Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет навчально-науковий центр заочної форми навчання
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

АТЕСТАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Модель автоматизованої обробки даних в IoT

(тема)

Виконав:
студент II курсу, групи СПзм-18-2
Соловйова А.С.
(прізвище, ініціали)

Спеціальність
123 – Комп'ютерна інженерія
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма
Системне програмування
(повна назва освітньої програми)

Керівник: проф. Міхаль О.П.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ _____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет навчально-науковий центр заочної форми навчання

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 – Комп'ютерна інженерія
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
"____" _____ 20___ р.

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ

студентові Соловйовій Аліні Сергіївні
(прізвище, ім'я, по батькові)

1. Тема роботи Модель автоматизованої обробки даних в IoT

затверджена наказом по університету від "30" березня 2020 р. № 43 Стз

2. Термін подання студентом роботи до екзаменаційної комісії 18 травня 2020 р.

3. Вхідні дані до роботи

IoT

геоконтекстні дані

інформаційна система

4. Перелік питань, що потрібно опрацювати в роботі

Обробка геоконтекстних даних в IOT

Модель обробки геоконтекстних даних в інтернеті речей

Архітектура інформаційної системи

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 14 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Огляд літератури за темою	31.03.20-12.04.20	
	дослідження		
2	Вибір та обгрунтування методики	13.04.20-20.04.20	
	дослідження		
3	Вибір інструментальних засобів	21.04.20-27.04.20	
4	Проведення експериментів	28.04.20-05.05.20	
	Оформлення ПЗ	06.05.20-11.05.20	
6	Подання атестаційної роботи	12.05.20-13.05.20	

Дата видачі завдання 30 березня 2020 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Міхаль О.П.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка атестаційної роботи: 77 с., 18 рис., 3 табл., 1 дод., 23 джерел.

ІОТ, ІНТЕРНЕТ РЕЧЕЙ, ГЕОКОНТЕКСТНІ ДАНІ, АНАЛІЗ, ОБРОБКА ДАНИХ, ІНФОРМАЦІЙНА СИСТЕМА.

Метою атестаційної роботи є аналіз моделей та методів автоматизованої обробки даних в IoT.

Наведено формальний опис моделі даних для виконання геоконтекстной розмітки даних IoT. Описано основні сутності і структури даних. Визначено операції на даних множини. Наведені приклади використання різних видів операцій для основних сценаріїв роботи з геоконтекстнимі даними IoT. Описана особливість моделі - гнучкість як щодо формату самих джерел, так і щодо структури геоконтекстних даних. Показано, що введена модель задовольняє вимогам поставленого завдання на прикладі аналізу її операцій і їх зіставлення з вимогами IoT.

Описано метод автоматизованої обробки геоконтекстних даних IoT. Описано основні етапи обробки – перетворення даних, індивідуальна геоконтекстная розмітка, групова геоконтекстная розмітка.

ABSTRACT

Master’s thesis: 77 pages, 18 figures, 3 tables, 1 appendics, 23 sources.

IOT, INTERNET OF THINGS, GEOCONTEXT DATA, ANALYSIS, DATA PROCESSING, INFORMATION SYSTEM.

The major goal of this thesis is nalysis of models and methods of automated data processing in IoT.

A formal description of the model of the Danish for the end of the geocontext rozmitki Dano IoT. The basic daily and structure of the Danish are described. Visited operations on these many. Directly put the victorious views of the operation for the main scenarios of work with geo-contextual data IoT. The specificity of the model is described – the nullity is typical of the format of the dzherel itself, so it is the structure of geocontextual data. It is shown that a model for the content of the vimogs of the delivered assignment for the application analysis of the operation and installation of the ioT vimogs was introduced. The method of automated processing of geo-contextual data IoT is described. The main stages of processing are described - rebuilding, individual, geo-contextual rozmitka, group geo-contextual rozmika.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ГІС – геоінформаційна система

ІР – Інтернет речей

ІС – інформаційна система

ВСТУП

Інтернет Речей (IP) – концепція міжмашинної взаємодії, вперше сформульована Кевіном Ештоном в 1999 році [19]. Суть концепції полягає в тому, що всі електронні пристрої («речі») об’єднуються в обчислювальну мережу завдяки вбудованим технологіям для передачі даних. При цьому окремі пристрої можуть як генерувати власні дані, так і споживати (перетворювати) дані від інших пристроїв с допомогою відповідних функцій апаратного і програмного забезпечення.

На сьогоднішній день концепція IP пройшла шлях від прототипів до широко використовуваної технології. Згідно з дослідженням, проведеним компанією Gratner, Inc., кількість використовуваних пристроїв IP в 2020 році складе 11 мільйонів, а в 2021 році цей показник буде подвоєний. При цьому, використання даних пристроїв і програмних рішень на їх основі зачіпає досить багато галузей. В аналітичному дослідженні компанії IoT Analytics найбільш важливими відзначені наступні застосування IB: екосистема розумного міста, програмні системи для індустріальних виробничих систем, будівництво, автомобільні інформаційні системи, енергоспоживання, охорону здоров’я, розумні будинки.

Згідно з рекомендацією Міжнародного союзу електрозв’язку концепція IB архітектурно представима за допомогою наступного набору сутностей:

- фізичні і віртуальні речі, що представляють собою об’єкти фізичного і віртуального світу відповідно, для яких можливо проводити процедуру вимірювання будь-яких параметрів за допомогою сенсорів;

- пристрої – елементи обладнання, що володіють можливостями зв’язку, введення, передачі і обробки даних, а також управління фізичними речами;

- мережі зв’язку забезпечують середу для взаємодії пристроїв;

- додатки являють собою програмні системи, що займається агрегуванням, подальшою обробкою та інтерпретацією даних пристроїв.

