методи інтеграції знань у графових моделях

Інтеграція знань у графових моделях є важливим етапом побудови інформаційних систем, що працюють із різноманітними даними. Її основна мета полягає в об’єднанні окремих фрагментів інформації з різних джерел у єдину графову структуру [30].

Одним із поширених методів є інтеграція на рівні схеми, коли перед об’єднанням даних визначаються типи сутностей, властивостей і зв’язків. Важливу роль також відіграє інтеграція на рівні сутностей, яка передбачає встановлення відповідностей між об’єктами з різних джерел.

У RDF-орієнтованих моделях інтеграція знань часто здійснюється шляхом об’єднання RDF-трійок виду “суб’єкт – предикат – об’єкт”. Якщо різні джерела використовують однакові ідентифікатори або узгоджені простори імен, їхні дані можуть бути автоматично об’єднані в один RDF-граф. Це створює основу для формування єдиної графової моделі знань [31].

Для інтеграції знань важливим є питання узгодженості ідентифікаторів. Якщо різні джерела використовують однакові URI для позначення тих самих сутностей, граф формується природним шляхом [30]. В інших випадках можуть застосовуватися словники термінів, онтології або алгоритми порівняння атрибутів [31]. Використання принципів Linked Data допомагає поєднувати розподілені описи сутностей у спільний простір знань [32]. Уточнення графів

знань також може передбачати доповнення, перевірку та покращення вже наявних зв'язків між сутностями [33].

Після інтеграції даних у графову модель з'являється можливість аналізувати не лише окремі сутності, а й структуру їхніх взаємозв'язків. Це дозволяє виявляти прямі та непрямі залежності, визначати центральні вузли та досліджувати зв'язки між об'єктами. Таким чином, інтеграція знань у графових моделях є основою для подальшого аналізу та виявлення зв'язків між гетерогенними сутностями.

1.6 Методи виявлення зв'язків між гетерогенними сутностями у графових моделях

Виявлення зв'язків між гетерогенними сутностями є одним із ключових завдань аналізу графових структур знань. Якщо інтеграція даних забезпечує об'єднання різнорідних об'єктів у спільну модель, то виявлення зв'язків дозволяє встановити, яким чином ці об'єкти пов'язані між собою [34]. У графових базах даних та мережевому аналізі такі задачі пов'язуються з пошуком шляхів, аналізом сусідства й оцінюванням важливості вузлів.

Найпростішим способом виявлення зв'язків є аналіз безпосередніх ребер графа. Якщо між двома вершинами існує пряме ребро, зв'язок можна вважати явно зафіксованим у моделі [35].

Однак у реальних інформаційних структурах важливі залежності часто не задані безпосередньо, а виникають через інші сутності. Тому для повнішого аналізу необхідно враховувати також непрямі зв'язки.

Одним із базових методів аналізу є пошук шляхів у графі. Якщо між двома сутностями існує послідовність зв'язків через проміжні вершини, це може свідчити про наявність структурного або змістового зв'язку між ними. Наприклад, дві сутності можуть бути пов'язані через спільний документ, тему або джерело інформації. [34]

Як зображено на рисунку 1.3, прямий зв'язок передбачає наявність одного ребра між сутностями, тоді як непрямий формується через ланцюг проміжних вершин. Аналіз шляхів дозволяє виявляти залежності, які залишаються прихованими у табличних або файлових структурах даних.

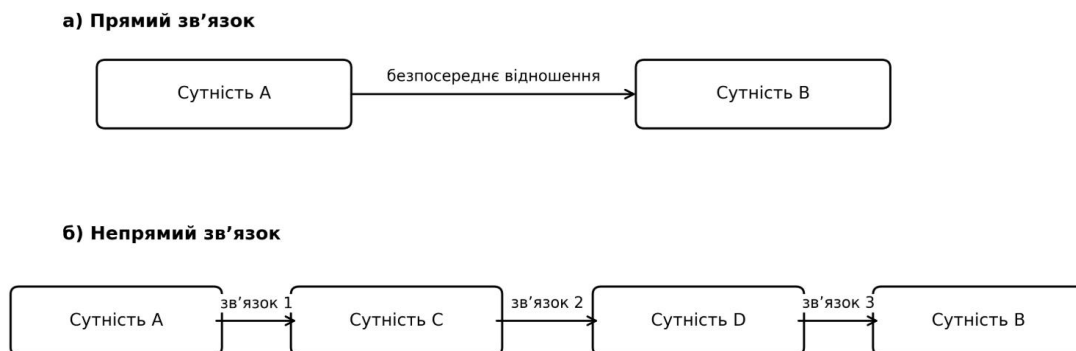


Рисунок 1.3 – Приклад прямого та непрямого зв'язку між сутностями

Для практичного аналізу часто використовується метод пошуку найкоротшого шляху між вершинами. Він показує мінімальну кількість переходів, необхідних для встановлення зв'язку між сутностями. У гетерогенних графах важливо враховувати не лише довжину шляху, а й типи ребер, через які він проходить.

Ще одним методом є аналіз сусідства вершини, який дозволяє визначити локальний контекст сутності – об'єкти, з якими вона безпосередньо взаємодіє [35]. Важливим показником також є центральність вершини, що характеризує значущість окремих вузлів у структурі графа [36]. Реляційне машинне навчання для графів знань розглядає зв'язки між сутностями як основу для виявлення прихованих закономірностей у даних [37].

Отже, виявлення зв'язків між гетерогенними сутностями у графових структурах здійснюється за допомогою аналізу прямих і непрямих зв'язків, пошуку шляхів, дослідження сусідства, оцінювання центральності вузлів та врахування типів відношень. Поєднання цих методів дозволяє виявляти

приховані залежності та формувати цілісну модель взаємозв'язків між різнорідними об'єктами.

1.7 Постановка проблеми та обґрунтування напрямку дослідження

Проведений аналіз показує, що сучасні інформаційні системи дедалі частіше працюють із множиною взаємопов'язаних об'єктів, які можуть мати різну структуру, семантику та джерела походження. У таких умовах традиційні способи організації даних не завжди забезпечують достатню гнучкість для подання складних зв'язків між сутностями.

Основна проблема полягає в тому, що гетерогенні дані часто існують у вигляді окремих фрагментів, які важко аналізувати як єдину систему. Різні джерела можуть описувати пов'язані між собою сутності, однак без спільної моделі ці зв'язки залишаються неочевидними.

Отже, актуальною є розробка підходу, який забезпечить інтеграцію JSON-LD описів у єдину графову модель знань і дозволить виконувати аналіз зв'язків між гетерогенними сутностями. Запропонований підхід передбачає формування графової структури, у якій різнотипні об'єкти та відношення між ними можуть бути представлені в узгодженому вигляді.

1.8 Висновки до першого розділу

У першому розділі було розглянуто теоретичні основи моделювання та інтеграції знань у графових структурах. Проаналізовано основні підходи до представлення знань у сучасних інформаційних системах та встановлено, що графові моделі є доцільними для подання складних предметних областей.

Визначено, що графи знань дозволяють представляти інформацію як систему взаємопов'язаних об'єктів, де вершини відповідають сутностям, а ребра – відношенням між ними. Окрему увагу приділено проблемі гетерогенності

даних, яка проявляється у відмінностях типів сутностей, структури, форматів і джерел інформації.

Також було розглянуто роль RDF та JSON-LD у семантичному поданні даних. RDF забезпечує формалізоване представлення знань у вигляді трійок, а JSON-LD дозволяє описувати такі дані у JSON-орієнтованому форматі та інтегрувати їх у єдину графову модель.

Отже, проведений аналіз підтвердив доцільність використання графових структур для інтеграції та аналізу різнорідних даних. Отримані результати створюють основу для подальшої розробки архітектури програмної системи та реалізації засобів виявлення зв'язків між гетерогенними сутностями.

РОЗДІЛ 2. МЕТОДИ ТА ЗАСОБИ МОДЕЛЮВАННЯ Й ІНТЕГРАЦІЇ ЗНАНЬ У ГРАФОВІЙ СТРУКТУРІ

2.1 Опис предметної області графової структури

Предметною областю дослідження є процес підготовки, оформлення, перевірки та захисту кваліфікаційної роботи здобувача освіти. У межах цієї предметної області взаємодіють різні учасники, документи, етапи перевірки та організаційні структури, які разом формують складну систему взаємопов'язаних сутностей. Такий процес є зручним прикладом для моделювання у вигляді графової структури, оскільки він містить не лише окремі об'єкти, а й велику кількість змістових зв'язків між ними [38]. Для подальшого аналізу таких зв'язків доцільно використовувати інструменти побудови та обробки графів [39].

Особливе місце у предметній області займає пояснювальна записка до кваліфікаційної роботи. Це складена сутність, що містить кілька частин: бланк завдання, перший розділ, другий розділ, третій розділ і четвертий розділ. Такий підхід дає змогу точніше відобразити внутрішню структуру документа та зв'язки окремих частин із різними учасниками процесу. Наприклад, четвертий розділ може перевірятися консультантами з охорони праці та безпеки життєдіяльності, тоді як уся пояснювальна записка перевіряється керівником, нормоконтролером, рецензентом та екзаменаційною комісією.

Як зображено на рисунку 2.1, процес виконання та захисту кваліфікаційної роботи можна подати як систему взаємопов'язаних об'єктів, у якій здобувач створює основні документи, керівник перевіряє та підписує роботу, рецензент формує рецензію, відповідальний за антиплагіат перевіряє скорочену версію пояснювальної записки, а екзаменаційна комісія розглядає підготовлений пакет матеріалів під час захисту.

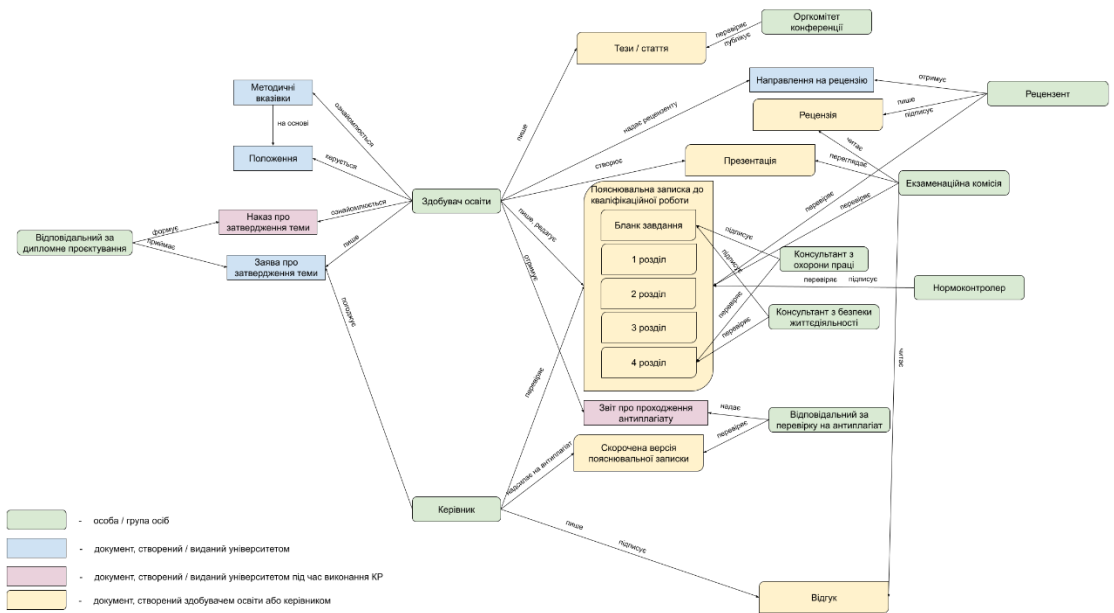


Рисунок 2.1 – Предметна область виконання та захисту кваліфікаційної роботи

Предметній області властива гетерогенність. В одному інформаційному просторі поєднані люди, групи, нормативні документи, документи здобувача, складені документи, частини документів і документи перевірки. Крім того, ці сутності мають різну структуру. Наприклад, пояснювальна записка містить окремі розділи, тоді як рецензія або відгук керівника є окремими документами без внутрішньої деталізації в межах даної моделі. Це створює підстави для використання графового підходу, який дозволяє гнучко описувати різні типи сутностей і зв'язків між ними.

У межах дослідження предметна область подається через набір JSON-LD файлів, кожен із яких описує окремий фрагмент загальної системи. Один файл може містити відомості про учасників процесу, інший – про нормативні документи, третій – про структуру пояснювальної записки, четвертий – про процес перевірки, рецензування та захисту. Після цього всі описи інтегруються в єдину графову модель знань. Такий підхід дозволяє продемонструвати не лише моделювання окремих сутностей, а й інтеграцію знань із кількох джерел.

2.2 Визначення типів гетерогенних сутностей у графовій структурі

У межах дослідження предметна область кваліфікаційної роботи розглядається як сукупність взаємопов'язаних об'єктів різної природи. До таких об'єктів належать учасники процесу виконання роботи, документи, частини документів, групові суб'єкти, етапи перевірки та матеріали, що супроводжують підготовку й захист кваліфікаційної роботи. Саме різнотипність цих об'єктів дає підстави розглядати їх як гетерогенні сутності.

Основні типи сутностей, використані у графовій моделі, наведено в таблиці 2.1.

Таблиця 2.1 – Основні типи сутностей у графовій моделі

Тип сутності	Приклад	Призначення
Person	здобувач освіти, керівник, рецензент, нормоконтролер	Представлення індивідуальних учасників процесу виконання, перевірки та захисту кваліфікаційної роботи
UniversityDocument	методичні вказівки, положення, наказ про затвердження теми	Представлення нормативних та організаційних документів університету
CompositeDocument	пояснювальна записка до кваліфікаційної роботи	Представлення складеного документа, який містить окремі структурні частини
DocumentPart	бланк завдання, 1 розділ, 2 розділ, 3 розділ, 4 розділ	Представлення частин пояснювальної записки

Як видно з таблиці 2.1, у графовій моделі поєднуються сутності різних категорій: люди, групи, документи, частини документів і документи оцінювання.

Такий підхід дозволяє не обмежуватися лінійним описом процесу підготовки кваліфікаційної роботи, а подати його як мережу взаємопов'язаних об'єктів.

Окрему увагу в моделі приділено сутності “CompositeDocument”, яка використовується для опису пояснювальної записки до кваліфікаційної роботи. Така сутність відрізняється від звичайного документа тим, що має внутрішню структуру і складається з кількох частин. Наприклад, пояснювальна записка включає бланк завдання, перший, другий, третій і четвертий розділи. Це дозволяє моделювати не лише сам факт існування документа, а й його внутрішню будову.

2.3 Визначення типів зв'язків між сутностями у графовій структурі

У графовій моделі знань важливу роль відіграють не лише сутності, а й зв'язки між ними. Якщо сутності відображають окремі об'єкти предметної області, то зв'язки показують, яким чином ці об'єкти взаємодіють між собою. Саме завдяки зв'язкам набір окремих даних перетворюється на цілісну структуру знань, придатну для подальшого аналізу.

У межах цієї роботи зв'язок розглядається як спрямоване відношення між двома сутностями. Наприклад, здобувач освіти може бути пов'язаний із пояснювальною запискою через відношення “writes”, керівник – через відношення “checks”, а пояснювальна записка зі своїми розділами – через відношення “hasPart”. Таке подання відповідає логіці RDF-моделі, у якій кожен факт описується у вигляді трійки “суб'єкт – предикат – об'єкт”. У результаті кожне відношення не лише фіксує факт взаємодії між об'єктами, а й уточнює його зміст, що є важливим для подальшого пошуку прямих і непрямих зв'язків у графі. Це також дає змогу не просто зберігати інформацію про об'єкти, а простежувати логіку їх взаємодії в межах усїєї предметної області.

Як видно з таблиці 2.2, у моделі використовуються різні за змістом типи зв'язків. Авторські зв'язки описують створення документів здобувачем освіти. Наприклад, здобувач пише пояснювальну записку, готує презентацію або

створює скорочену версію роботи. Такі зв'язки дозволяють показати, які саме матеріали були створені учасником процесу.