Рекомендація наводить такі характеристики програмних систем IP:

- Велика кількість речей, які можуть бути підключені до ІВ.
- ІВ надає сервіси, що реалізують взаємодію з речами в рамках обмежень (контроль доступу, узгодженість).
- Гетерогенність апаратних платформ, мереж передачі даних, протоколів і форматів даних.
- Динамічні зміни станів (доступності) пристроїв.
- Великий обсяг обміну даними.

Метою атестаційної роботи є аналіз моделей та методів автоматизованої обробки даних в IoT.

Об’єкт дослідження: програмні системи IoT, які використовують геоконтекстні дані.

Завдання:

- аналіз використання геоконтекстних даних в програмних системах IoT;
- розробка моделі та обробки геоконтекстних даних в програмних системах IoT;
- розробка архітектури IoT обробки геоконтекстних даних для програмних систем.

1 ОБРОБКА ГЕОКОНТЕКСТНИХ ДАНИХ В ІОТ

1.1 Моделі даних Інтернету Речей

Грунтуючись на характеристиці IP, а також на описах найбільш популярних сценаріїв використання можна зробити висновок про те, що в основі даних IP лежать прив’язані до часу вимірювання сенсорів, що формуються з високою швидкістю. Крім вимірювань, дані ІВ містять також метадані, що описують речі, пристрої, мережі та програмні системи. При цьому формат представлення даних та їх якісний склад в загальному випадку є різним через різноманіття пристроїв. Тому можна вважати, що дані ІВ є слабоструктурированное [36]. З технічної точки зору дані ІВ також володіють великою різноманітністю форматів представлення. Найбільш популярними форматами є JSON, XML, CSV, RDF.

В якості моделей для даних ІВ використовується три формалізму, які можуть бути застосовані разом: реляційна модель, модель «сутність-атрибут-значення» (САЗ) [24, 45] і семантичні технології.

Реляційна модель (Рисунок 1.1) даних являє собою логічну модель даних, засновану на операціях реляційної алгебри. Дані структуруються у вигляді реляційних відносин (таблиць), структурними елементами яких є кортежі (рядки), що містять фіксований набір значень атрибутів (стовпців). Набір атрибутів реляційного відносини приймається незмінним до з точки зору назв, так і з точки зору типів значень. Над реляційними відносини визначаються операції, що дозволяють здійснювати теоретико-множинні перетворення над кортежами, пов’язувати реляційні відносини між собою на підставі значень атрибутів. Розширена реляційна модель реалізується в класі інструментів SQL БД і застосовується в якості інструменту для моделювання даних IP.

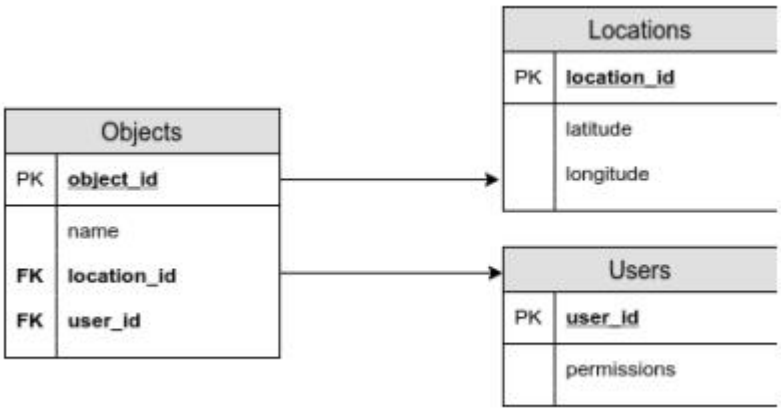

Рисунок 1.1 – Реляційна модель даних

Базові Поняття моделі САЗ (див Малюнок 1.2) є Сутність (описуванний предмет або явище в широкому сенсі), атрибут (характеристика суті) і значення атрибута. Дані сутності структуруються у вигляді довільних наборів пар атрибут-значення. Таким чином, дані в рамках моделі САЗ утворюють аналог розрідженої матриці. При цьому, на відміну від реляційної моделі, САЗ не визначає будь-яких обмежень для атрибутів, тому її також називають моделлю з гнучкою схемою. Як операції в САЗ реалізуються аналоги операцій реляційної алгебри, за винятком операції скріплення двох відносин. Модель САЗ використовується в якості основи для наступних класів інструментів: СКБД «ключ-значення», документо- орієнтовані СКБД, СКБД для зберігання тимчасових даних, в ряду яких існують інструменти для обробки даних IP.

Рисунок 1.2 – Модель даних САЗ

Семантичні технології моделювання даних мають на увазі використання інструментів семантичного вебу [27] для опису структури даних і допустимих операцій. Як інструмент структурування даних виступає Resource Description Framework (RDF). Дана модель структурування даних використовує в якості базового елементу трійки

<Суб'єкт, предикат, об'єкт>, де об'єктами і суб'єктами в загальному випадку являють собою якийсь ресурс. Приклад моделі наведено на рисунку 1.3. Предикат є довільне відношення між об'єктом і суб'єктом. Модель дозволяє визначати власні типи даних за допомогою стандарту RDFS. Тріпл моделі утворюють граф, в якому крім стандартних операцій CRUD, можливі також операції логічного висновку. Інструменти на основі RDF широко використовуються для зберігання даних:

MarkLogic, Jena, Virtuoso. Використання семантичних технологій для докладно висвітлено в дослідженнях.

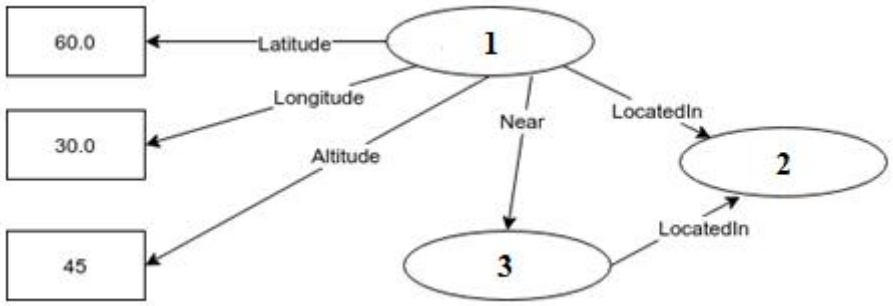

Рисунок 1.3 – Семантична модель даних

Підводячи підсумок, можна сказати, що спільною рисою моделей даних IP є підвищення гнучкості опису предметної області, що досягається за рахунок розширень (реляційна модель), структури моделі (модель EAV і семантичні технології моделювання). Дана вимога є основоважливою, так як кількість порушених IB предметних областей велике і при цьому використовуючі дані є слабкоструктуровані. В таких умовах не можна використовувати жорстку схему даних, оскільки набір атрибутів для об'єктів не є постійним.

1.2 Моделі даних геоінформаційних систем

1.2.1 Характеристика геоінформаційних систем

Термін географічні інформаційні системи (ГІС) вперше був згаданий в статті 1968 года. У сучасному розумінні ГІС представляють собою спеціалізовані інструменти для взаємодії з просторовими даними. Згідно До завдань, які вирішуються ГІС [3] входить збір, зберігання, обробка, доступ, відображення і розповсюдження просторових-тимчасових даних. ГІС можуть бути класифіковані:

- за масштабом охоплюваній просторової області (глобальні, регіональні, субрегіональні),
- по предметній області (міські, недропользовательскіе, гірничо геологічні інформаційні системи),
- за специфікою вирішуваних завдань.

Завдання збору просторових-тимчасових даних в ГІС полягає в завантаженні даних з різних джерел і їх перетворення у внутрішні формати ГІС. Джерелами даних найчастіше служать різні карти місцевості [31] у векторному (оцифровані знімки, зібрані вручну GPS- треки) або растровому форматі (необроблені фотознімки, зроблені літальними, космічними і наземними апаратами). Процедура перетворення включає в себе етап векторизації, на якому в автоматизованому або автоматичному режимі на базі вихідних даних виділяються примітивні просторові об’єкти - точки, лінії і багатокутники, які відповідають різним об’єктам на знімках. Також, в процесі збору відбувається перетворення географічних систем координат вихідних даних до географічної системі координат ГІС. Важливо відзначити, що для різних предметних областей і типів ГІС використовуються різні системи координат і формати вихідних даних.

Завдання зберігання просторово-часових даних в ГІС має на увазі забезпечення роботи спеціалізованої СКБД, а також моделювання даних цієї

СКБД. При цьому, ГІС може як забезпечувати власне рішення СКБД, так і існуючі рішення по обробці просторово-часових даних. На сьогоднішній день подібні рішення можуть бути реалізовані в рамках практично будь-якої логічної моделі даних.

Завдання аналізу просторово-часових даних в рамках ГІС в загальному випадку розпадається на наступні підзадачі:

Топологічне моделювання, що відповідає на питання про перетин, включення і близькість різних просторових об’єктів;

Побудова маршрутів і операції над графами з просторових об’єктів (дорожні мережі, інженерні комунікації);

Пряме і зворотне геокодування - перетворення географічних координат в адреси і навпаки, адрес в координати;

Географічна статистика, що включає в себе завдання інтерполяції і прогнозування.

Поєднання шарів карт, що дозволяє отримати нову інформацію про об’єкти, розташовані в межах однієї області.

Завдання доступу і поширення полягає в забезпеченні доступу до даних ГІС для програмних систем. Дане завдання вирішується допомогою використання спеціальних модулів і бібліотек на різних мовах програмування, інкапсулірувана взаємодія з даної ГІС. Також, частина ГІС надає інтерфейси для доступу через мережі передачі даних.

Завдання відображення полягає в візуалізації просторово-часових даних ГІС у вигляді карт. При цьому, карти можуть як відображати реальний рельєф земної поверхні, так і містити додаткову інформацію про об’єкти, що зберігаються в ГІС.

Таким чином, ГІС представляють собою спеціалізовані інструменти для виконання комплексу різних завдань в рамках конкретних предметних областей. Стосовно до програмних систем ІВ кожна з цих завдань є актуальною, а їх рішення ускладнюється специфікою різнорідних даних ІВ.

1.2.2 Моделі даних ГІС

Не дивлячись на спільність базових понять і низькорівневість завдань, що вирішуються ГІС, конкретні інструменти можуть мати проблемами сумісності, так як через відмінності предметних областей додатки ГІС до самих систем пред’являються специфічні вимоги до функціональності, а також використовуються предметно-залежні терміни моделювання. Існує ряд підходів, спрямованих на подолання цього бар’єру, описаних в рамках статті.

Одним з ключових методів є стандартизація термінів моделювання. Розглянемо принципи моделювання даних в ГІС. Перш ніж описувати конкретні види моделей, необхідно охарактеризувати специфіку даних для ГІС.

В силу особливостей земної поверхні і складності географічних обчислень, розташування об’єктів можуть задаватися по-різному. Сучасні ГІС в якості основної системи координат підтримують одну з геодезичних систем координат. Геодезичні системи координат задають місце розташування об’єкта щодо поверхні референц-еліпсоіда у вигляді трьох чисел: широта, довгота і висоти. Найбільш популярною системою координат на сьогоднішній день є WGS-84.