Таблиця 2.2 – Основні групи зв'язків у графовій моделі

Група зв'язків	Приклади предикатів	Зміст	Приклад
Авторські зв'язки	writes, creates	створення або підготовка документа	здобувач пише пояснювальну записку
Перевірочні зв'язки	checks, reviews, approves	перевірка, рецензування або затвердження матеріалів	рецензент рецензує роботу
Структурні зв'язки	hasPart, partOf	відношення між складеним документом і його частинами	пояснювальна записка має розділи
Комунікаційні зв'язки	sends, receives, passesTo, submittedTo	передача або подання документів між учасниками	робота подається до комісії
Нормативні зв'язки	basedOn, regulatedBy	зв'язок із нормативними або методичними документами	методичні вказівки базуються на положенні
Організаційні зв'язки	formedBy, acceptedBy, readBy, signs	оформлення, прийняття, ознайомлення або підписання документів	керівник підписує пояснювальну записку

Структурні зв'язки використовуються для опису складених сутностей. У межах цієї роботи особливе значення має пояснювальна записка до кваліфікаційної роботи, оскільки вона розглядається не як один простий

документ, а як складена сутність. Вона містить бланк завдання, перший, другий, третій і четвертий розділи. Таке подання дозволяє окремо моделювати не лише саму пояснювальну записку, а й її частини.

У межах розробленої системи всі зв'язки подаються як предикати JSON-LD. Після завантаження файлів вони перетворюються у RDF-трійки, а потім використовуються для побудови графової моделі. У графі сутності подаються як вузли, а зв'язки між ними – як ребра. Наприклад, якщо вхідний файл містить опис того, що керівник перевіряє пояснювальну записку, то в графі буде створено ребро між сутністю керівника та сутністю пояснювальної записки.

Таким чином, визначені типи зв'язків дозволяють подати процес виконання та захисту кваліфікаційної роботи як цілісну графову структуру. Вони охоплюють створення документів, їх перевірку, рецензування, підписання, передавання, подання до захисту, а також внутрішню структуру складених документів. Надалі ці зв'язки використовуються для побудови інтегрованого графа знань і виявлення прямих та непрямих залежностей між гетерогенними сутностями.

2.4 Модель вхідних JSON-LD даних у графовій структурі

Для побудови графової моделі знань у роботі використано набір вхідних даних у форматі JSON-LD. Такий формат було обрано тому, що він дозволяє описувати не лише окремі об'єкти, а й семантичні зв'язки між ними. На відміну від звичайного JSON, JSON-LD містить контекст “@context”, за допомогою якого назви властивостей пов'язуються з відповідними ідентифікаторами, а також дає змогу явно задавати типи сутностей і відношення між ними.

Вхідні дані не подано одним суцільним файлом, а розділено на кілька окремих JSON-LD файлів. Це зроблено для того, щоб показати процес інтеграції знань із різних джерел. Кожен файл описує окремий фрагмент предметної області: учасників процесу, університетські документи, структуру кваліфікаційної роботи та процес її перевірки й захисту. Після завантаження ці

файли об'єднуються в один RDF-граф, на основі якого формується спільна графова модель знань.

Структуру вхідних JSON-LD файлів наведено в таблиці 2.3.

Таблиця 2.3 – Структура вхідних JSON-LD файлів

Файл	Що описує	Приклади сутностей
actors.jsonld	Учасників процесу виконання та захисту кваліфікаційної роботи	здобувач освіти, керівник, рецензент, нормоконтролер, екзаменаційна комісія
university_documents.jsonld	Нормативні та організаційні документи університету	методичні вказівки, положення, наказ про затвердження теми, заява про затвердження теми
thesis_package.jsonld	Документи, які створюються здобувачем у межах кваліфікаційної роботи	пояснювальна записка, презентація, тези / стаття, скорочена версія пояснювальної записки
review_defense_process.jsonld	Процеси перевірки, рецензування, проходження антиплагіату та захисту	рецензія, відгук керівника, звіт антиплагіату, направлення на рецензію

Як видно з таблиці 2.3, кожен файл відповідає за окрему частину предметної області. Наприклад, файл “actors.jsonld” містить описи осіб і груп, які беруть участь у процесі підготовки та захисту кваліфікаційної роботи. До таких

сутностей належать здобувач освіти, керівник, рецензент, консультанти, нормоконтролер, відповідальний за перевірку на антиплагіат та екзаменаційна комісія.

Кожен JSON-LD файл має спільний простір ідентифікаторів “ex”, що дозволяє об’єднувати сутності з різних файлів в одну графову модель. Наприклад, сутність “ex:qualification_work_note” може згадуватися в кількох файлах: в одному – як документ, який пише здобувач, в іншому – як документ, який перевіряє керівник або рецензент, а також як матеріал, що подається до екзаменаційної комісії. Завдяки однаковому ідентифікатору ці описи не створюють різні дублікати, а доповнюють одну спільну сутність у графі.

Таким чином, вхідні JSON-LD дані виконують роль формалізованого опису предметної області. Вони дозволяють представити різні типи сутностей, різні види зв’язків і різні джерела інформації у вигляді єдиної семантичної структури. Це створює основу для подальшого перетворення даних у RDF-граф, побудови графової моделі та виявлення зв’язків між гетерогенними сутностями.

2.5 Моделювання пояснювальної записки як складеної сутності

У межах розробленої графової моделі окрему увагу приділено представленню пояснювальної записки до кваліфікаційної роботи, оскільки вона не є простою однорідною сутністю. На відміну від таких документів, як рецензія, відгук керівника або звіт про проходження перевірки на академічний плагіат, пояснювальна записка має внутрішню структуру та складається з кількох логічно пов’язаних частин. Тому в графовій моделі вона подається як складена сутність типу “CompositeDocument” [39]. Для реалізації інтерфейсу роботи з такою моделлю може використовуватися веборієнтований інструмент Streamlit [40].

Як зображено на рисунку 2.2, пояснювальна записка моделюється як документ, що містить кілька частин: бланк завдання, перший розділ, другий розділ, третій розділ і четвертий розділ. Такий підхід дозволяє подати документ

не як один неподільний об'єкт, а як структуровану сутність, елементи якої можуть мати власні зв'язки з іншими об'єктами предметної області.

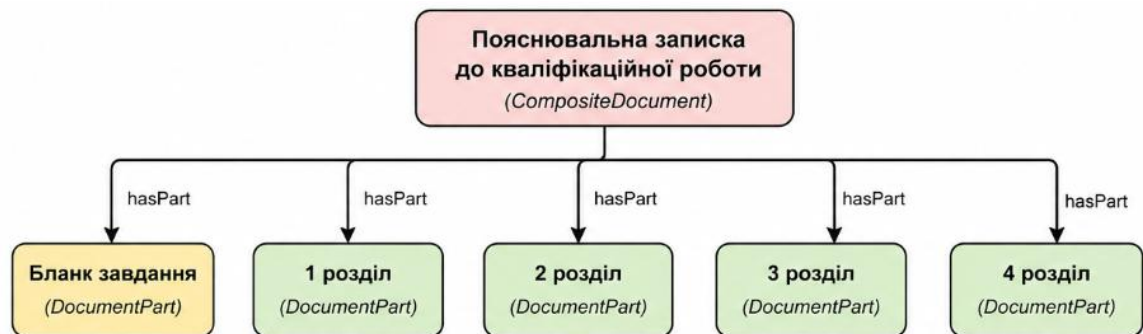


Рисунок 2.2 – Структура пояснювальної записки як складеної сутності

Особливо важливим є те, що окремі частини пояснювальної записки можуть брати участь у власних зв'язках. Наприклад, четвертий розділ пов'язаний з консультантами з охорони праці та безпеки життєдіяльності, які його перевіряють і підписують. Тобто зв'язки в моделі можуть встановлюватися не лише на рівні всього документа, а й на рівні його окремих частин. Це робить модель більш точною та наближеною до реального процесу підготовки кваліфікаційної роботи.

Такий підхід демонструє структурну гетерогенність даних у межах предметної області. Одні документи моделюються як прості сутності, наприклад рецензія або відгук керівника, тоді як пояснювальна записка має складену структуру. Відповідно, графова модель дозволяє одночасно представляти як прості, так і складені об'єкти, не порушуючи цілісності загальної структури знань.

Отже, представлення пояснювальної записки як складеної сутності дозволяє точніше моделювати реальну структуру кваліфікаційної роботи та її взаємозв'язки з іншими об'єктами. Це забезпечує можливість аналізу не лише загальних зв'язків між документами, учасниками й процесами, а й детальніших залежностей на рівні окремих частин документа. Саме така деталізація є

важливою для побудови графової моделі знань, орієнтованої на виявлення зв'язків між гетерогенними сутностями.

2.6 Архітектура програмної системи інтеграції знань

Архітектура програмної системи розроблена з урахуванням основної мети дослідження – забезпечити інтеграцію кількох JSON-LD описів у єдину графову модель знань та створити можливість подальшого аналізу зв'язків між гетерогенними сутностями. Система не обмежується лише зчитуванням вхідних даних, а реалізує повний цикл їх опрацювання: від завантаження JSON-LD файлів до побудови RDF-графа, формування графової структури, візуалізації та пошуку зв'язків між сутностями [41]. Табличне подання результатів опрацювання даних реалізується засобами Pandas [42].

Загальна архітектура програмної системи зображена на рисунку 2.3.

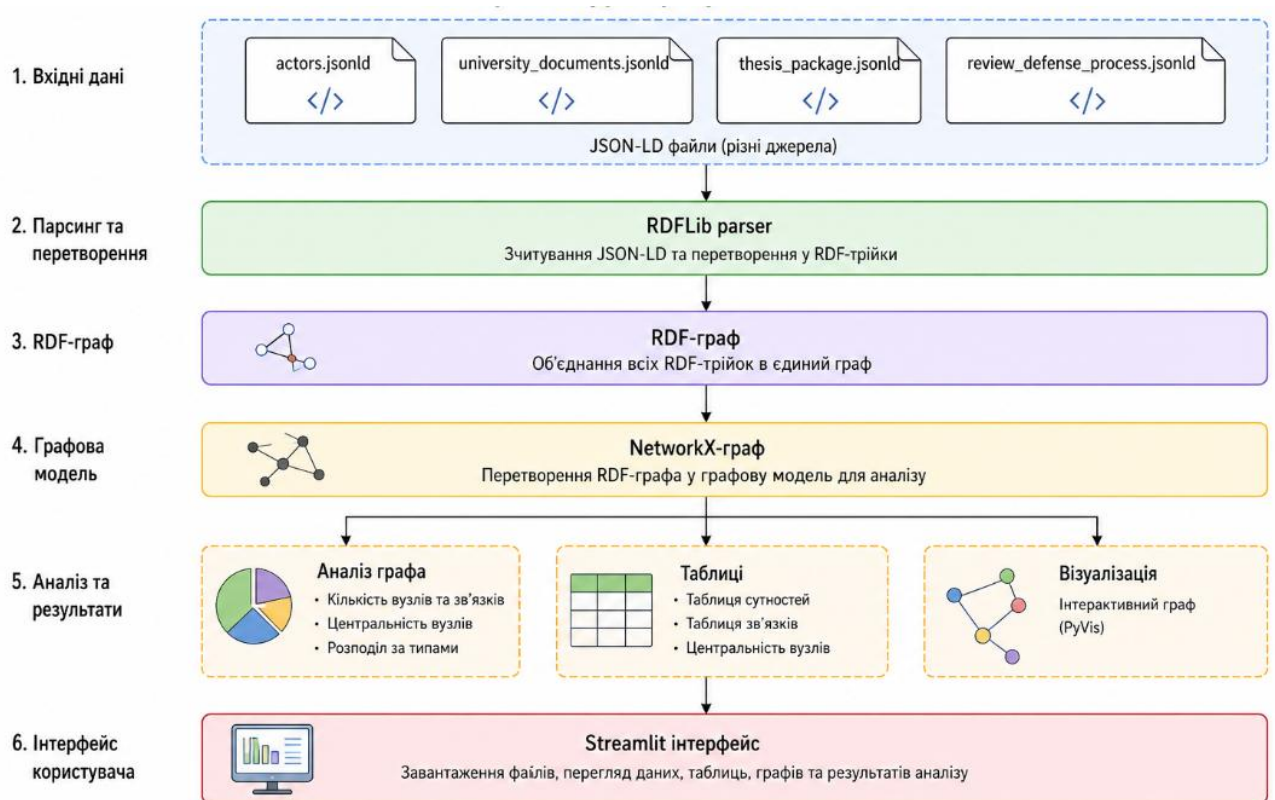


Рисунок 2.3 – Архітектура програмної системи

Вхідними даними системи є кілька JSON-LD файлів, кожен із яких описує окремий фрагмент предметної області. Один файл може містити інформацію про учасників процесу виконання кваліфікаційної роботи, інший – про університетські документи, третій – про структуру пояснювальної записки, а четвертий – про процес перевірки, рецензування та захисту. Такий підхід демонструє джерельну гетерогенність даних, оскільки інформація надходить із кількох логічно відокремлених файлів.

На першому етапі система завантажує JSON-LD файли через вебінтерфейс, реалізований за допомогою Streamlit. Користувач може обрати один або кілька файлів, після чого система зчитує їх вміст і передає на подальшу обробку. Streamlit у цій архітектурі виконує роль інтерфейсного шару, що забезпечує взаємодію користувача з програмною системою.

Другим етапом є перетворення JSON-LD описів у RDF-граф за допомогою бібліотеки RDFLib. Вона забезпечує зчитування семантичних даних і формування RDF-трійок у вигляді “суб’єкт – предикат – об’єкт”. Кожен JSON-LD файл перетворюється у набір RDF-трійок, після чого ці трійки об’єднуються в один спільний RDF-граф. Саме на цьому етапі відбувається інтеграція знань, оскільки окремі фрагменти предметної області поєднуються в єдину семантичну структуру.

Третім етапом є перетворення RDF-графа у графову модель NetworkX. Це необхідно для подальшого аналізу структури графа: визначення кількості вузлів і ребер, пошуку шляхів між сутностями та розрахунку центральності. У створеній моделі вершини відповідають сутностям предметної області, наприклад здобувачу освіти, керівнику, пояснювальній записці, рецензії або екзаменаційній комісії. Ребра відповідають зв’язкам між цими сутностями, зокрема “writes”, “checks”, “signs”, “hasPart”, “reviews”, “approves”.

Після побудови графової моделі система формує таблиці сутностей і зв’язків. Таблиця сутностей містить ідентифікатор вузла, його назву, тип, загальну кількість зв’язків, кількість вхідних і вихідних зв’язків. Таблиця зв’язків відображає початкову сутність, тип відношення та цільову сутність. Ці

таблиці використовуються для аналізу результатів інтеграції та перевірки коректності об'єднання даних.

Важливим компонентом системи є модуль аналізу зв'язків, який реалізує пошук найкоротшого шляху між двома вибраними сутностями. Завдяки цьому система дозволяє знаходити не лише прямі, а й непрямі зв'язки. Наприклад, здобувач освіти може бути пов'язаний з екзаменаційною комісією не безпосередньо, а через пояснювальну записку, яка подається до захисту. Такий підхід відповідає основній ідеї роботи – виявленню змістових залежностей між гетерогенними сутностями.

Проектована система поділяється на кілька логічних компонентів. Інтерфейсний компонент відповідає за завантаження файлів і відображення результатів. Компонент побудови графа забезпечує перетворення JSON-LD описів у RDF-граф і подальше формування графової моделі. Компонент табличного подання формує таблиці сутностей, зв'язків і показників центральності. Компонент аналізу відповідає за пошук шляхів між сутностями, а компонент візуалізації – за графічне представлення побудованої моделі знань.

Отже, запропонована архітектура забезпечує виконання основних етапів дослідження: завантаження гетерогенних JSON-LD джерел, їх інтеграцію в RDF-граф, побудову графової моделі, візуалізацію структури знань і виявлення зв'язків між сутностями. Така архітектура є гнучкою для подальшого розширення та відповідає завданню моделювання й інтеграції знань у графових структурах.

2.7 Алгоритм інтеграції JSON-LD описів у графову структуру

Інтеграція JSON-LD описів у графову модель є одним із ключових етапів розробленої системи, оскільки саме на цьому етапі окремі фрагменти даних, подані у різних файлах, об'єднуються в єдину структуру знань. У межах роботи вхідні дані подаються у вигляді кількох JSON-LD файлів, кожен з яких описує окрему частину предметної області: учасників процесу виконання

кваліфікаційної роботи, університетські документи, структуру пояснювальної записки, а також процеси перевірки, рецензування та захисту. Для візуального подання результатів інтеграції може застосовуватися графова візуалізація [43].

Загальний алгоритм інтеграції JSON-LD описів у графову модель подано на рисунку 2.4.

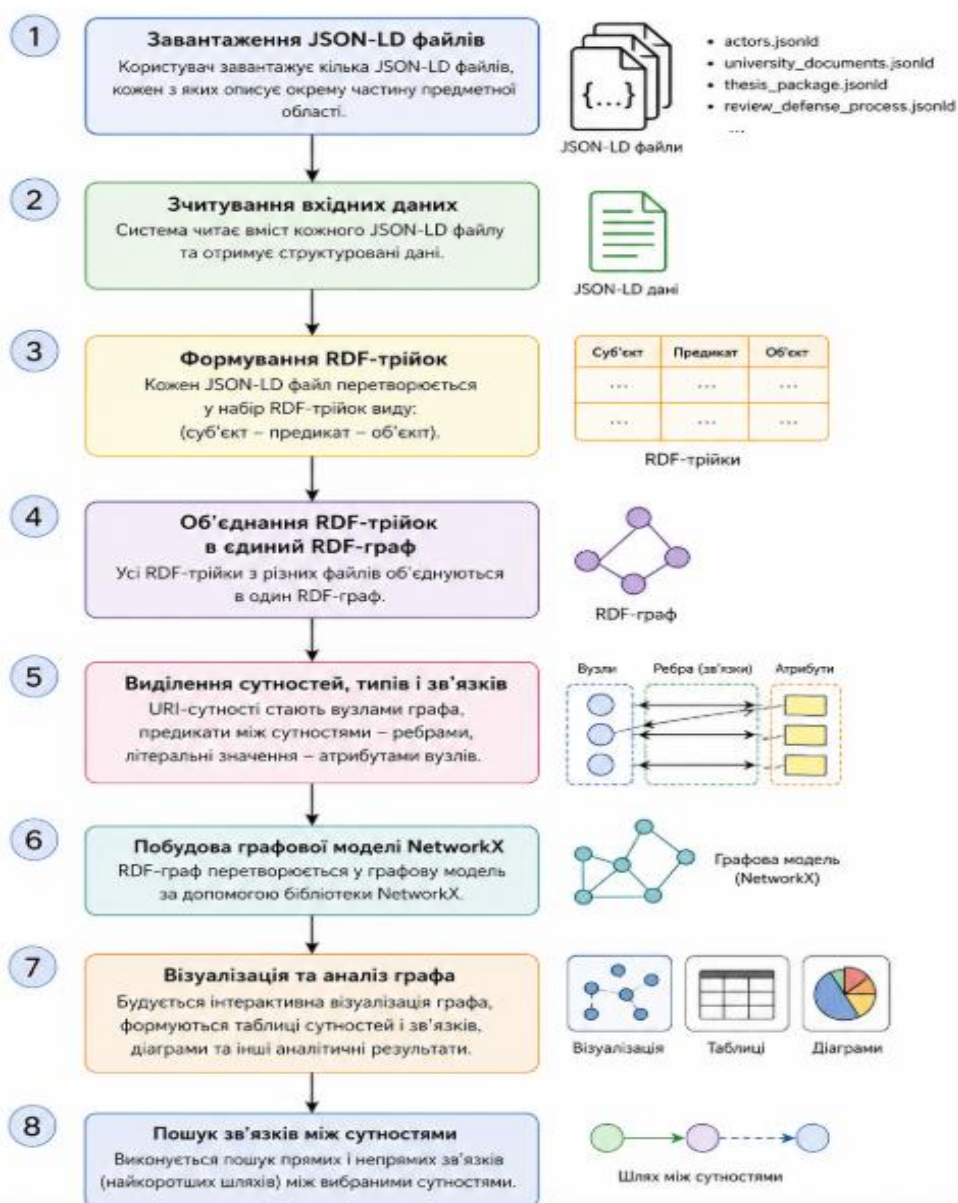


Рисунок 2.4 – Алгоритм інтеграції JSON-LD описів у граф знань

Як зображено на рисунку 2.4, першим етапом є завантаження JSON-LD файлів. У розробленій системі передбачено можливість завантаження не одного, а кількох файлів, що дозволяє моделювати джерельну гетерогенність даних.

Наприклад, один файл може містити опис учасників процесу, інший – опис документів, третій – структуру пояснювальної записки, а четвертий – зв’язки, пов’язані з рецензуванням і захистом. Такий підхід дає змогу не зберігати всю інформацію в одному монолітному файлі, а представляти предметну область як сукупність окремих взаємопов’язаних фрагментів.

Третій етап полягає у перетворенні JSON-LD описів у RDF-трійки. Кожна трійка має структуру:

На четвертому етапі RDF-трійки, отримані з різних JSON-LD файлів, об’єднуються в один RDF-граф. Саме цей етап є етапом інтеграції, оскільки окремі джерела перестають існувати ізольовано й формують спільний простір знань. Якщо в різних файлах використовується однаковий ідентифікатор сутності, наприклад “ex:qualification_work_note”, система розглядає ці описи як інформацію про один і той самий об’єкт. Таким чином, один файл може описувати сам документ, інший – його частини, а третій – процес його перевірки або рецензування.

П’ятий етап передбачає виділення сутностей, їх типів і зв’язків. Сутності, які мають URI-ідентифікатори, перетворюються на вузли графа. Наприклад, такими вузлами є “ex:student”, “ex:supervisor”, “ex:qualification_work_note”, “ex:reviewer”, “ex:defense_committee”. Предикати, що поєднують одну URI-сутність з іншою, перетворюються на ребра графа. Наприклад, “writes”, “checks”, “signs”, “reviews”, “hasPart”, “submittedTo” стають типами зв’язків між вузлами.

На шостому етапі RDF-граф перетворюється у графову модель NetworkX. У цій моделі вузли відповідають сутностям предметної області, а ребра – семантичним відношенням між ними. Використання NetworkX дає змогу не лише зберігати структуру графа, а й виконувати подальший аналіз: визначати кількість зв’язків для кожної сутності, обчислювати центральність вузлів, а також знаходити шляхи між об’єктами.

Останній етап алгоритму пов’язаний із візуалізацією та аналізом побудованого графа. Після інтеграції даних система формує інтерактивне графове представлення, таблицю сутностей, таблицю зв’язків, а також діаграми

розподілу сутностей і зв'язків за типами. Це дає змогу наочно перевірити, чи правильно були об'єднані дані з різних JSON-LD файлів, а також виявити прямі й непрямі зв'язки між гетерогенними сутностями.

Отже, запропонований алгоритм забезпечує поетапне перетворення розрізнених JSON-LD описів у цілісну графову структуру знань. Це створює основу для подальшого виявлення зв'язків між гетерогенними сутностями, що безпосередньо відповідає меті даної кваліфікаційної роботи.

2.8 Методика виявлення зв'язків між гетерогенними сутностями у графовій структурі

Після інтеграції JSON-LD описів у єдину графову модель знань важливим етапом є виявлення зв'язків між гетерогенними сутностями. У межах цієї роботи під гетерогенними сутностями розуміються об'єкти різної природи, що належать до однієї предметної області, але мають різні типи, структуру та функціональне призначення. До таких сутностей належать здобувач освіти, керівник, рецензент, екзаменаційна комісія, пояснювальна записка, її окремі розділи, методичні вказівки, рецензія, відгук керівника, звіт про перевірку на плагіат та інші елементи процесу виконання й захисту кваліфікаційної роботи.

У межах запропонованої методики розрізняються прямі та непрямі зв'язки між сутностями. Прямий зв'язок існує тоді, коли між двома сутностями є безпосереднє ребро графа. Наприклад, зв'язок: “ex:student -> writes -> ex:qualification_work_note”

Непрямий зв'язок виникає тоді, коли дві сутності не мають прямого ребра, але можуть бути пов'язані через одну або кілька проміжних сутностей. Наприклад, здобувач освіти може не мати прямого зв'язку з екзаменаційною комісією, однак такий зв'язок можна виявити через пояснювальну записку: “ex:student -> writes -> ex:qualification_work_note -> submittedTo -> ex:defense_committee”

Для пошуку непрямих зв'язків у системі використовується пошук найкоротшого шляху між двома вершинами графа. Такий підхід дозволяє визначити мінімальну послідовність проміжних сутностей і відношень, які поєднують два об'єкти. Наприклад, між консультантом з охорони праці та пояснювальною запискою може бути знайдений шлях: “ex:occupational_safety_consultant -> checks -> ex:chapter_4 -> partOf -> ex:qualification_work_note”

Окреме значення має аналіз центральності вузлів графа. Центральність дозволяє визначити, які сутності мають найбільшу кількість зв'язків і відіграють ключову роль у структурі предметної області. У межах розробленої моделі очікувано центральними можуть бути такі сутності, як пояснювальна записка, здобувач освіти, керівник або екзаменаційна комісія. Це пояснюється тим, що саме ці об'єкти поєднують різні частини процесу виконання, перевірки, рецензування та захисту кваліфікаційної роботи.

Методика виявлення зв'язків між гетерогенними сутностями включає такі основні етапи:

- 1) формування інтегрованого RDF-графа на основі кількох JSON-LD файлів;
- 2) перетворення RDF-графа у графову структуру NetworkX;
- 3) визначення типів сутностей і типів зв'язків між ними;
- 4) побудова прямої множини ребер між сутностями;
- 5) пошук найкоротшого шляху між вибраними сутностями;
- 6) інтерпретація знайденого шляху як змістового зв'язку;
- 7) аналіз центральності вузлів для виявлення ключових об'єктів графа.

Таким чином, запропонована методика дозволяє перейти від набору окремих JSON-LD описів до цілісної графової моделі знань, у якій можна аналізувати як явні, так і опосередковані зв'язки. Це є важливим для дослідження гетерогенних предметних областей, оскільки різноманітні сутності часто пов'язані не напряму, а через проміжні документи, учасників, процеси або структурні

елементи. У результаті графова модель не лише відображає наявні дані, а й дає змогу виявляти додаткові змістові залежності між об'єктами.

2.9 Висновки до другого розділу

У другому розділі було визначено методичну основу моделювання та інтеграції знань у графових структурах для виявлення зв'язків між гетерогенними сутностями. Основну увагу було зосереджено на формалізації предметної області, визначенні типів сутностей і зв'язків, побудові структури вхідних JSON-LD даних, а також описі архітектури програмної системи та алгоритму інтеграції даних у графову модель.

Отже, у другому розділі сформовано концептуальну й методичну основу практичної реалізації системи. Визначено предметну область, типи гетерогенних сутностей, типи зв'язків, структуру вхідних JSON-LD файлів, архітектуру програмної системи, алгоритм інтеграції даних і методику виявлення зв'язків. Отримані результати створюють підґрунтя для практичної реалізації системи та аналізу результатів інтеграції знань у третьому розділі.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ СИСТЕМИ ТА АНАЛІЗ РЕЗУЛЬТАТІВ ІНТЕГРАЦІЇ ЗНАНЬ

3.1 Засоби реалізації програмної системи інтеграції знань

Для практичної реалізації системи моделювання та інтеграції знань у графових структурах було обрано засоби, які забезпечують завантаження JSON-LD файлів, формування RDF-графа, побудову графової моделі, її візуалізацію та аналіз зв'язків між сутностями.

Для роботи з JSON-LD та RDF використано бібліотеку RDFLib. Вона перетворює JSON-LD описи у RDF-граф, де факти подаються у формі трійок “суб’єкт – предикат – об’єкт”. Документація RDFLib також описує можливості парсингу RDF-даних і серіалізації графів у різних форматах [44].

Для побудови та аналізу графової структури застосовано NetworkX. Після перетворення RDF-графа вузли відповідають сутностям предметної області, а ребра – зв'язкам між ними. Офіційні матеріали NetworkX описують інструменти для аналізу мереж, пошуку шляхів і розрахунку метрик графа [45].

Для інтерактивної візуалізації графової моделі використано PyVis, а для побудови діаграм – Plotly. Табличне подання результатів реалізовано засобами Pandas.

Вебінтерфейс системи створено за допомогою Streamlit. Він дає змогу завантажувати JSON-LD файли, переглядати граф, таблиці, діаграми та результати пошуку шляхів між сутностями [46].

Перевагою обраного набору засобів є те, що кожен інструмент виконує окрему функцію в загальному процесі обробки даних. RDFLib відповідає за семантичне подання інформації, NetworkX – за графову структуру та її аналіз, а Streamlit, PyVis, Plotly і Pandas забезпечують зручне подання результатів для користувача. Завдяки такому поєднанню система охоплює повний цикл роботи з даними: від завантаження вхідних файлів до візуального й табличного аналізу сформованого графа знань.

Основні засоби реалізації програмної системи наведено в таблиці 3.1.

Таблиця 3.1 – Засоби реалізації програмної системи

Засіб реалізації	Призначення в системі
Python	Основна мова програмування, що забезпечує реалізацію логіки системи
RDFLib	Зчитування JSON-LD файлів і формування RDF-графа
NetworkX	Побудова графової структури та аналіз зв'язків між сутностями
PyVis	Інтерактивна візуалізація графової моделі знань
Streamlit	Створення вебінтерфейсу для роботи з системою
Pandas	Формування таблиць сутностей, зв'язків і центральності
Plotly	Побудова діаграм для аналізу складу графа

Загальна логіка роботи системи є такою: користувач завантажує кілька JSON-LD файлів, система об'єднує їх у спільний RDF-граф, перетворює RDF-трийки у граф NetworkX, а потім формує інтерактивну візуалізацію, таблиці сутностей і зв'язків, діаграми розподілу та результати пошуку найкоротших шляхів між сутностями.

3.2 Структура програмного проєкту інтеграції знань

Розроблена програмна система має модульну структуру, що розділяє завантаження даних, побудову RDF-графа, перетворення у NetworkX, формування таблиць, аналіз зв'язків і візуалізацію [47].

Головним файлом системи є `app.py`, який відповідає за вебінтерфейс, завантаження даних і відображення результатів. Основна логіка винесена в окремі модулі: `graph_builder.py` формує RDF- і NetworkX-графи, `tables.py` створює таблиці результатів, `analysis.py` виконує пошук шляхів, а `visualization.py` будує інтерактивну візуалізацію.

Структура програмного проєкту наведена на рисунку 3.1.

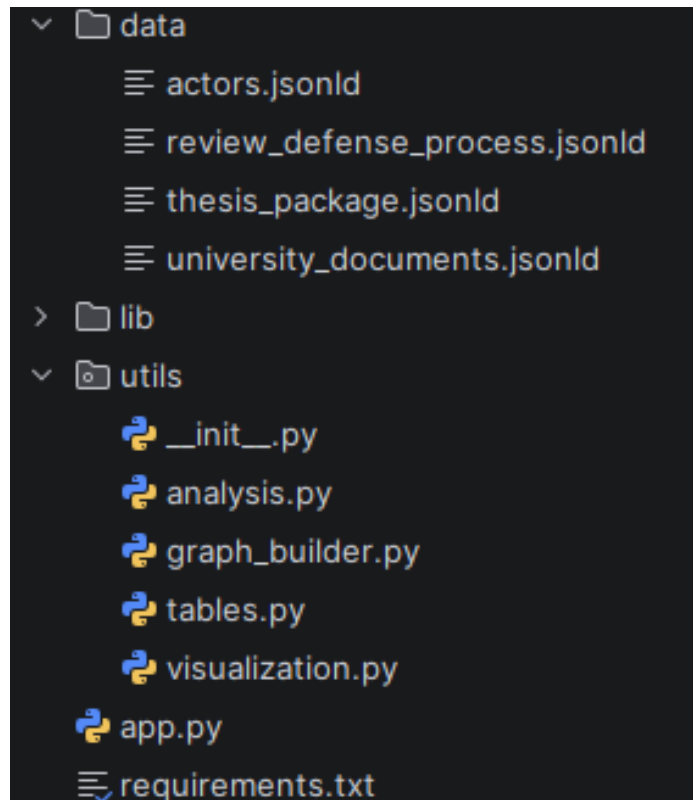


Рисунок 3.1 – Структура програмного проєкту

Для дослідження використовувалися кілька JSON-LD файлів, що описують різні частини предметної області: учасників процесу, університетські документи, структуру пояснювальної записки, а також процес перевірки, рецензування та захисту. Такий поділ демонструє джерельну гетерогенність даних.

3.3 Реалізація завантаження та обробки JSON-LD файлів

На початковому етапі реалізовано завантаження кількох JSON-LD файлів. Кожен файл описує окремий фрагмент предметної області, а система надалі інтегрує ці фрагменти в єдину графову модель знань.

Завантаження файлів здійснюється через вебінтерфейс системи. Користувач може вибрати один або декілька файлів формату “.json” або “.jsonld”, після чого система зчитує їхній вміст і передає дані на подальшу обробку.

Як зображено на рисунку 3.2, система підтримує одночасне завантаження кількох JSON-LD файлів. Це дозволяє розглядати вхідні дані як набір окремих джерел, які надалі інтегруються в єдину графову модель знань.