У ряді випадків ГІС може знадобитися візуалізувати окремі ділянки земної поверхні на площині. Наприклад, в завданнях візуалізації карт і в завданнях імпорту знімків місцевості. У таких випадках використовуються методи отримання і взаємного перерахунку картографічних проекцій, що дозволяють проводити обчислення в місцевій тривимірній декартової системі координат, прив’язаній до однієї з точок використовуваної області.

З точки зору прикладних моделей кожна ГІС пропонує власні терміни для моделювання предметної області. У монографії [10] на основі аналізу існуючих рішень проведений аналіз моделей даних ГІС та дана їх принципова класифікація. В результаті, моделі даних сучасних ГІС можна розділити на дві групи: пошарові і об’єктно-орієнтовані.

В основі пошарових моделей лежить поділ об’єктів на шари за їх призначенням, типу або іншими ознаками, в контексті розв’язуваної задачі. При цьому, об’єкти, віднесені до одного шару представляють собою єдиний елемент даних. Пошарова організація даних дозволяє оперувати об’єктами, що входять до складу шарів як єдиним цілим. Організацією OpenGeoConsortium, що займається стандартизацією діяльності в області геопросторових сервісів, розроблений ряд документів, покликаних виробити загальний підхід до створення пошарових моделей і розумію базових понять ГІС. В рамках цього стандарту визначається головний принцип пошарового моделювання – просторово-часова інформація є практично повсюдної для будь-якого набору даних. Вводиться чотири колекції, що характеризують модель (рисунок 1.4)

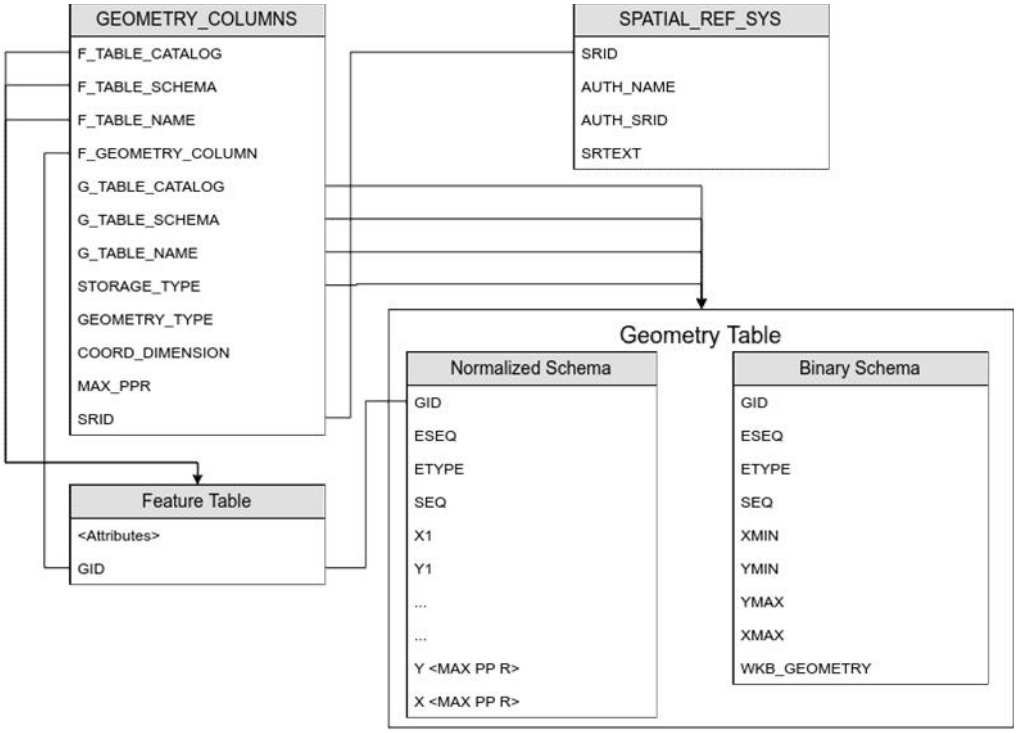

Рисунок 1.4 – Схема БД моделі OGC

Перераховані вище колекції визначають терміни для моделювання предметних областей, які сформульовані OGC як спеціалізованих мов розмітки просторових областей. Основною мовою моделювання є Geographic

Marckup Language (GML) [52], який формулює XML-граматику для вираження географічних об’єктів. GML виступає не тільки як мова моделювання, але і як формат обміну просторовими даними в мережі Інтернет. Завдяки використанню формату XML, частина моделі описується в вигляді спеціальної різновиди XML-схеми – GML-схеми, яка задає типи даних. Термінологія GML за рахунок своєї подробиці дозволяє описувати моделювати об’єкти практично будь-якої предметної області. З точки зору досліджуваної проблеми найважливішими елементами GML є [55] Особливість (Feature) – об’єкт фізичного світу, який не обов’язково має місце розташування. Геометрія (Geometry) – геометрична характеристика географічного об’єкта, що задається за допомогою координат. Система координат (Coordinate reference system), яка використовується для завдання геометрії. Топологія (Topology) географічних об’єктів, що задає властивості їх взаємного розташування. Тимчасові позначки (Time) об’єктів.

Правила відображення карт (Map presentation styling rules), що визначають способи візуалізації даних. Ключовим поняттям моделі OGC є геометрія. Вона може бути задана за допомогою наступних класів об’єктів: Точка (Point), що задає місце розташування об’єктів з однієї координатою. Крива (Curve), що задає місце розташування протяжного об’єкта у вигляді ламаної лінії. Поверхня (Surface), що задає опис просторових об’єктів у вигляді многокутників. Колекція геометрій (Geometry Collection), що задає місце розташування об’єкта у вигляді набору однотипних геометрій, визначених вище. Як інструмент розширення GML більш специфічними термінами конкретних предметних областей OGC пропонує два механізми: використання механізму призначених для користувача GML схем даних GML Application Schema і предметно-орієнтовані мови в рамках власних стандартів:

- CityGML,
- ARML2.0,
- GeoSciML,

- Geospatial eXtensible Access Control Markup Language (GeoXACML),
- GroundwaterML,
- IndoorGML,
- KML,
- LandInfra / InfraGML,
- WaterML.

З технічної точки зору, дана модель найчастіше втілюється за допомогою реляційних СУБД, тому в ряді джерел про неї говорять як про геореляціонной.

Пошарова модель дозволяє ефективно вирішувати завдання відображення карт, а також вести паралельну обробку даних. Однак, зі зростанням кількості збережених даних демонстрована продуктивність швидко зменшується. Крім цього, в рамках такої моделі важко використовувати складні опису геометрії для просторових об’єктів. Також недоліком є принципове обмеження на кількість класів об’єктів, яке частково долається за рахунок предметно-орієнтованих мов і додаткових схем даних.

Об’єктно-орієнтована модель (ООМ) [98] використовує інший принцип організації інформації. В рамках моделі відбувається оперування НЕ типами, а самими об’єктами. В рамках моделі вводяться такі колекції:

- просторові об’єкти (chart_obj),
- опис типів просторових наборів (chart_type),
- заголовки просторових наборів (chart_header),
- інформація про просторових наборах (chart_info),
- графічні об’єкти (graph_obj),
- типи атрибутів (type_attr),
- опис типів просторових об’єктів (obj_type),
- атрибути просторового об’єкта (obj_info),
- типи атрибутів просторових об’єктів (obj_info_type).

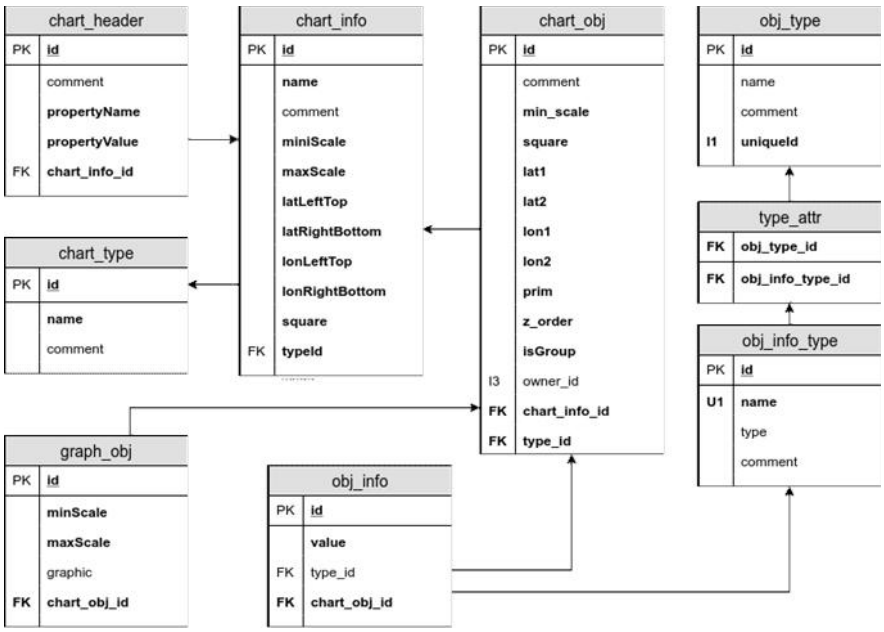

Рисунок 1.5 – Схема БД об’єктно-орієнтованої моделі

ООМ не має обмежень на кількість атрибутів і класів об’єктів. Це досягається тим, що, на відміну від OGC моделі, де атрибути і класи фіксуються моделлю досить чітко, в ООМ дані поняття мають досить загальну структуру, частково дублюючу модель EAV. Крім того, даний принцип застосовується і для геометрії – в моделі можна використовувати різні класи геометрії для одного і того ж об’єкта. Це призводить до одночасного підвищення продуктивності і високої надмірності даних – в порівнянні з моделями OGC експерименти над рішеннями з ООМ показали прискорення обробки даних на порядок [10].

Підводячи підсумок опису моделей даних ГІС можна зробити висновок про те, що найбільш перспективною з точки зору ІВ є OGC-модель, так як, на відміну від ООМ, вона реалізована і експлуатується в рамках безлічі технологічних рішень.

1.2.3 Оцінки швидкодії моделей ГІС

Розглянемо грубі оцінки часової складності для виконання типових операцій над елементами наборів геоконтекстних даних ІВ, які може

включати в себе процедура обробки. Під елементом набору будемо розуміти окрему сутність, що володіє довільним набором атрибутів, одним з яких є місце розташування. Як операцій будуть досліджені:

Додавання нового елемента в модель.

Вибірка елементів, які мають певну парою «атрібут- значення».

Розширення кількості атрибутів елемента в рамках моделі.

Вибірка елементів, чиє місце розташування знаходиться в рамках кола, визначеного радіусу.

Так як реалізації моделей OGC і OOM можуть базуватися на різних структурах даних, що реалізують просторові індекси, то позначимо складність вставки і пошуку елементів символами CI (n) і CS (n) відповідно в рамках використовуваного індексу. Середній обсяг набору даних позначимо як N, середня кількість атрибутів у елемента позначимо як M. При розгляді будемо вважати, що інформація про типи атрибутів і геометрій вже внесена в БД, а геометрії елементів мають однаковий тип.