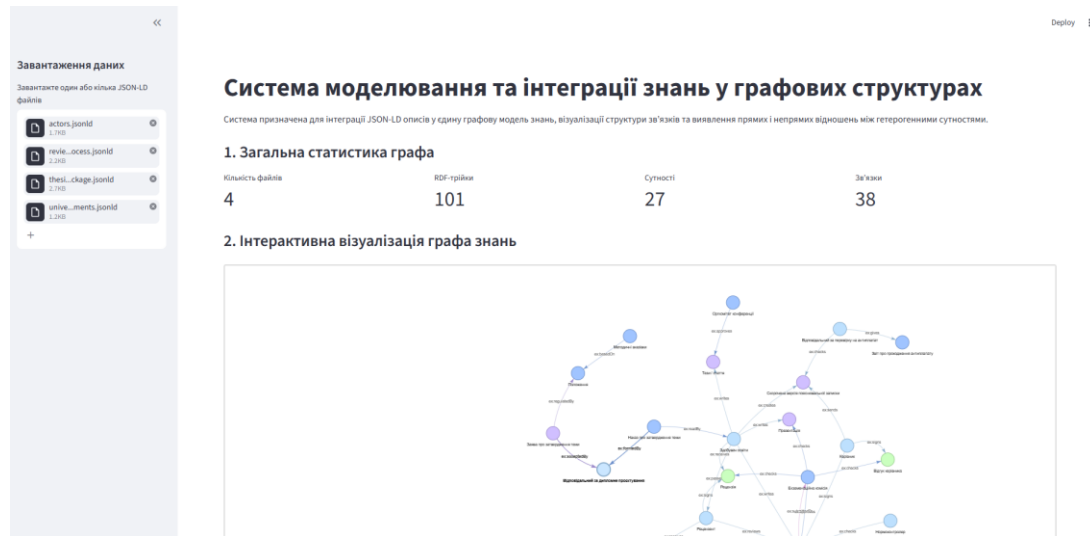


Рисунок 3.2 – Завантаження кількох JSON-LD файлів у системі

Після вибору файлів система зчитує їхній вміст, декодує у форматі UTF-8 і перетворює у Python-об'єкти за допомогою модуля json [47].

У результаті формується список JSON-LD об'єктів, де кожен елемент відповідає окремому завантаженому файлу. Це дозволяє обробляти кілька джерел без їх попереднього ручного об'єднання.

Вхідні дані поділено на логічні JSON-LD файли, що описують учасників процесу, університетські документи, структуру пояснювальної записки та етапи перевірки й захисту. Це дає змогу показати інтеграцію різних частин предметної області.

Після завантаження JSON-LD дані передаються до модуля “graph_builder.py”, де виконується їх перетворення у RDF-граф. Для цього використовується бібліотека RDFLib. Кожен JSON-LD об'єкт парситься окремо, але всі отримані RDF-триїки додаються до одного спільного графа.

У процесі обробки система створює спільний RDF-граф і послідовно додає до нього триїки з кожного JSON-LD файлу. Якщо одна сутність згадується у

різних файлах, вона об'єднується в межах одного вузла графа завдяки спільному ідентифікатору.

На цьому етапі гетерогенність проявляється у різних типах сутностей, різних зв'язках і кількох джерелах даних. Після формування RDF-графа система переходить до побудови моделі NetworkX.

Після формування RDF-графа система переходить до наступного етапу – перетворення RDF-трійок у графову модель NetworkX. У цій моделі URI-сутності стають вузлами графа, а предикати між ними – ребрами. Літеральні значення, наприклад назви документів або ролі учасників, зберігаються як атрибути вузлів.

3.4 Побудова RDF-графа та графової моделі NetworkX

Після завантаження JSON-LD файлів система формує RDF-граф і перетворює його у модель NetworkX, придатну для подальшого аналізу, візуалізації та пошуку зв'язків. Такий підхід дає змогу спочатку зберегти семантичну структуру даних у формі RDF, а потім використати можливості графового аналізу для дослідження взаємозв'язків між сутностями.

RDFLib подає кожен факт у формі трійки “суб’єкт – предикат – об’єкт”. Наприклад, зв’язок між здобувачем і пояснювальною запискою перетворюється на формалізоване твердження в RDF-графі. У такому поданні суб’єктом виступає початкова сутність, предикат визначає тип відношення, а об’єктом є інша сутність або значення певної властивості. Завдяки цьому дані набувають чіткої структури, придатної для автоматизованої обробки.

RDF-граф формується на основі кількох JSON-LD файлів, кожен із яких описує окремий фрагмент предметної області. Завдяки спільним ідентифікаторам ці фрагменти інтегруються в одну структуру. Наприклад, якщо в одному файлі описано здобувача освіти, а в іншому – пояснювальну записку та її розділи, то за наявності спільних посилань між ними система може об’єднати ці дані в єдину графову модель.

Після цього RDF-граф перетворюється у модель NetworkX. У ній URI-сутності подаються як вузли графа, предикати між сутностями – як ребра, а текстові або числові значення можуть використовуватися як атрибути відповідних вузлів. Саме така модель використовується далі для побудови таблиць сутностей і зв'язків, візуалізації графа та пошуку шляхів між об'єктами.

Приклади RDF-трійок, які формуються системою після обробки JSON-LD файлів, наведено в таблиці 3.2.

Таблиця 3.2 – Приклади RDF-трійок, сформованих системою

Суб'єкт	Предикат	Об'єкт
ex:student	ex:writes	ex:qualification_work_note
ex:qualification_work_note	ex:hasPart	ex:chapter_1
ex:qualification_work_note	ex:hasPart	ex:chapter_4
ex:supervisor	ex:checks	ex:qualification_work_note
ex:supervisor	ex:signs	ex:qualification_work_note
ex:reviewer	ex:reviews	ex:qualification_work_note
ex:norm_control	ex:checks	ex:qualification_work_note
ex:defense_committee	ex:checks	ex:presentation
ex:conference_committee	ex:approves	ex:thesis_article
ex:occupational_safety_consultant	ex:checks	ex:chapter_4

Як видно з таблиці 3.2, RDF-трійки дозволяють представити різні типи сутностей і зв'язків у єдиному форматі, зручному для подальшого перетворення у графову модель.

Під час перетворення у NetworkX URI-сутності стають вузлами, предикати між URI-сутностями – ребрами, літеральні значення – атрибутами вузлів, а rdf:type використовується для визначення типу сутності.

Тобто якщо об'єкт має власний ідентифікатор, наприклад “ex:student” або “ex:qualification_work_note”, він стає окремим вузлом графа. Якщо між двома URI-сутностями є предикат, наприклад “ex:writes”, то між відповідними вузлами

створюється ребро. Якщо ж об'єктом RDF-трійки є звичайне текстове значення, наприклад назва документа або ім'я учасника, воно зберігається як атрибут відповідного вузла.

Таким чином, на цьому етапі система переходить від набору окремих семантичних описів до єдиної графової структури, яка може бути використана для аналізу зв'язків між гетерогенними сутностями, побудова RDF-графа та його перетворення у NetworkX дозволили перейти від окремих JSON-LD файлів до єдиної графової структури знань.

3.5 Візуалізація графової моделі знань

Після формування графової структури система виконує її інтерактивну візуалізацію. Це дає змогу наочно перевірити результат інтеграції та побачити взаємозв'язки між сутностями.

Для візуалізації використано PyVis, який відображає сутності як вузли, а зв'язки між ними – як підписані ребра. Вузли автоматично зафарбовуються залежно від типу сутності [48].

Як зображено на рисунку 3.3, побудований граф містить різні типи сутностей і зв'язків, що демонструє інтеграцію гетерогенних даних у спільну модель знань.

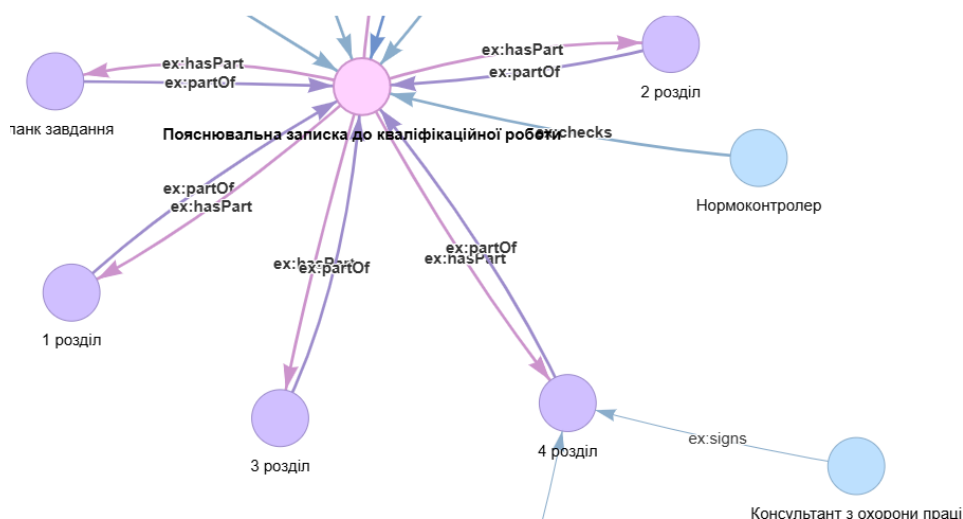


Рисунок 3.3 – Інтерактивна візуалізація інтегрованого графа знань

Таким чином, візуалізація виконує не лише ілюстративну, а й аналітичну функцію: вона допомагає перевірити правильність інтеграції та побачити основні вузли предметної області.

3.6 Формування таблиць сутностей і зв'язків графової структури

Для доповнення графової візуалізації система формує табличне подання результатів. Це дає змогу аналізувати побудовану модель у структурованому вигляді.

Після обробки JSON-LD файлів формуються дві основні таблиці: таблиця сутностей і таблиця зв'язків. Перша містить вузли графа та їх характеристики, друга – семантичні відношення між об'єктами [49].

Як зображено на рисунку 3.4, таблиця сутностей відображає основні характеристики кожного об'єкта графової моделі.

3. Таблиця сутностей

Entity ID	Label	Type	Degree	In-degree	Out-degree	
0 ex:chapter_1	1 розділ	ex:DocumentPart		2	1	1
1 ex:reviewer	Рецензент	ex:Person		4	1	3
2 ex:review	Рецензія	ex:ReviewDocument		3	3	0
3 ex:qualification_work_note	Пояснювальна записка до кваліфікаційної роботи	ex:CompositeDocument		16	10	6
4 ex:defense_committee	Екзаменаційна комісія	ex:Group		5	1	4
5 ex:civil_safety_consultant	Консультант з безпеки життєдіяльності	ex:Person		1	0	1
6 ex:chapter_4	4 розділ	ex:DocumentPart		4	3	1
7 ex:supervisor	Керівник	ex:Person		3	0	3
8 ex:short_note_version	Скорочена версія пояснювальної записки	ex:StudentDocument		3	3	0
9 ex:antiplagiarism_report	Звіт про проходження антиплагіату	ex:UniversityDocument		1	1	0

Рисунок 3.4 – Таблиця сутностей у системі

У таблиці сутностей кожен рядок відповідає окремому вузлу графа. Для нього подаються ідентифікатор, назва, тип, а також показники “Degree”, “In-degree” та “Out-degree”.

Наступним етапом є формування таблиці зв'язків. Як зображено на рисунку 3.5, ця таблиця відображає семантичні відношення між сутностями графа.

4. Таблиця зв'язків

	Source	Relation	Target
0	ex:chapter_1	ex:partOf	ex:qualification_work_note
1	ex:reviewer	ex:signs	ex:review
2	ex:reviewer	ex:receives	ex:review_assignment
3	ex:reviewer	ex:reviews	ex:qualification_work_note
4	ex:qualification_work_note	ex:submittedTo	ex:defense_committee
5	ex:qualification_work_note	ex:hasPart	ex:chapter_1
6	ex:qualification_work_note	ex:hasPart	ex:chapter_3
7	ex:qualification_work_note	ex:hasPart	ex:chapter_4
8	ex:qualification_work_note	ex:hasPart	ex:assignment_form
9	ex:qualification_work_note	ex:hasPart	ex:chapter_2

Рисунок 3.5 – Таблиця зв'язків у системі

У таблиці зв'язків кожен рядок відповідає окремому ребру графа. Поля “Source”, “Relation” і “Target” показують початкову сутність, тип відношення та цільову сутність.

Табличне подання є важливим доповненням до графової візуалізації, оскільки дозволяє детально перевірити склад сутностей і зв'язків після інтеграції.

3.7 Аналіз складу графа знань

Після побудови графової моделі було виконано аналіз її складу. Основна мета цього етапу полягала у визначенні кількості файлів, RDF-тріжок, сутностей, зв'язків і їх розподілу за типами.

Оскільки вхідні дані подані у кількох JSON-LD файлах, отриманий граф відображає інтеграцію різних фрагментів предметної області в одну структуру.

Загальні характеристики побудованого графа наведено в таблиці 3.3.

Таблиця 3.3 – Загальні характеристики побудованого графа

Показник	Значення
Кількість JSON-LD файлів	4
Кількість RDF-трійок	101
Кількість сутностей	27
Кількість зв'язків	38

Для детальнішого аналізу побудовано діаграми розподілу сутностей і зв'язків за типами. Вони дозволяють оцінити склад сформованого графа [50].

Як зображено на рисунку 3.6, у графі представлені різні типи сутностей, зокрема особи, групи, університетські документи, документи здобувача, частини документів і документи оцінювання.

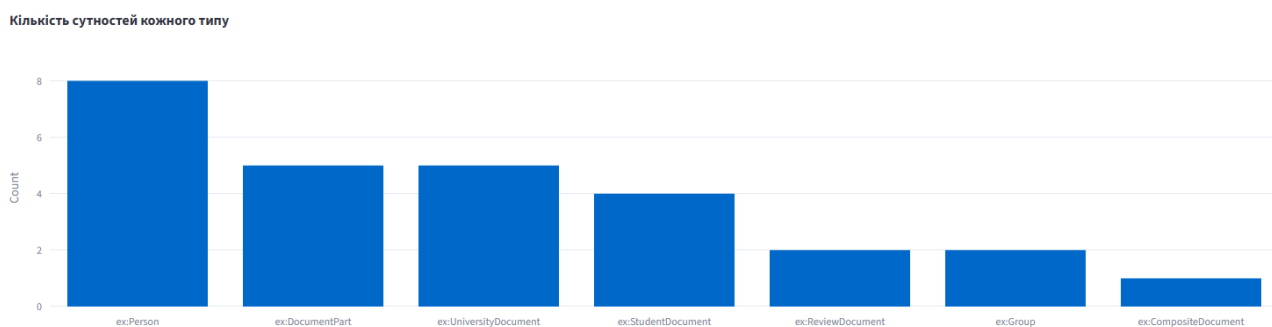


Рисунок 3.6 – Розподіл сутностей за типами

Наявність кількох типів сутностей підтверджує гетерогенність предметної області, оскільки процес виконання та захисту кваліфікаційної роботи включає різні за природою об'єкти.

Окрім аналізу сутностей, було виконано аналіз типів зв'язків між ними. Це дозволяє визначити, які відношення найчастіше використовуються у побудованій графовій моделі.

Як зображено на рисунку 3.7, у графі представлені різні типи зв'язків, зокрема “writes”, “checks”, “signs”, “hasPart”, “partOf”, “reviews”, “approves” та “submittedTo”.

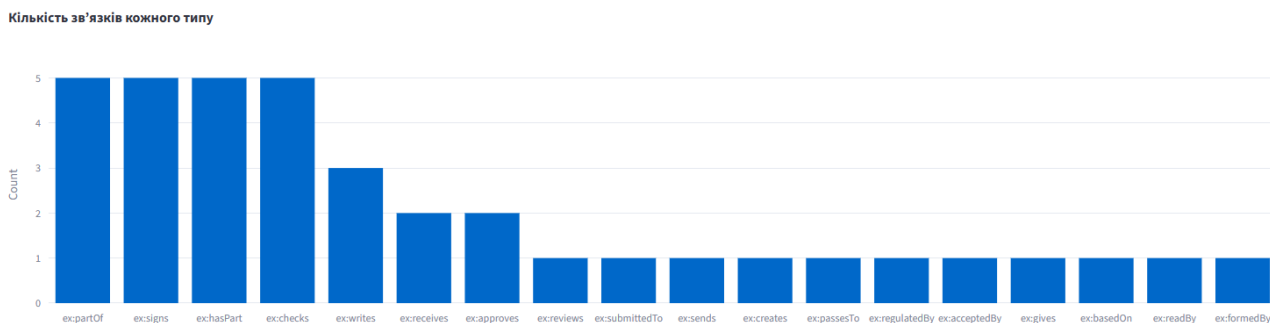


Рисунок 3.7 – Розподіл зв'язків за типами

Отримані результати показують, що побудована графова структура є придатною для подальшого аналізу зв'язків між гетерогенними сутностями. Зокрема, на основі сформованого графа можна визначати центральні елементи моделі, аналізувати прямі відношення та знаходити непрямі зв'язки між сутностями через проміжні вузли. Це створює основу для подальшого етапу дослідження – виявлення прямих і непрямих зв'язків між об'єктами предметної області.

3.8 Виявлення прямих і непрямих зв'язків між сутностями графа знань

Одним із ключових завдань системи є виявлення зв'язків між гетерогенними сутностями, інтегрованими в єдину графову модель знань.

У роботі розглядаються прямі та непрямі зв'язки. Прямий зв'язок існує тоді, коли між сутностями є безпосереднє ребро, а непрямий – коли шлях проходить через проміжні вузли.

Для пошуку непрямих зв'язків використано алгоритм найкоротшого шляху між двома вузлами графа. Користувач обирає початкову та кінцеву сутність, після чого система визначає послідовність пов'язаних об'єктів.

Таблиця 3.4 – Приклади виявлених прямих і непрямих зв'язків між сутностями

Початкова сутність	Кінцева сутність	Знайдений шлях	Тип зв'язку
Здобувач освіти	Пояснювальна записка	ex:student -> ex:qualification_work_note	прямий
Здобувач освіти	Екзаменаційна комісія	ex:student -> ex:qualification_work_note -> ex:defense_committee	непрямий
Керівник	Рецензент	ex:supervisor -> ex:qualification_work_note -> ex:reviewer	непрямий
Консультант з охорони праці	Пояснювальна записка	ex:occupational_safety_consultant -> ex:chapter_4 -> ex:qualification_work_note	непрямий
Оргкомітет конференції	Здобувач освіти	ex:conference_committee -> ex:thesis_article -> ex:student	непрямий

Як видно з таблиці 3.4, графова модель дозволяє виявляти як прямі, так і непрямі зв'язки. Це особливо важливо для предметної області, де більшість залежностей проходить через документи або процеси перевірки.

Після виконання пошуку система відображає результат у вигляді текстового шляху між сутностями. Як зображено на рисунку 3.8, користувач бачить послідовність вузлів і типи зв'язків між ними.

7. Пошук зв'язку між сутностями

Початкова сутність: `ex:civil_safety_consultant` | Кінцева сутність: `ex:antiplagiarism_responsible`

Знайти найкоротший шлях

Зв'язок знайдено.

Шлях:

```
ex:civil_safety_consultant → ex:chapter_4 → ex:qualification_work_note → ex:supervisor → ex:short_note_version → ex:antiplagiarism_responsible
```

Інтерпретація зв'язків:

- `ex:civil_safety_consultant --ex:signs--> ex:chapter_4`
- `ex:chapter_4 --ex:partOf-> ex:qualification_work_note`
- `ex:supervisor --ex:signs-> ex:qualification_work_note`
- `ex:supervisor --ex:sends-> ex:short_note_version`
- `ex:antiplagiarism_responsible --ex:checks-> ex:short_note_version`

Рисунок 3.8 – Результат пошуку найкоротшого шляху між сутностями

Таким чином, механізм пошуку шляхів підтверджує, що система може використовуватися для виявлення прямих і непрямих зв'язків у складній предметній області.

3.9 Аналіз центральності вузлів графа знань

Після побудови інтегрованої графової моделі доцільно визначити, які сутності відіграють у ній найбільш важливу роль. Для цього використано показник “degree centrality”.

Центральність вузла показує, наскільки активно певна сутність пов'язана з іншими елементами предметної області. Вузли з більшою кількістю зв'язків виконують важливішу роль у структурі графа.

Результати аналізу центральності подано у вигляді таблиці, де вказано ідентифікатор сутності, її назву, тип і значення показника. Фрагмент результатів наведено в таблиці 3.5.

Таблиця 3.5 – Найбільш центральні сутності графа

Сутність	Тип	Центральність
Пояснювальна записка до кваліфікаційної роботи	CompositeDocument	0.6154
Здобувач освіти	Person	0.2692
Екзаменаційна комісія	Group	0.1923
Рецензент	Person	0.1538
4 розділ	DocumentPart	0.1538

Як видно з таблиці 3.5, найбільш центральною сутністю у побудованій моделі є пояснювальна записка до кваліфікаційної роботи, оскільки вона поєднує різні фрагменти предметної області.

Топ-10 центральних сутностей наведено на рисунку 3.9.

8. Центральність вузлів графа

Entity ID	Label	Type	Degree centrality
3	exqualification_work_note	ex:CompositeDocument	0.6154
10	ex:student	ex:Person	0.2692
4	ex:defense_committee	ex:Group	0.1923
1	ex:reviewer	ex:Person	0.1538
6	ex:chapter_4	ex:DocumentPart	0.1538
8	ex:short_note_version	ex:StudentDocument	0.1154
2	ex:review	ex:ReviewDocument	0.1154
7	ex:supervisor	ex:Person	0.1154
0	ex:chapter_1	ex:DocumentPart	0.0769
12	ex:topic_approval_application	ex:StudentDocument	0.0769

Топ-10 центральних сутностей графа

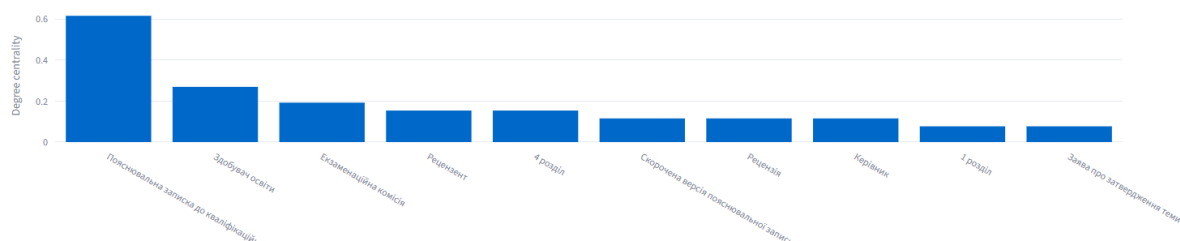


Рисунок 3.9 – Топ-10 центральних сутностей графа

Отримані результати підтверджують, що графова модель дозволяє не лише зберігати різномірні сутності, а й аналізувати їхню роль у загальній структурі знань.

3.10 Оцінювання результатів реалізації системи інтеграції знань

Оцінювання результатів здійснювалося на основі тестового набору JSON-LD файлів, які описують різні фрагменти предметної області виконання та захисту кваліфікаційної роботи.

У результаті завантаження файлів система сформувала єдиний RDF-граф, у якому дані з різних джерел були об'єднані у спільну графову структуру. Подальше перетворення у NetworkX забезпечило можливість візуалізації та аналізу зв'язків.

Узагальнення функціональних можливостей системи наведено в таблиці 3.6.

Таблиця 3.6 – Оцінювання функціональних можливостей системи

Функціональна можливість	Реалізовано	Отриманий результат
Завантаження кількох JSON-LD файлів	Так	Система дозволяє завантажити декілька окремих файлів, які описують різні фрагменти предметної області
Формування RDF-графа	Так	Дані з усіх файлів перетворюються у RDF-трійки та об'єднуються в один граф
Побудова графової моделі NetworkX	Так	RDF-граф перетворюється у спрямований граф, у якому вершини відповідають сутностям, а ребра – зв'язкам
Визначення типів сутностей	Так	Система зберігає типи об'єктів, наприклад Person, Group, CompositeDocument, DocumentPart, ReviewDocument

Функціональна можливість	Реалізовано	Отриманий результат
Визначення типів зв'язків	Так	У графі зберігаються різні типи відношень: writes, checks, signs, hasPart, reviews, approves
Інтерактивна візуалізація графа	Так	Побудований граф можна переглядати в інтерфейсі, переміщувати вузли та аналізувати зв'язки між ними
Формування таблиці сутностей	Так	Система формує таблицю з ідентифікатором, назвою, типом і показниками зв'язності кожної сутності
Формування таблиці зв'язків	Так	Система формує таблицю, у якій відображено початкову сутність, тип зв'язку та цільову сутність
Побудова діаграм	Так	Система будує діаграми розподілу сутностей і зв'язків за типами
Пошук найкоротшого шляху	Так	Система дозволяє знайти прямий або непрямий шлях між двома вибраними сутностями
Аналіз центральності вузлів	Так	Система визначає найбільш зв'язані сутності графа

Система реалізує основні функції, необхідні для демонстрації інтеграції знань у графовій структурі: завантаження JSON-LD файлів, формування RDF-графа, побудову графової моделі, візуалізацію, табличне подання та пошук зв'язків. Отже, результати реалізації підтверджують, що розроблена система забезпечує повний цикл роботи з графовою моделлю знань: від завантаження JSON-LD файлів до побудови RDF-графа, візуалізації, аналізу структури та виявлення зв'язків між гетерогенними сутностями.

3.11 Висновки до третього розділу

У третьому розділі було описано реалізацію програмної системи для моделювання та інтеграції знань у графових структурах. Система забезпечує завантаження кількох JSON-LD файлів, їх перетворення у RDF-граф, побудову графової моделі, візуалізацію отриманої структури та аналіз зв'язків між сутностями.

Практична реалізація підтвердила можливість інтеграції даних із кількох джерел у єдину графову модель знань. У межах системи різні фрагменти предметної області, зокрема учасники процесу, документи, частини документів, групи та процеси перевірки, об'єднуються в одну зв'язану структуру.

Побудована графова модель дозволяє представляти гетерогенні сутності у вигляді вузлів, а зв'язки між ними – у вигляді ребер. Це дає змогу не лише зберігати дані, а й аналізувати їхню структуру, визначати найбільш пов'язані об'єкти та встановлювати змістові залежності між різними типами сутностей.

Окремо було реалізовано пошук прямих і непрямих зв'язків між сутностями. Це підтверджує, що графова модель є зручною для виявлення неочевидних залежностей, які можуть проходити через проміжні об'єкти.

Отже, результати реалізації показали практичну придатність запропонованого підходу. JSON-LD може використовуватися для семантичного опису різномірних даних, RDF – для їх формалізованого подання, а графова структура – для інтеграції, візуалізації та аналізу зв'язків між гетерогенними сутностями.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Принципи ергономіки робочого місця

Ергономіка досліджує працю людини в умовах сучасного виробництва, щоб удосконалити засоби праці, організувати робочий процес найбільш ефективно та створити комфортні умови для працівника [51].

За даними наукових досліджень у сфері безпеки та гігієни праці, приблизно чверть причин лікарняних серед офісних працівників пов'язана з проблемами спини. Через захворювання опорно-рухового апарату підприємства щороку зазнають значних збитків, що оцінюються в мільярди доларів. Багато працівників щодня не виходять на роботу через так званий офісний синдром – сукупність симптомів, які виникають унаслідок тривалого перебування в одній позі, переважно під час роботи за комп'ютером в офісі. До них належать міофасціальний больовий синдром, синдром зап'ястного каналу, тендиніт та інші порушення.

Саме тому створення ергономічного робочого місця, зручного середовища та відповідних меблів для людини є важливим завданням.

Робоче місце – це частина простору, обладнана всім необхідним устаткуванням, де виконує свої трудові обов'язки один працівник або група працівників.

Раціональна організація робочого місця повинна забезпечувати зручне розташування знарядь і предметів праці, запобігати дискомфорту, знижувати рівень втоми працівника та сприяти підвищенню продуктивності. Площа робочого місця має бути достатньою для того, щоб працівник не виконував зайвих рухів і не відчував незручностей під час роботи. Також важливо, щоб людина могла змінювати робочу позу, зокрема положення тулуба, рук і ніг. Водночас фізіологічно неприродні та незручні положення тіла потрібно усувати або зводити до мінімуму.

Результати досліджень свідчать, що за умови правильної організації робочих місць продуктивність праці може підвищуватися на 15–25%.

В Україні головні ергономічні вимоги до проектування робочого місця користувача комп'ютера в системі «людина – техніка – виробниче середовище» регулюються державними стандартами. До них належать: ДСТУ 8604:2015 «Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги»; ДСТУ 7299:2013 «Дизайн і ергономіка. Робоче місце оператора. Взаємне розташування елементів робочого місця. Загальні вимоги ергономіки»; ДСТУ 7234:2011 «Дизайн і ергономіка. Обладнання виробниче. Загальні вимоги дизайну та ергономіки» [52].

Організація робочого місця включає:

- правильне розташування робочого місця у виробничому приміщенні;
- вибір обґрунтованої з погляду ергономіки робочої пози та виробничих меблів з урахуванням антропометричних особливостей людини;
- раціональне розміщення обладнання на робочому місці;
- урахування змісту, характеру та специфіки трудової діяльності.

Статичне напруження працівника під час роботи виникає через необхідність утримувати предмети й знаряддя праці в нерухомому положенні, а також через тривале збереження робочої пози.

Робоча поза – це основне положення тіла працівника у просторі під час виконання роботи. Зручна робоча поза повинна забезпечувати стійке положення тулуба, ніг, рук і голови, потребувати мінімальних енергетичних витрат та сприяти високій ефективності праці.

Найчастіше під час трудової діяльності використовують положення сидячи або стоячи. Під час проектування робочого місця слід враховувати, що для роботи з фізичним навантаженням більш доцільною є поза стоячи, а для роботи з невеликими зусиллями – поза сидячи.

Положення стоячи викликає більшу втому, ніж положення сидячи. Воно потребує приблизно на 10% більше енергії, може спричинити підвищення

артеріального й венозного тиску, розширення вен на ногах, ушкодження ступень та викривлення хребта.

Під час виконання роботи в положенні сидячи нижня частина тіла перебуває у більш розслабленому стані, тоді як основне статичне навантаження припадає на м'язи шиї, спини, таза та стегон. Неправильна сидяча поза може призводити до застою крові в нижніх кінцівках, а за умови значного навантаження на пальці рук – до запалення суглобів.

Організація робочого місця користувача комп'ютера має забезпечувати відповідність усіх його елементів, а також їхнього взаємного розміщення, ергономічним вимогам, наведеним на рисунку 4.1.

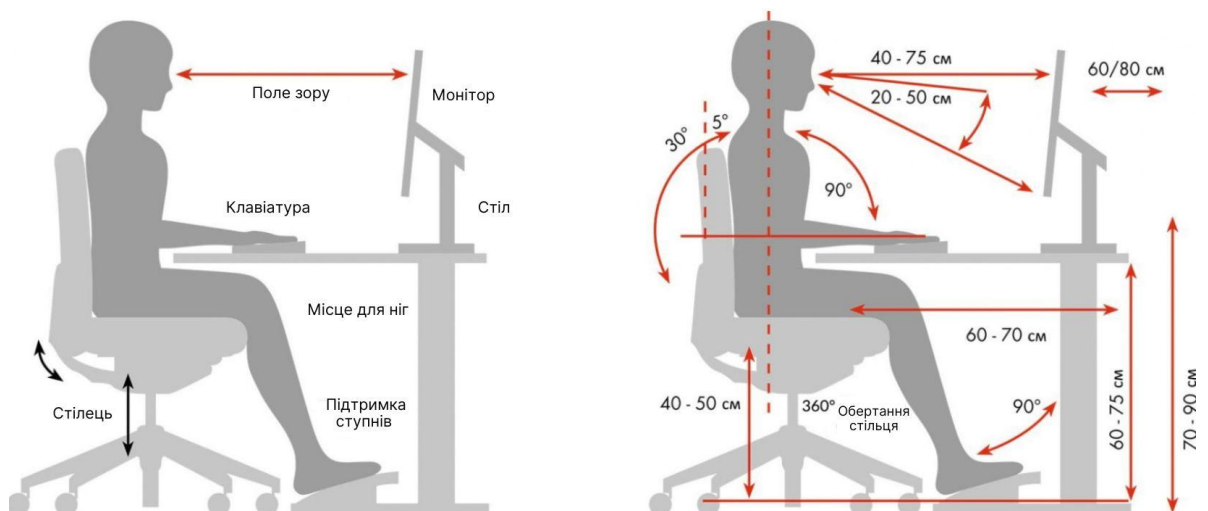


Рисунок 4.1 – Ергономічно ідеальне облаштування робочого місця

Під час організації робочого місця слід враховувати такі ергономічні аспекти інтер'єру:

Підтримка тіла – це створення таких умов, за яких працівник може зручно й правильно розмістити тіло під час роботи, зменшити навантаження на хребет та мінімізувати ризик появи болю й м'язового напруження.