1.3 Контекстно-залежні обчислення і геоконтекст

Поняття контекстно-залежних (Context-aware computing) обчислень було вперше сформульовано в 1994 співробітниками компанії Xerox. На сьогоднішній день під цим терміном мають на увазі концепцію створення програмних систем, які адаптують свою роботу під мінливі умови користувача і його оточення, які і є контекстом. Найбільш часто використовувана в літературі формулювання [44] визначає контекст як будь-яку інформацію, яка може бути використана для характеристики ситуації або сутності. Під сутністю мається на увазі будь-який суб’єкт взаємодії «користувач-програмна система», або зовнішні суб’єкти, пов’язані з ними безпосередньо. Найпростішими прикладами контекстних додатків є мобільні операційні системи, які підлаштовують якість відображення на екрані в залежності від зовнішніх умов (освітленість, заряд акумулятора).

Геоконтекстом будемо називати звуження поняття контексту до інформації про місцезнаходження учасників взаємодії «користувач-програмна система». За аналогією з контекстними обчисленнями можна визначити геоконтекстние обчислення, в завдання яких входить використання інформації про місцезнаходження даних про місцезнаходження. В англомовній літературі для їх позначення використовується термін «Location-aware computing». Приклади додатків, що реалізують парадигму геоконтекстних обчислень широко представлені на мобільних платформах в силу доступності точної інформації про місцезнаходження, одержуваної від датчиків супутникових навігаційних систем, даних мережі або даних від бездротових мереж передачі даних. До геоконтекстним додатків належать автомобільні навігатори, інтерактивні карти, ігри в доповненої реальності.

Геоконтекст може містяться в даних, що мають різну форму. У загальному випадку [67, 100] можна виділити наступні принципові форми представлення даних про місцезнаходження точкового об’єкта:

- координати (29.99, 60.00),
- адреси,
- назви географічних об’єктів,
- вказівки місця розташування.

При цьому, вихідні годинне не обов’язково структуруються у вигляді окремих точок. У загальному випадку можна вважати, що різні варіанти опису розташування можуть бути вичерпно описані в термінах геометрії OGC. Для вилучення геоконтекста з геоданих потрібно застосування спеціальних методів, які можна розділити на дві групи – статистична обробка та методи геокодування.

Статистичні методи аналізують множинні числові вимірювання. Приклад статистичних методів вилучення геоконтекста є фільтрація координат з метою зниження похибки і усереднення траєкторій транспортних засобів [16]).

Методи геокодування дозволяють проводити двосторонню перетворення географічних координат в адреси або назви географічних об’єктів. Процедура зіставлення координат словесними описами розташування називається прямим геокодування, а процедура отримання адреси або назви географічного об’єкта – зворотним геокодування.

На основі аналізу існуючих інструментів для роботи з контекстно-залежними обчисленнями і для розробки рішень IP були виділені наступні етапи життєвого циклу контексту в IP-системах:

- Отримання контексту від джерел даних
- Моделювання контексту в рамках термінів цільової системи.
- Витяг нових знань з існуючих контекстів.
- Поширення контексту, а також витягнутих з нього знань.

Відображаючи етапи життєвого циклу на завдання обробки геоданих в рамках програмних систем ІВ очевидно, що найбільш трудомісткими етапами є моделювання і витяг нових знань завдяки необхідності поєднання наборів геоданих і наборів даних IP. Трудомісткість складається з необхідності узгодження даних IP і ГІС для забезпечення вирішення завдань відображення і обробки даних. Додаткова трудомісткість виникає при одночасному використанні декількох різних ГІС зарахунок того, що через специфічність окремих ГІС-рішень це призводить до необхідності підтримки великого числа прикладних моделей. Тому, для застосування геоконтекстних операцій в рамках завдань ІВ необхідно вирішити задачу забезпечення єдиної моделі геоданих для уніфікації їх подання, а також дослідити способи інтеграції програмних систем Ір з ГІС рішеннями на технічному рівні, так як особливості конкретних ГІС знижують ефективність використання геоконтекстной розмітки.

1.4 Перспективи інтеграції моделей ГІС і IP

У попередніх розділах було проведено аналіз використовуваних моделей даних для IP і ГІС. Розглянемо, чи можна на базі існуючих

формалізацій реалізувати рішення по використанню геоданих в програмних системах ІР.

Найбільш перспективною моделлю ГІС є OGC-модель. Однак, для її використання з програмними системами ІР необхідно забезпечити більшу гнучкість термінів моделювання, так як дана концепція має на увазі роботу з великим спектром предметних областей, кожна з яких в свою чергу має на увазі свої терміни. Існуючі механізми розширення термінів моделювання для OGC-моделі не можуть вирішити дану проблему, так як розробка DSL або спеціальних схем даних істотно ускладнює процедуру моделювання даних, а також ускладнює підтримку програмних систем ІР. Крім, того важливим перешкодою на шляху використання OGC-моделі для програмних систем ІВ є гетерогенність використовуваних даних. Для ефективного використання моделі буде потрібно передбачити гнучкі процедури зв’язування геоданих і даних ІВ, так як, з огляду на обсяги даних, процедура повного перетворення формату може вимагати істотної кількості обчислювальних ресурсів.

Таким чином, для ефективного використання геоданих в програмних системах ІВ необхідно розробити модель даних, що базується на принципах OGC-моделі з точки зору обробки геоданих і враховує специфіку роботи програмних систем ІР. Крім цього, через високу фрагментації форматів і великої кількості різних джерел використання геоданих в ІР пов’язане із забезпеченням вирішення завдань зберігання і обробки даних, що вимагає вибору технологій обробки даних, що задовольняють специфіці ІР.