Для забезпечення належної підтримки тіла потрібно звернути увагу на такі чинники:

- Сидіння повинно мати можливість регулювання висоти, спинки та підлокітників. Воно має бути достатньо гнучким і зручним, щоб пристосовуватися до форми тіла та забезпечувати підтримку спини, шиї й попереку. Для додаткової підтримки шиї можна використовувати подушку або спеціальний підголівник.

- Стілець слід обирати з правильною формою та належною підтримкою, щоб він забезпечував стабільну основу і комфортне положення під час сидіння. Регулювання висоти дає змогу пристосувати його до індивідуальних потреб користувача.

- Підставка для ніг сприяє зменшенню навантаження на нижню частину спини та покращує кровообіг у ногах.

- Ергономічна клавіатура та миша з підтримкою зап'ястя допомагають знизити навантаження на руки й запобігти розвитку синдрому карпального каналу. Їх потрібно розміщувати на такій висоті та відстані, щоб руки й зап'ястя перебували у природному, зручному положенні.

Розміщення меблів і дотримання правильних відстаней допомагають створити зручні умови для роботи, гармонійний дизайн приміщення, зменшити навантаження на тіло та підвищити ефективність діяльності. Під час розташування меблів і обладнання слід враховувати такі принципи:

- Висота робочого столу та крісла повинна регулюватися, щоб їх можна було налаштувати відповідно до індивідуальних потреб користувача. Робочий стіл має бути достатньо просторим для розміщення всіх необхідних предметів і забезпечення свободи рухів. Відстань між робочою поверхнею та нижньою частиною стільниці повинна дозволяти зручно розміщувати ноги.

- Монітор потрібно встановлювати на рівні очей, щоб зменшити напруження шиї та очей. Оптимальна відстань від очей до екрана має становити приблизно 50–70 см [53].

- Меблі в робочій зоні слід розташовувати так, щоб зменшити потребу в частих поворотах і нахилах тулуба. Підтримувати правильне положення тіла допомагає ергономічне крісло.

- Вільний простір перед столом, кріслом та іншими робочими елементами має бути достатнім, щоб користувач міг без перешкод рухатися, змінювати положення тіла та розтягуватися під час роботи.

Правильно організоване освітлення допомагає уникнути перевтоми очей, знижує ризик появи напруження й головного болю, а також сприяє кращій концентрації та підтриманню позитивного настрою. У приміщенні варто максимально використовувати природне світло. Меблі доцільно розміщувати так, щоб вікно або інше джерело світла знаходилося збоку або спереду. Для регулювання інтенсивності освітлення можна використовувати штори чи жалюзі.

Для штучного освітлення слід обирати стабільні лампи з природним світлом, які не створюють мерехтіння та відблисків. Також варто звертати увагу на можливість регулювання яскравості й колірної температури світла, що дозволяє пристосувати освітлення до індивідуальних потреб і вподобань користувача.

Доступність передбачає, що всі необхідні меблі, матеріали та обладнання мають бути розміщені зручно для користувача й відповідати його робочій висоті. Важливо, щоб усі потрібні предмети та інструменти були поруч із робочим місцем або безпосередньо на його поверхні. Їх слід розташовувати так, щоб працівник міг легко дістати їх без сильних нахилів, прогинів чи надмірного витягування рук. Для зручності варто використовувати системи зберігання й організації простору: шухляди, полиці, шафи та контейнери.

Загальні принципи організації робочого місця:

- на робочому місці не має бути зайвих предметів; усе необхідне для роботи повинно знаходитися поруч із працівником, але не створювати перешкод;
- предмети, які використовуються найчастіше, слід розміщувати ближче, ніж ті, до яких звертаються рідше;
- речі, якими користуються лівою рукою, доцільно розташовувати зліва, а ті, які беруть правою рукою, – справа;

- якщо предмет або пристосування використовується обома руками, його місце потрібно визначати з урахуванням зручності захоплення двома руками;
- робоче місце не повинно бути перевантажене або захаращене;
- організація робочого простору має забезпечувати працівнику достатній огляд.

4.2 Правила електробезпеки при роботі з електронними пристроями

У ХХІ столітті складно уявити життя без електронних пристроїв. Вони дають змогу постійно підтримувати зв'язок із рідними та близькими, допомагають швидше, легше й ефективніше здобувати освіту, а також сприяють продуктивнішій роботі. Значна частина трудової діяльності сьогодні вже пов'язана з використанням комп'ютера. Проте робота майже всіх електронних пристроїв залежить від електроенергії, тому під час їх використання необхідно дотримуватися правил електробезпеки.

Електробезпека – це система організаційних і технічних заходів та засобів, спрямованих на захист людини від небезпечного й шкідливого впливу електричного струму, електричної дуги, електромагнітного поля та статичної електрики [54].

Правильне використання електроенергії допомагає запобігти випадкам ураження електричним струмом. Під час користування електроенергією слід дотримуватися таких основних вимог:

1. Візуальна перевірка обладнання – перед увімкненням електронного пристрою необхідно оглянути його шнур і переконатися, що він не має пошкоджень, щоб уникнути короткого замикання або ураження струмом.

2. Захист від коротких замикань – автоматичні вимикачі та пробкові запобіжники в електропроводці мають бути справними. Заміна заводських запобіжників будь-якими металевими провідниками, навіть тимчасово, може спричинити нещасний випадок або пожежу.

3. Заземлення – електроприлад повинен бути надійно заземлений відповідно до правил його встановлення.

4. Правильне підключення – потрібно користуватися лише справними та сертифікованими приладами. Не можна одночасно підключати багато пристроїв до одного подовжувача, щоб не перевантажити електромережу. Електрошнур слід спочатку під'єднати до приладу, а вже потім – до мережі.

5. Захист від вологи – електронні пристрої не можна використовувати в умовах підвищеної вологості або поруч із водою, оскільки це може призвести до короткого замикання. Заборонено працювати з електроприладами мокрими руками.

6. Очищення від пилу – пристрої потрібно регулярно очищувати від пилу, щоб запобігти їх перегріванню або загорянню.

7. Запобігання перегріву – необхідно забезпечити достатню вентиляцію для електроприладів, щоб вони не перегрівалися. Не слід накривати прилади під час їх роботи.

8. Вимикання пристроїв – коли прилад не використовується, його потрібно вимикати з розетки. Це зменшує ризик пожежі або пошкодження пристрою під час перепадів напруги.

9. Заборона самостійного ремонту – категорично заборонено самостійно виконувати будь-які ремонтні роботи з електроприладами.

Якщо електричне обладнання загорілося, насамперед необхідно негайно відключити живлення в приміщенні або на відповідній ділянці електромережі. Це дає змогу зменшити ризик подальшого поширення пожежі та запобігти ураженню працівників електричним струмом. Якщо знеструмити електромережу неможливо, слід пам'ятати, що для гасіння електрообладнання, яке перебуває під напругою, категорично заборонено використовувати воду та пінні вогнегасники, оскільки вони проводять електричний струм і можуть створити додаткову небезпеку для людини. У такому випадку дозволяється застосовувати лише порошкові вогнегасники, призначені для гасіння електроустановок.

Порятунок людини, яка зазнала ураження електричним струмом, часто залежить від того, наскільки швидко, правильно та обережно діють особи, які надають допомогу. Передусім потрібно якнайшвидше припинити дію електричного струму на потерпілого. Якщо немає можливості вимкнути електрообладнання з мережі, необхідно одразу звільнити потерпілого від струмоведучих частин, не торкаючись його безпосередньо руками. Для цього слід використовувати сухі предмети, які не проводять електричний струм, наприклад дерев'яну палицю, дошку, сухий одяг або інші ізолювальні матеріали. Після звільнення потерпілого від дії струму необхідно оцінити його стан, за потреби викликати екстрену медичну допомогу та надати домедичну допомогу до прибуття медичних працівників.

Під час роботи з комп'ютерною технікою також слід дотримуватися вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями, визначених НПАОП 0.00-7.15-18, а також Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин. Дотримання цих вимог спрямоване на запобігання перевтомі працівників, зниження навантаження на органи зору, опорно-руховий апарат і нервову систему. Робоче місце користувача комп'ютерної техніки має бути організоване таким чином, щоб забезпечити зручне положення тіла, достатнє освітлення, безпечне розміщення обладнання та вільний доступ до робочої зони.

До роботи на комп'ютеризованому робочому місці з екранними пристроями можуть допускатися лише працівники, які пройшли вступний і первинний інструктаж з охорони праці та пожежної безпеки. Працівник повинен знати основні правила безпечної експлуатації комп'ютерної техніки, порядок дій у разі виникнення аварійної ситуації, пожежі або виявлення несправності обладнання. Забороняється самостійно розбирати, ремонтувати або підключати електрообладнання без відповідного дозволу та спеціальної підготовки.

Користувач персональної електронно-обчислювальної машини повинен проходити повторний інструктаж з питань охорони праці один раз на шість місяців [55]. Проведення такого інструктажу дає змогу підтримувати належний рівень знань працівників щодо безпечної роботи з комп'ютерною технікою, правил пожежної безпеки та порядку дій у разі виникнення небезпечних ситуацій.

4.3 Висновки до четвертого розділу

У четвертому розділі розглянуто основні вимоги до безпечної організації робочого місця користувача під час роботи з комп'ютерною технікою. Визначено, що ефективна робота з програмною системою потребує не лише технічної справності обладнання, а й дотримання ергономічних умов, зокрема правильного розміщення монітора, клавіатури, миші, робочого стола та крісла.

Окрему увагу приділено вимогам до роботи з екранними пристроями. Дотримання оптимальної робочої пози, достатнього освітлення, раціонального режиму праці та перерв сприяє зменшенню втоми, напруження зору, статичного навантаження та загального дискомфорту під час тривалої роботи за комп'ютером.

Також у розділі узагальнено правила електробезпеки під час використання електронних пристроїв. Наголошено на необхідності перевірки справності кабелів, розеток, блоків живлення, недопущенні перевантаження електромережі, використанні пошкодженого обладнання та самостійному ремонті пристроїв без відповідної підготовки.

Отже, дотримання ергономічних вимог, правил охорони праці та електробезпеки є важливою умовою безпечного використання програмної системи. Це дозволяє зменшити ризики для здоров'я користувача, забезпечити стабільну роботу комп'ютерного обладнання та створити належні умови для виконання дослідницьких і практичних завдань кваліфікаційної роботи.

ВИСНОВКИ

У кваліфікаційній роботі досліджено та реалізовано підхід до моделювання й інтеграції знань у графових структурах для виявлення зв'язків між гетерогенними сутностями. Поставлену мету роботи досягнуто шляхом розроблення графової моделі знань, інтеграції JSON-LD описів у RDF-граф та реалізації програмної системи для побудови, візуалізації й аналізу зв'язків між сутностями. Запропонований підхід дозволяє поєднувати різнорідні дані в єдиному інформаційному просторі та використовувати графову структуру для подальшого дослідження взаємозв'язків.

Проаналізовано сучасні підходи до представлення знань, гетерогенних даних і графових структур. Визначено, що графові моделі забезпечують ефективне відображення складних взаємозв'язків між об'єктами різної природи, а використання семантичних технологій сприяє підвищенню структурованості та інтерпретованості даних. Це є особливо важливим у випадках, коли інформація надходить із різних джерел і має різну структуру, формат або рівень деталізації.

Досліджено особливості використання RDF і JSON-LD для семантичного подання та інтеграції знань. Встановлено, що RDF забезпечує формалізований опис сутностей і зв'язків, тоді як JSON-LD дозволяє інтегрувати семантичні дані у зручному для обробки форматі, що спрощує обмін інформацією між різними системами. Поєднання цих технологій дає змогу зберігати семантичний зміст даних і водночас забезпечувати їхню практичну придатність для програмної обробки.

Розроблено модель подання гетерогенних сутностей і зв'язків у єдиній графовій структурі. У межах моделі визначено основні типи сутностей, зв'язків між ними та принципи їх інтеграції. Запропонований підхід забезпечує уніфіковане представлення даних із різних джерел, підтримує опис різнорідних типів об'єктів та їхніх взаємозв'язків і створює основу для подальшого аналізу знань. Така модель дає можливість не лише зберігати окремі факти, а й простежувати логічні та семантичні зв'язки між елементами предметної області.

Спроектовано та реалізовано програмну систему для побудови, візуалізації й аналізу графа знань. Система забезпечує завантаження JSON-LD файлів, формування RDF-графа, перетворення даних у графову модель NetworkX, візуалізацію структури знань, формування таблиць сутностей і зв'язків, а також виконання базового аналізу графа. Реалізований функціонал дозволяє наочно відображати структуру знань і швидко визначати ключові сутності та зв'язки в побудованій моделі.

Проведено аналіз отриманих результатів і оцінено ефективність запропонованого підходу для виявлення зв'язків між гетерогенними сутностями. У результаті було показано, що побудована графова структура підвищує якість інтеграції даних, полегшує пошук прихованих взаємозв'язків і забезпечує більш повне представлення знань у порівнянні з традиційними підходами. Отримані результати підтверджують доцільність використання графових структур і семантичних технологій для роботи з різнорідними даними.

ПЕРЕЛІК ДЖЕРЕЛ

1. DAMA International. DAMA-DMBOK: Data Management Body of Knowledge. 2nd ed. Technics Publications, 2017. 624 p.
2. Doan A., Halevy A., Ives Z. Principles of Data Integration. Morgan Kaufmann, 2012. 520 p.
3. Hogan, Aidan, et al. "Knowledge graphs." *ACM Computing Surveys (Csur)* 54.4 (2021): 1-37.
4. Гашинський Р. Використання json-ld для семантичного подання та інтеграції знань, 2026.
5. Brevus, V., Brevus, H., Gashynskiy, R., Kashosi, A., & Yuzefovych, R. AI-driven multimodal data fusion for hazardous object detection in maritime and coastal environments, 2025.
6. Russell, Stuart J. *Artificial intelligence a modern approach*. Pearson Education, Inc., 2010.
7. Codd, Edgar F. "A relational model of data for large shared data banks." *Communications of the ACM* 13.6 (1970): 377-387.
8. Davis, Randall, Howard Shrobe, and Peter Szolovits. "What is a knowledge representation?." *AI magazine* 14.1 (1993): 17-17.
9. Brachman, Ronald, and Hector Levesque. *Knowledge representation and reasoning*. Elsevier, 2004.
10. Sowa, John F. "Knowledge Representation: Logical, Philosophical, and Computational Foundations."
11. Gruber, Thomas R. "Toward principles for the design of ontologies used for knowledge sharing?." *International journal of human-computer studies* 43.5-6 (1995): 907-928.
12. Noy, Natalya Fridman, and Mark A. Musen. "Evaluating Ontology-Mapping Tools: Requirements and Experience." *EON*. 2002.
13. Kawarabayashi, Ken-ichi, and Bojan Mohar. "Some recent progress and applications in graph minor theory." *Graphs and combinatorics* 23.1 (2007): 1-46.

14. RDF 1.1 Concepts and Abstract Syntax. W3C Recommendation. 25 February 2014. URL: <https://www.w3.org/TR/rdf11-concepts/>
15. Berners-Lee, Tim, James Hendler, and Ora Lassila. "Web semantic." *Scientific American* 284.5 (2001): 34-43.
16. Ehrlinger, Lisa, and Wolfram Wöß. "Towards a definition of knowledge graphs." *SEMANTiCS (Posters, Demos, SuCCESS)* 48.1-4 (2016): 2.
17. Angles, Renzo, and Claudio Gutierrez. "Survey of graph database models." *ACM computing surveys (CSUR)* 40.1 (2008): 1-39.
18. Robinson, Ian, Jim Webber, and Emil Eifrem. *Graph databases: new opportunities for connected data*. " O'Reilly Media, Inc.", 2015.
19. Shadbolt, Nigel, Tim Berners-Lee, and Wendy Hall. "The semantic web revisited." *IEEE intelligent systems* 21.3 (2006): 96-101.
20. Ziegler, Patrick, and Klaus R. Dittrich. "Three decades of data integration—All problems solved?." Building the Information Society: IFIP 18th World Computer Congress Topical Sessions 22–27 August 2004 Toulouse, France. Boston, MA: Springer US, 2004.
21. Halevy, Alon Y. "Answering queries using views: A survey." *The VLDB Journal* 10.4 (2001): 270-294.
22. Bellahsene Z., Bonifati A., Rahm E. *Schema Matching and Mapping*. Springer, 2011. 320 p.
23. JSON-LD 1.1: A JSON-based Serialization for Linked Data. W3C Recommendation. 16 July 2020. URL: <https://www.w3.org/TR/json-ld11/>
24. Heath T., Bizer C. *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool, 2011. 136 p.
25. Data on the Web Best Practices. W3C Recommendation. 31 January 2017. URL: <https://www.w3.org/TR/dwbp/>
26. RDF Schema 1.1. W3C Recommendation. 25 February 2014. URL: <https://www.w3.org/TR/rdf-schema/>
27. OWL 2 Web Ontology Language Document Overview. W3C Recommendation. 11 December 2012. URL: <https://www.w3.org/TR/owl2-overview/>

28. SPARQL 1.1 Query Language. W3C Recommendation. 21 March 2013. URL: <https://www.w3.org/TR/sparql11-query/>
29. Turtle: Terse RDF Triple Language. W3C Recommendation. 25 February 2014. URL: <https://www.w3.org/TR/turtle/>
30. Euzenat J., Shvaiko P. *Ontology Matching*. 2nd ed. Springer, 2013. 512 p.
31. Lenzerini M. *Data Integration: A Theoretical Perspective*. Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems. 2002. P. 233–246.
32. Bizer C., Heath T., Berners-Lee T. *Linked Data – The Story So Far*. International Journal on Semantic Web and Information Systems. 2009. Vol. 5, No. 3.
33. Paulheim H. *Knowledge Graph Refinement: A Survey of Approaches and Evaluation Methods*. Semantic Web. 2017. Vol. 8, No. 3. P. 489–508.
34. Newman M. *Networks: An Introduction*. Oxford University Press, 2010. 784 p.
35. Freeman L. C. *Centrality in Social Networks: Conceptual Clarification*. Social Networks. 1979. Vol. 1, No. 3. P. 215–239.
36. Brandes U. *A Faster Algorithm for Betweenness Centrality*. Journal of Mathematical Sociology. 2001. Vol. 25, No. 2. P. 163–177.
37. Nickel M., Murphy K., Tresp V., Gabrilovich E. *A Review of Relational Machine Learning for Knowledge Graphs*. Proceedings of the IEEE. 2016. Vol. 104,
38. RDFLib Documentation. URL: <https://rdflib.readthedocs.io/>
39. NetworkX Documentation. URL: <https://networkx.org/documentation/stable>
40. Streamlit Documentation. URL: <https://docs.streamlit.io/>
41. PyVis Documentation. URL: <https://pyvis.readthedocs.io/>
42. pandas Documentation. URL: <https://pandas.pydata.org/docs/>
43. Plotly Python Open Source Graphing Library Documentation. URL: <https://plotly.com/python/>
44. Бабак В.П., Куц Ю.В., Мислович М.В., Фриз М.Є., Щербак Л.М. *Об’єктно-орієнтована ідентифікація стохастичних шумових сигналів*. Київ: Наукова думка, 2024. 240 с. <https://doi.org/10.15407/978-966-00-1883-9> .

45. V. Babak, A. Zaporozhets, Y. Kuts, M. Fryz, L. Scherbak. Noise signals: Modelling and Analyses. Cham: Springer Nature Switzerland, 2025. 222 p. DOI: <https://doi.org/10.1007/978-3-031-71093-3>
46. A. Zaporozhets, Y. Kuts, B. Mlynko, M. Fryz, and L. Scherbak, “EEG Signal Classification Using Linear Process Model-Based Feature Extraction and Supervised Learning,” in *Advanced System Development Technologies II. Studies in Systems, Decision and Control*, M. Bezuglyi, N. Bouraou, V. Mykytenko, G. Tymchyk, and A. Zaporozhets, Eds., Cham: Springer Nature Switzerland, 2025, pp. 235–257. doi: 10.1007/978-3-031-82035-9_7.
47. Python Documentation. URL: <https://docs.python.org/3/>
48. Fryz M., Mlynko B. Property analysis of multivariate conditional linear random processes in the problems of mathematical modelling of signals // *Technol. Audit Prod. Reserv.* 2022. Vol. 3, No 2(65). P. 29–32.
49. Бабак В.П., Марченко Б.Г., Фриз М.Є. Теорія ймовірностей, випадкові процеси та математична статистика. – К.: Техніка, 2004. – 288 с.
50. M. Fryz, “Conditional linear random process and random coefficient autoregressive model for EEG analysis,” 2017. doi:10.1109/UKRCON.2017.8100498.
51. Голобородько В. М. Вибрані глави проєктивної ергономіки. Антропометричний фактор: навчальний посібник. Харків: ХДАДМ, 2004. 216 с.
52. ДСТУ 8604:2015. Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидючи. Загальні ергономічні вимоги. Київ, 2016. 7 с.
53. Вимоги щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями: НПАОП 0.00-7.15-18. Наказ Міністерства соціальної політики України від 14.02.2018 № 207.
54. Лапін В. М. Безпека життєдіяльності людини: навч. посіб. 6-те вид., перероб. і доп. Київ: Знання, 2007. 332 с.
55. Закон України «Про охорону праці». URL: <https://zakon.rada.gov.ua/laws/show/2694-12#Text>

ДОДАТКИ

Публікація на конференцію Proceedings of X International Scientific and Practical Conference Chicago, USA

УДК 004.62

ВИКОРИСТАННЯ JSON-LD ДЛЯ СЕМАНТИЧНОГО ПОДАННЯ ТА ІНТЕГРАЦІЇ ЗНАТЬ

Гашиньський Роман Ігорович,
здобувач другого (магістерського) рівня вищої освіти
Тернопільський національний технічний
університет імені Івана Пулюя
м. Тернопіль, Україна

Анотація: У роботі розглянуто використання JSON-LD (JavaScript Object Notation for Linked Data) для семантичного подання та інтеграції знань. Охарактеризовано його основні елементи: контекст, унікальні ідентифікатори та типи сутностей. Показано переваги JSON-LD у роботі з гетерогенними даними, побудові єдиної моделі знань і виявленні зв'язків між сутностями.

Ключові слова: JSON-LD, семантичне подання, інтеграція знань, гетерогенні дані, граф знань, модель знань.

У сучасних інформаційних системах часто виникає потреба працювати не лише з окремими наборами даних, а також із значними масивами взаємопов'язаної інформації. Дані можуть надходити з різних непов'язаних між собою джерел, мати різні формати, структуру та рівень деталізації [1]. Наприклад, одна система може зберігати інформацію про користувачів у вигляді таблиць, інша – у JSON-документах, а третя – у вигляді неструктурованого тексту. У такому випадку важливо не просто об'єднати дані, а й зберегти їхній зміст, контекст та зв'язки між сутностями.

Складним аспектом є робота з гетерогенними даними, тобто такими, що описують різнотипні об'єкти. До них можуть належати люди, організації, документи, події, процеси, ресурси, продукти або інші сутності певної предметної області. Такі сутності мають різну природу, різні властивості та різні типи зв'язків між собою [2]. Тому для їх ефективного опрацювання

потрібні підходи, які дозволяють зберігати не лише самі дані, а й їхній змістовий контекст. Одним із способів реалізації цього є JSON-LD. Він поєднує зручність звичайного JSON із можливостями семантичного опису даних [3]. Звичайний JSON добре підходить для збереження структурованої інформації, однак він не завжди пояснює, що саме означають ті чи інші поля. Наприклад, поле *"name"* може означати ім'я людини, назву документа, назву організації або назву події. JSON-LD дозволяє конкретизувати значення таких полів через спеціальний контекст [4], завдяки чому дані стають простішими для сприйняття не лише для людини, а й для комп'ютерної системи (табл. 1).

Таблиця 1

Порівняння JSON та JSON-LD

Критерій	JSON	JSON-LD
Основне призначення	Передача і збереження структурованих даних	Семантичне подання та зв'язування даних
Опис значення полів	Обмежений	Через <i>@context</i>
Підтримка унікальних ідентифікаторів	Не обов'язкова	Реалізовано через <i>@id</i>
Опис типу сутності	Зазвичай неформальний	Реалізовано через <i>@type</i>
Подання зв'язків між об'єктами	Можливе, але без чіткої семантики	Підтримується як частина Linked Data
Придатність для графів знань	Обмежена	Висока

Основними елементами JSON-LD є *@context*, *@id* і *@type* [5]. Компонент *@context* вказує, як потрібно інтерпретувати поля документа; *@id* визначає унікальний ідентифікатор сутності для уникнення дублювання об'єктів; *@type* задає тип сутності, наприклад: "студент", "документ", "організація", "подія" або "етап процесу". Завдяки цьому кожен об'єкт у системі має не лише набір характеристик, а й чітке місце в загальній моделі знань.

Семантичне подання даних за допомогою JSON-LD є доцільним у процесі інтеграції знань. Якщо дані надходять із різних джерел, вони можуть описувати одну й ту саму сутність по-різному. Наприклад, в одному джерелі може бути використана назва *"student"*, в іншому – *"learner"*, а в третьому – *"person"*. Без

семантичного узгодження система сприйматиме ці записи як різні об'єкти. JSON-LD дозволяє пов'язати такі описи з єдиним змістовим поняттям і таким чином зменшити ризик дублювання або помилкової інтерпретації.

У межах графової моделі знань JSON-LD дозволяє представляти дані у вигляді взаємопов'язаних сутностей. У такій моделі об'єкти можна розглядати як вершини графа, а зв'язки між ними – як ребра. Наприклад, у навчальній предметній області можна подати такі зв'язки: студент пише наукову роботу, керівник перевіряє роботу. Такий підхід дозволяє бачити не лише окремі об'єкти, а й повну систему їхніх взаємозв'язків (рис. 1).

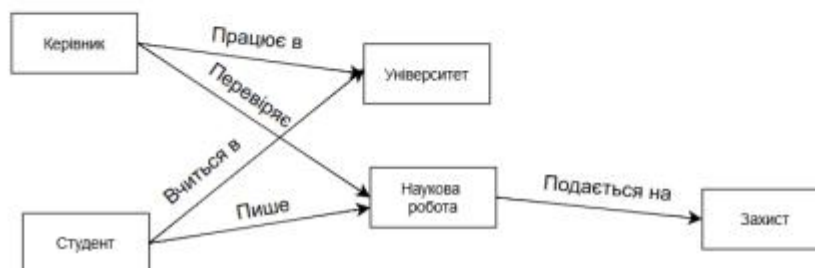


Рис. 1. Приклад зв'язків між гетерогенними сутностями

Перевага такого підходу полягає в тому, що після інтеграції даних можна виявляти не лише прямі, а й непрямі зв'язки. Наприклад, якщо студент пов'язаний із науковою роботою, а наукова робота пов'язана з керівником, то можна встановити зв'язок між студентом і керівником. З огляду на це, JSON-LD стає не просто форматом збереження даних, а й основою для подальшого аналізу знань.

Важливо також, що JSON-LD підтримує поступове розширення моделі. Якщо в систему потрібно додати нові типи сутностей або нові зв'язки, це можна зробити без повної перебудови всієї структури. Наприклад, до вже наявної моделі можна додати такі об'єкти, як "рецензент", "коментар", "версія документа", "дата подання" або "результат захисту". Це робить JSON-LD зручним для динамічних інформаційних систем, у яких дані постійно

змінюються та доповнюються.

Отже, JSON-LD є ефективним засобом семантичного подання та інтеграції знань. Його використання дозволяє формально описувати сутності, задавати їхні типи, унікальні ідентифікатори, властивості та зв'язки. Особливо важливим цей формат є для роботи з гетерогенними даними, оскільки він допомагає узгоджувати різні джерела інформації та формувати єдину модель знань. Завдяки цьому JSON-LD можна використовувати як основу для побудови графових структур, виявлення прихованих зв'язків між сутностями та подальшого інтелектуального аналізу даних.

СПИСОК ЛІТЕРАТУРИ

1. Rozony, Farhana Zaman, et al. "A systematic review of big data integration challenges and solutions for heterogeneous data sources." *Academic Journal on Business Administration, Innovation & Sustainability* 4.04 (2024): P. 1-18.
2. Borowicc, Silvia Lucia, and Solange Nice Alves-Souza. "Heterogeneous Data Integration: A Literature Scope Review." *ICEIS (I)*. 2024. P. 189-200.
3. Sivaranjani, S., V. Tharunika, and V. Bhuvanewari. "Semantic Web for Addressing Data Integration Challenges Semantic Data Fabric for Healthcare." *Advances in Healthcare Using Machine Learning*. CRC Press, 2025. P. 240-267.
4. Awan, Mahrukh, et al. "Towards Efficient Structured Description Generation for Data Marketplace Offerings." *2025 IEEE Conference on Standards for Communications and Networking (CSCN)*. IEEE, 2025. P. 1-7
5. Spomy, Manu, et al. "JSON-LD 1.0." *W3C recommendation* 16 (2014): 41.

Публікація на конференцію 2nd International Workshop on Advanced Applied Information Technologies (AdvAIT-2025)

AI-driven multimodal data fusion for hazardous object detection in maritime and coastal environments*

Vitaly Brevus^{1,*,†}, Halyna Brevus^{1,†}, Roman Gashynskiy^{1,†}, Aser Kashosi^{2,†} and, Roman Yuzefovych³

¹ Ternopil Ivan Puluj National Technical University, 56 Ruska str., Ternopil 46001, Ukraine

² Rework-Space LLC, 10 Berezhanska str, office 82, Ternopil 46027, Ukraine

³ Karpenko Physico-Mechanical Institute NAS of Ukraine, Department of Methods and Facilities for Acquisition and Processing of Diagnostic Signals, 5 Naukova Str., Lviv 79060, Ukraine

Abstract

This paper introduces a novel AI-driven platform that integrates satellite imagery, unmanned aerial vehicle data, and Automatic Identification System signals into an analytics pipeline for hazardous object detection in maritime and coastal environments. The system leverages YOLO11 for object detection and a knowledge graph based on Blue Brain Nexus to achieve semantic interoperability. Results demonstrate the ability of the developed information technology to detect maritime debris, oil spills, and vessel activity, while enabling adaptive route planning and decision support. This approach provides a scalable framework for emergency response and environmental monitoring, aligning with current advances in Artificial Intelligence, machine learning, and applied modeling in information technologies.

Keywords

AI, multimodal data fusion, knowledge graph, satellite imagery, emergency response

1. Introduction

The frequency of natural disasters, maritime accidents, and climate-related hazards has increased significantly in recent years [1]. Traditional monitoring systems that rely solely on satellite data or unmanned aerial vehicle (UAV)-based inspections often fail to provide a complete and timely situational picture [2, 3]. There is a growing need for integrated approaches that combine multiple heterogeneous data sources with Artificial Intelligence (AI) to support emergency response [4, 5]. This is particularly relevant in the context of digital sovereignty and cloud infrastructure projects like Gaia-X [6]. For near-real-time hazardous object detection, and actionable insights for safer maritime operations within coastal areas or large rivers one needs the solution, merging satellite data with accurate drone imagery and heterogeneous data sources enriching the precision and accuracy of the system.

This paper presents a multimodal AI platform that merges satellite imagery, UAV data, and Automatic Identification System (AIS) signals into a unified knowledge graph, enabling near-real-time hazardous object detection and decision support. The goal of this paper is to demonstrate the feasibility of multimodal fusion for maritime monitoring, present a knowledge-driven architecture for semantic interoperability, and validate the system.

*AdvAIT-2025: 2nd International Workshop on Advanced Applied Information Technologies: AI & DSS, December 05, 2025, Khmelnytskyi, Ukraine, Zilina, Slovakia

† Corresponding author.

‡ These authors contributed equally.

✉ v_brevus@ntnu.edu.ua (V. Brevus); ceo@rework-space.com (H. Brevus); rgashynskiy@rework-space.com (R. Gashynskiy); akashosi@rework-space.com (A. Kashosi); roman.yuzefovych@ipm.lviv.ua (R. Yuzefovych)

© 0000-0002-7035-9905 (V. Brevus); 0009-0009-5614-0241 (H. Brevus); 0009-0002-1681-4378 (R. Gashynskiy); 0000-0002-9059-2358 (A. Kashosi); 0000-0001-5546-453X (R. Yuzefovych)

 © 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

2. Related works

Research on AI for remote sensing has expanded rapidly [3], focusing on applications such as vessel detection, oil spill monitoring, and disaster assessment, based on developments of neural networks for image analysis [7, 8, 9]. Multimodal data fusion has been studied extensively [10], with approaches ranging from simple statistical techniques to advanced semantic integration. Knowledge graphs (e.g., Blue Brain Nexus) have emerged as effective tools for organizing heterogeneous data while ensuring interoperability [11]. Despite progress, existing works rarely integrate near-real-time AI models, multimodal fusion, and knowledge graphs into a single system for maritime safety and emergency response. Integration of cutting-edge object detection algorithms, such as YOLO, becomes critical for enhancing near-real-time situational awareness and rapid response in maritime environments. The YOLO object detection algorithm [9, 12] is used for visual recognition of quadcopter streaming videos. Cargo ships can use drones for monitoring dangerous sea routes. Having such drones in air and sea all the time would be too expensive and ineffective. Our platform combines available satellite and tracking data for preliminary analysis and deploy sea drones or UAVs when needed, to receive local accurate data, transform these and analyze them all in real time.