1.5 Аналіз концепцій обробки даних

Крім моделей даних, важливою властивістю системи інтеграції геоданих в рамках ІВ є концепція обробки даних – принцип організації інформаційної системи. Покажемо, що дані ІВ відповідають визначенню Великих Даних (Big Data). Згідно зі статтею [42] Великі Дані являють собою інформаційні активи, що характеризуються високими величинами обсягу, швидкості генерації і внутрішньої різноманітності, а також вимагають

використання спеціальних технологій і методів аналізу для обробки. Дані ІВ відповідають цьому визначенню так як Швидкість генерації і, як наслідок обсяг, даних ІВ є високими, так як одним з основних джерел є сенсори. Крім цього дані ІВ мають великою різноманітністю як внутрішньої структури, так і форматів представлення.

На сьогоднішній день існує два основних підходи до організації обробки Великих Даних - сховища даних (Data Warehouse) і озера даних (Data Lake). Розглянемо їх характеристики та відмінні риси стосовно до задачі обробки геоконтекстних даних в програмних системах ІР. При порівнянні будемо розглядати наступні критерії, що визначаються специфікою ІР:

- підтримка довільної схеми даних є важливим критерієм для даних ІВ через їх гетерогенності.
- наявність прив’язки до часу є необхідною вимогою для здійснення обробки даних.
- способи доступу до інструмента визначають можливості його застосування для вирішення завдань ІВ.

1.5.1 Сховища даних

Парадигма сховищ даних була запропонована в 1980 році компанією IBM [40] в рамках проекту «Сховище даних для бізнесу». На сьогоднішній день системи, побудовані за принципом сховища даних є інформаційні системи, які виконують завдання зберігання і аналізу даних в рамках конкретної предметної області, найчастіше пов’язаної з бізнес-додатками [43]. При цьому, система працює з даними надходять з декількох різнорідних джерел. В один і той же час одночасно зберігаються як історичні дані, так і дані поточного відрізка часу. При цьому, структура аналізованих даних перетвориться до внутрішнього формату сховища, а самі дані проходять предобработку. В якості вихідного інтерфейсу сховища даних виступає вітрина даних (data mart) -спеціальна система, що дає доступ до результатів

обробки наборів даних, що відповідають одній або кільком предметних областях.

Архітектура сховищ даних має на увазі інтеграцію даних відповідно до проблемно-предметної схемою. Дані системи групуються по тим завданням і областям, до яких вони належать, а не по конкретним додатків, їх використовують. Збережені дані не коригуються і не видаляються в процесі роботи сховища. До кожного елементу даних ставитися у відповідність тимчасова мітка.

Сценарії використання сховищ даних досить сильно різняться з точки зору предметної області і переслідуваних цілей [40]. У звіті аналітичної компанії Gartner [46] була проведена систематизація відомих застосувань даної концепції і на підставі зібраних даних була сформульована класифікація сценаріїв на кшталт проведеного аналізу і обробки даних.

Безперервна (близька до обробки в реальному часі) завантаження даних в систему для формування звітів, агрегованих уявлень, статистичних зведень і т.д.

Пакетне завантаження даних, що передбачає періодичне завантаження даних в сховище. В іншому еквівалентна попереднім сценарієм. Автоматизація створення стандартизованих звітів на основі наявних даних. Тактичний бізнес-аналіз існуючих багатовимірних даних, таких, як наприклад OLAP-куби.

Підтримка аналітичних запитів на вимогу.

Підтримка функцій транзакційних систем такі як низькорівневі операції над даними. Класифікація сценаріїв використання демонструє, що метою обробки даних більшості випадків є формування деякого стандартизованого уявлення. Прикладами таких уявлень можуть служити, статистичні звіти, прогнози. При цьому, обробка даних для отримання подібних уявлень може бути досить складно організована з обчислювальної точки зору, що накладає додаткові обмеження на формат аналізованих даних.

1.5.2 Озера даних

Для вирішення завдання обробки великих масивів різнорідних даних Джеймсом Діксоном в 2011 році була запропонована концепція озер даних (Data Lake). Суть концепції полягає в створенні программної системи для зберігання даних в їх вихідному уявленні і їх аналізу. Приклади рішень Apache Hadoop, Azure Data Lake [38], Amazon S3 [18].

Ключовою особливістю озер даних є завантаження всіх даних, важливих для даного завдання якої предметної області без винятків (дані не відкидаються) [95]. При цьому відбувається збереження вихідних форматів даних, що імпортуються на рівні окремих елементів. Перетворення даних і накладення схеми здійснюється в процесі аналізу.

Сценарій використання озер даних має на увазі наступні кроки:

Джерела даних реалізують власні сценарії використання, в рамках яких ігнорується існування озера даних.

Потоки даних від джерел направляються в озеро, де зберігаються в початковому вигляді. Обмежене коло операторів здійснює доступ до озера даних і проводить аналіз його вмісту. Найбільш представницькі результати аналізу передаються іншим користувачам для подальшого використання.

Тому, виходячи із специфіки самого завдання обробки даних у вихідній формі, до роботи озера даних пред’являється ряд додаткових вимог. Імпортовані дані повинні бути відстежувати, тобто для кожного набору необхідно мати можливість ідентифікувати його джерело і час імпорту, інакше користувачі системи не зможуть повною мірою їх використовувати. Також, інфраструктура зберігання повинна мати на увазі високу горизонтальну масштабованість, так як потенційно озеро даних може зберігати досить великі обсяги даних. Крім цього, як правило озера даних представляють інтерес для досить вузького прошарку користувачів – data scientists. Тому подібні системи не повинні містити загальних інтерфейсів доступу.

1.5.3 Порівняння концепцій обробки даних

Порівняємо властивості проаналізованих концепцій відповідно до специфіки завдання обробки геоданих в програмних системах ІВ. Для цього розглянемо такі критерії як використовувана схема даних і пов’язані з нею обмеження, наявність прив’язки даних до часу.