Relevant data, offered by monitoring services, can be classified by source:

- Copernicus Marine Service (CMEMS): provides comprehensive sea surface temperature, salinity, and currents data, along with marine ecosystem data and sea ice concentration and extent measurements.
- European Marine Observation and Data Network (EMODnet): offers bathymetry, seabed habitats, and human activities information, complemented by oceanographic data including tides and currents, as well as geological, biological, and chemical datasets.
- National Oceanic and Atmospheric Administration (NOAA): contributes ocean temperatures, salinity, and currents data, marine debris monitoring and habitat mapping services, and weather and sea surface data collected from buoys and drones.
- Ocean Observatories Initiative (OOI): delivers ocean temperature, salinity, and chemical properties data, seafloor imaging and topography information, and biological observations.

To collect necessary data, MariNeXt service [13] can be utilized to classify critical maritime hazards like oil spills and debris. Its robust predictions ensure effective hazard monitoring for maritime surveillance, even in challenging environments.

3. Methodology

This study presents the development and validation of a UAV-based object detection system for maritime surveillance applications. While the broader multimodal data fusion framework encompasses satellite imagery (Sentinel-1 SAR, Sentinel-2), Automatic Identification System (AIS) data, and knowledge graph integration, this paper specifically focuses on the UAV component utilizing YOLOv11 object detection algorithms.

3.1. System architecture overview

The proposed information technology implements a five-component modular pipeline architecture (Figure 1). The data ingestion layer aggregates heterogeneous sources including Copernicus Sentinel Hub API, UAV-mounted sensors, and AIS data streams. Data preprocessing is performed through the QueryOptima™ platform for harmonization and normalization. A Blue Brain Nexus-powered knowledge graph manages semantic data storage and interoperability. The analytics layer integrates YOLOv11 object detection with graph-based reasoning, while the visualization component provides real-time dashboards and geospatial interfaces for end-user interaction.

3.2. UAV-based object detection system

A Minimum Viable Product (MVP) of the system was developed and tested during the Case Study: 8th CASSINI Hackathon (November 2024) at the POLE Product Design Center, Ukraine.

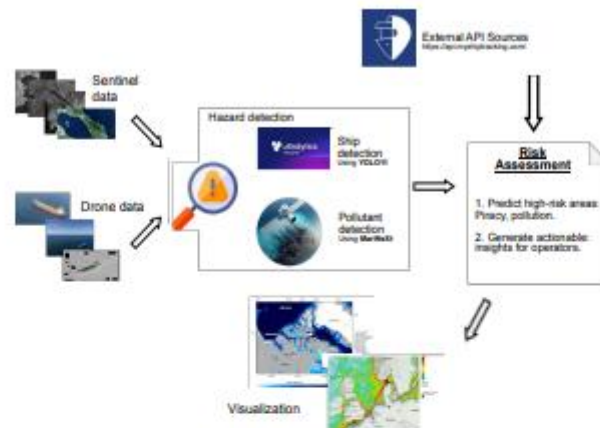


Figure 1: Diagram of AI-driven multimodal data fusion and risk assessment process.

The core contribution of this work centers on the development of a specialized UAV-based maritime object detection system employing the YOLOv11 architecture. The detection model was trained exclusively on the High-Resolution Ship Collections 2016 Multi-Scale (HRSC2016-MS) maritime dataset [14], representing a significant enhancement over the original HRSC2016 dataset.

Dataset Characteristics and Preprocessing. The HRSC2016-MS dataset comprises 1,680 high-resolution optical remote sensing images containing 7,655 annotated ship instances. The dataset exhibits comprehensive environmental diversity, encompassing maritime scenes across multiple operational conditions: sea and coastal environments, diurnal and nocturnal imaging scenarios, and varied meteorological conditions including clear and cloudy weather patterns. The dataset's multi-scale nature provides images with varying resolutions and aspect ratios, essential for training robust detection algorithms capable of identifying vessels across different scales and perspectives.

YOLOv11 Model Training and Fine-tuning. The YOLOv11 object detection architecture was selected for its demonstrated superiority in real-time object detection tasks and computational efficiency suitable for UAV deployment scenarios. The model was trained end-to-end on the HRSC2016-MS dataset, which was split into 60% for training, 20% for testing and 20% for validation, using transfer learning from pre-trained weights, with specific hyperparameter optimization for maritime object detection. Fine-tuning procedures incorporated UAV-specific imagery to enhance detection performance in coastal operational environments. The training process employed data augmentation techniques including geometric transformations, photometric adjustments, and multi-scale training to improve model generalization across diverse maritime conditions. Model convergence was monitored through validation metrics including precision, recall, and mean Average Precision (mAP) at multiple Intersection over Union (IoU) thresholds.

3.3. Integration framework

While this paper focuses on the UAV detection component, the broader system integrates with semantic knowledge representation through Blue Brain Nexus, enabling cross-modal data fusion and reasoning capabilities. The knowledge graph architecture facilitates semantic interoperability between UAV detection results, satellite imagery analysis, and AIS tracking data, supporting comprehensive maritimesituational awareness and risk assessment applications.

4. Results

The system was validated through three primary use cases: (1) detection and classification of oil spills, debris, and environmental anomalies using MariNeXt; (2) ship detection using Sentinel-1 SAR imagery (trained on SAR-Ship-Dataset²) split into 60% for training, 20% for testing and 20% for validation that was cross-validated with UAV imagery; and (3) prediction of vessel routes with anomaly detection using AIS data.

The comparison highlights that the YOLO11 model, fine-tuned with UAV imagery, demonstrates reliable ship detection in coastal environments. While some false positives occur, the overall bounding box alignment with labeled data confirms high precision (Figure 2).

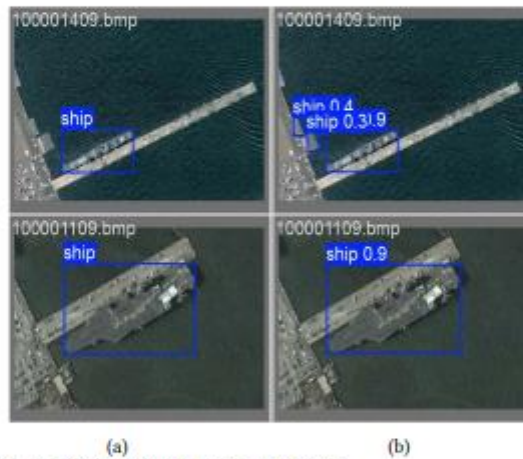


Figure 2: Comparison of labeled (a) and predicted (b) ships.

The Figure 3 illustrates the evolution of validation metrics during model training. The initial drop in precision is an artifact of the learning rate scheduler and optimizer state at the beginning of the training. Precision rapidly stabilizes around 0.9, while Recall converges near 0.75. The mean Average Precision at Intersection over Union threshold 0.5 reaches approximately 0.85, and the stricter metric mAP@[0.5:0.95] converges to 0.66. These results indicate consistent model improvement across epochs and confirm robust generalization for maritime object detection.

All losses decrease steadily, with the most significant reduction observed during the first 20 epochs, after which the curves gradually converge and stabilize. The absence of divergence between training

²<https://github.com/CAESAR-Radi/SAR-Ship-Dataset>

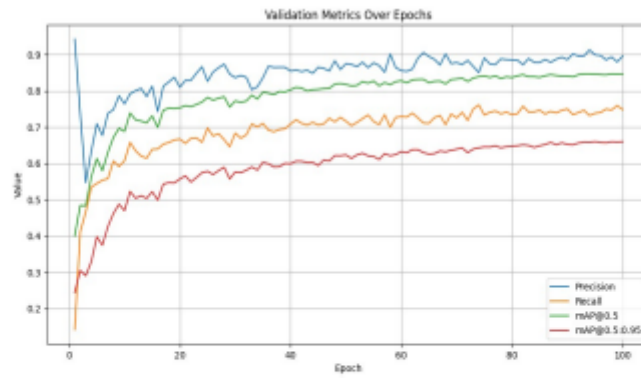


Figure 3: Validation metrics over training epochs.

and validation losses indicates that the model does not suffer from overfitting and generalizes well to unseen data (Figure 4).

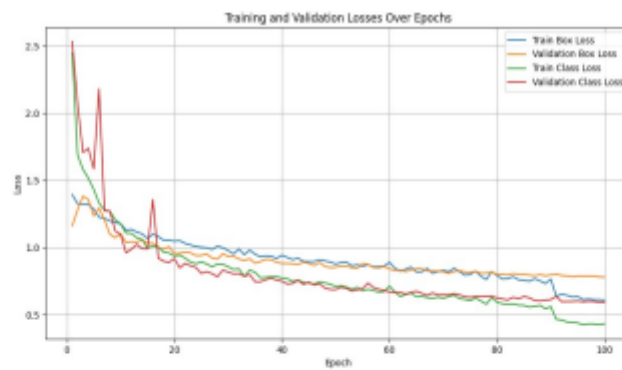


Figure 4: Training and validation losses over training epochs.

Therefore, the main results of the information technology validation can be summarized as follows:

- **Detection Accuracy:** The fine-tuned YOLO11 model achieved a precision of 90% and a recall of 75%, demonstrating robust performance in identifying maritime objects with UAV-enhanced data.
- **Processing Latency:** The analytics pipeline maintained an average processing time of under 5 seconds per frame, ensuring near-real-time performance suitable for time-critical applications.

- **Data Integration:** The platform successfully integrated multimodal data sources (satellite, UAV, and AIS) into a unified knowledge graph using Blue Brain Nexus, enabling effective semantic linking and querying.

The MVP was awarded third place at the national level of the 8th CASSINI Hackathon competition.

5. Discussion

The validation of the proposed information technology confirmed its strengths, including the provision of analytics with low latency, semantic interoperability through a knowledge graph, and a scalable pipeline applicable to multiple domains such as maritime, coastal, and disaster response. However, a few challenges were identified. These include limitations in data access, such as restricted AIS feeds and cloud dependency, the high computational demand of near-real-time pipelines, and the limited availability of labeled datasets for training YOLO11 on maritime hazards. Future work should focus on expanding the platform's capabilities beyond maritime safety to address other disaster types like floods and wildfires. Furthermore, important future developments include integration with European Data Spaces initiatives and ensuring compliance with FAIR data principles³ for open science.

6. Conclusion

The proposed information technology demonstrates the feasibility of an AI-driven multimodal data fusion system for maritime safety and environmental monitoring. By combining satellite, UAV, and AIS data within a knowledge graph architecture, the system provides near-real-time hazardous object detection and decision support. The fine-tuned YOLO11 model achieved a precision of 90% and a recall of 75%, with a mAP@[0.5:0.95] of 0.66, demonstrating robust performance. Data owners will benefit from the developed information technology by maximizing the value of their data assets to provide high-quality insights for maritime security.

The ship detection model from Sentinel-1 SAR data performs exceptionally well achieving 93% precision and 92.5% recall. This demonstrates high accuracy in identifying ships with minimal false positives, critical for cross validation with AIS third party API data.

The results from the CASSINI Hackathon validate its effectiveness and highlight the potential of such approaches in broader applications, including emergency response, sustainable maritime operations, and environmental protection.

Acknowledgements

This work is based on the DroneSight project, which was awarded third place in the Ukrainian national selection of the 8th CASSINI Hackathon: EU Space for Defence and Security⁴. The authors gratefully acknowledge the CASSINI Hackathon, an initiative of the European Union, for providing the framework and premium access to satellite imagery that were instrumental to this research. Rework-Space LLC acknowledges support from the European Innovation Council through the Seeds of Bravery (UASEEDs) project.

Declaration on Generative AI

During the preparation of this work, the authors used GPT-4 and Grammarly in order to: Grammar and spelling check. After using these tools/services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

³<https://www.go-fair.org/fair-principles/>

⁴<https://talkai.network/cassinihackathons/hackathons/euspace-defence-security/>

References

- [1] R. Khurana, D. Mugabe, X. L. Etienne, Climate change, natural disasters, and institutional integrity, *World Development* 157 (2022). doi:10.1016/j.worlddev.2022.105931.
- [2] I. Chandran, K. Vipin, Multi-UAV networks for disaster monitoring: challenges and opportunities from a network perspective, *Drone Systems and Applications* 12 (2024) 1–28. doi:10.1139/dsa-2023-0079.
- [3] J. Wang, K. Zhou, W. Xing, H. Li, Z. Yang, Applications, evolutions, and challenges of drones in maritime transport, *Journal of Marine Science and Engineering* 11 (2023). doi:10.3390/jmse11112056.
- [4] S. P. H. Boroujeni, A. Razi, S. Khoshdel, F. Afghah, J. L. Coen, L. O'Neill, P. Fule, A. Watts, N.-M. T. Kokolakis, K. G. Vamvoudakis, A comprehensive survey of research towards AI-enabled unmanned aerial systems in pre-, active-, and post-wildfire management, *Information Fusion* 108 (2024). doi:10.1016/j.inffus.2024.102369.
- [5] A. Bazrafkan, C. Igathinathane, N. Bandillo, P. Flores, Optimizing integration techniques for UAS and satellite image data in precision agriculture – a review, *Frontiers in Remote Sensing* 6 (2025). doi:10.3389/frsen.2025.1622884.
- [6] R. Adler-Nissen, K. A. Eggeling, The discursive struggle for digital sovereignty: Security, economy, rights and the cloud project Gaia-X, *JCMS: Journal of Common Market Studies* 62 (2024) 993–1011. doi:10.1111/jcms.13594.
- [7] I. Konovalenko, P. Maruschak, V. Brevus, Steel surface defect detection using an ensemble of deep residual neural networks, *Journal of Computing and Information Science in Engineering* 22 (2021). doi:10.1115/1.4051435.
- [8] I. Konovalenko, P. Maruschak, V. Brevus, O. Prentkovskis, Recognition of scratches and abrasions on metal surfaces using a classifier based on a convolutional neural network, *Metals* 11 (2021). doi:10.3390/met11040549.
- [9] L.-h. He, Y.-z. Zhou, L. Liu, W. Cao, J.-h. Ma, Research on object detection and recognition in remote sensing images based on YOLOv11, *Scientific Reports* 15 (2025). doi:10.1038/s41598-025-96314-x.
- [10] Y. Li, M. El Habib Daho, P.-H. Conze, R. Zeglache, H. Le Boité, R. Tadayoni, B. Cochener, M. Lamard, G. Quellec, A review of deep learning-based information fusion techniques for multimodal medical image classification, *Computers in Biology and Medicine* 177 (2024). doi:10.1016/j.combiomed.2024.108635.
- [11] M. F. Sy, B. Roman, S. Kerrien, D. M. Mendez, H. Genet, W. Wajerowicz, M. Dupont, I. Lavriushev, J. Machon, K. Pirman, D. Neela Mana, N. Stafeeva, A.-K. Kaufmann, H. Lu, J. Lurie, P.-A. Fonta, A. G. R. Martinez, A. D. Ulbrich, C. Lindqvist, S. Jimenez, D. Rotenberg, H. Markram, S. L. Hill, Blue brain nexus: An open, secure, scalable system for knowledge graph management and data-driven science, *Semantic Web* 14 (2023) 697–727. doi:10.3233/SW-222974.
- [12] B. Zhao, J. Zhao, R. Song, L. Yu, X. Zhang, J. Liu, Enhanced yolo11 for lightweight and accurate drone-based maritime search and rescue object detection, *PLOS ONE* 20 (2025) 1–24. doi:10.1371/journal.pone.0321920.
- [13] K. Kikaki, I. Kakogeorgiou, I. Hoteit, K. Karantzas, Detecting marine pollutants and sea surface features with deep learning in Sentinel-2 imagery, *ISPRS Journal of Photogrammetry and Remote Sensing* 210 (2024) 39–54. doi:10.1016/j.isprsjprs.2024.02.017.
- [14] W. Chen, B. Han, Z. Yang, X. Gao, Mssdet: Multi-scale ship-detection framework in optical remote-sensing images and new benchmark, *Remote Sensing* 14 (2022). URL: <https://www.mdpi.com/2072-4292/14/21/5460>. doi:10.3390/rs14215460.