Таблиця 1.1 - Порівняння концепцій обробки даних

	озера даних	сховища даних
схема даних	зберігається вихідна	Перетвориться до формату сховища
Прив’язка до часу	Так	Так
інтерфейси	Інтерфейси для вузького кола користувачів	Інтерфейси для широкого кола користувачів

Зіставлення концепцій показало, що найбільш підходящим принципом виступає озеро даних, так як дана концепція принципово використовує дані в рамках їх початкового формату, що є ключовою перевагою для гетерогенних даних ІВ. Однак, для повноцінного використання озер даних в задачах ІР необхідно вирішити технічну проблему представлення даних систем в машиночитаемом вигляді, так як особливістю ІР є міжмашинної взаємодії.

1.6 Класи задач обробки геоконтекстних даних

Для вироблення рішення по обробці геоконтекстних даних в рамках програмних систем ІВ, необхідно сформулювати класи завдань, які вирішуються даними системами, що дозволить оцінити необхідні критерії вимірювання ефективності вирішення завдань. Формулювання завдань вимагає класифікації і характеристики як використовуваних даних, так і

необхідних результатів в рамках кожного завдання. При цьому обмежимо розгляд результатів в рамках кожного завдання тільки результатами обробки, так як саме вона є найбільш трудомісткою операцією в порівнянні з іншими операціями ГІС, результати яких безпосередньо доступні користувачеві.

З точки зору джерел даних, що розглядаються завдання можуть бути охарактеризовані на підставі структури використовуваних джерел. Головним критерієм, що визначає обчислювальну складність завдання є спосіб представлення геоданих, який визначає формат опису і як наслідок необхідні операції для приведення в цифрову форму. Крім кількості джерел, важливою властивістю є необхідність їх зіставлення. Крім того, складність виконання завдання визначається обсягом вибірки даних – для задач, що використовують Великі дані потрібен спеціальний інструментарій. Підводячи підсумок, можна задати таку класифікацію задач за джерелами даних, в рамках якої можна виділити 8 класів задач, одержуваних комбінуванням значень по кожному з трьох ознак (рисунок 1.7):

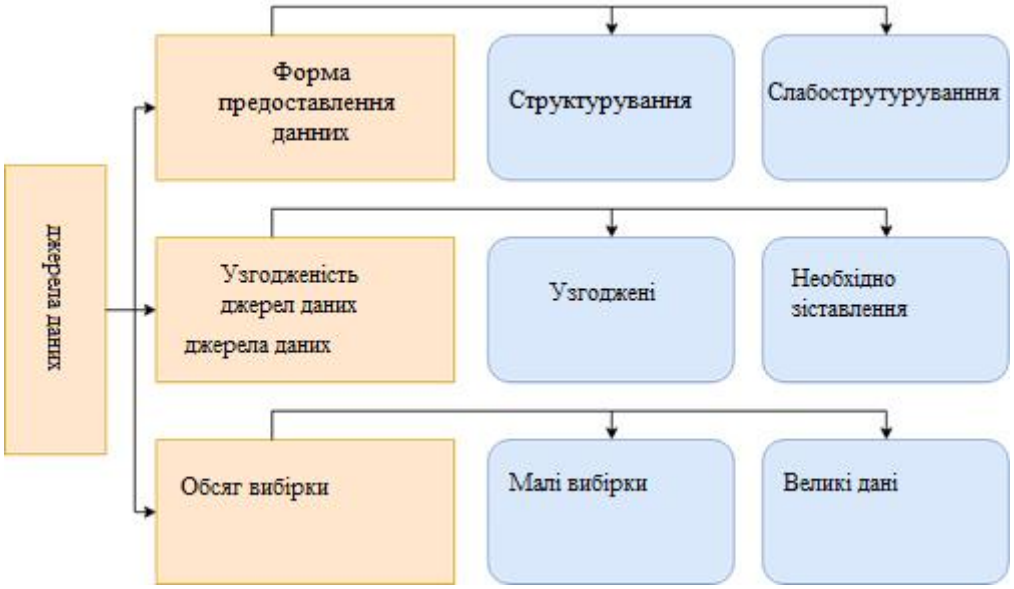

Рисунок 1. 6 – Класи завдань за різними джерелами даних

З точки зору необхідних результатів, що розглядаються завдання можна в першу чергу охарактеризувати за ступенем використання

геоконтекстних даних. Згідно розглянутої моделі життєвого циклу контексту, можливо два способи використання геоконтекста – використання безпосередньо змодельованого геоконтекста і використання з

нань, витягнутих з змодельованого геоконтекста. У першому випадку для задач аналізу використовується тільки цифрове представлення геоконтекста, отримане в ході розмітки елементів джерела даних. У другому випадку проводиться додаткова операція для отримання додаткової інформації, наприклад кластеризація, визначення середніх величин. Іншою важливою характеристикою завдань є періодичність використання рішень. Можна виділити наступні класи задач – виконання на вимогу, періодичне виконання (побудова звітів, планування), постійне виконання (моніторинг).

Таким чином, з точки зору необхідних результатів, завдання характеризуються двома критеріями, що утворюють 6 класів задач, одержуваних комбінуванням значень по кожному з двох ознак (рисунок 1.8):

Рисунок 1.7 – Класи завдань в залежності від необхідних результатів

У Таблиці 1.2 представлені обмеження часу вирішення завдань життєвого циклу контексту в програмних системах ІВ, які диктуються класами, визначеними вище. *Ta (S), Tm (S), Te (S)* відповідають часу отримання, моделювання та вилучення геоконтекста. *S* – параметр масштабу вихідних даних розв’язуваної задачі, який визначається кількістю джерел даних, структурою геоданих, а також необхідністю зіставлення окремих наборів, T – період проведення аналізу, Tt – час передачі даних від джерела

до ІС.

Таблиця 1.2 – Часові обмеження класів задач

	Використання змодельованого геоконтекста	Використання знань, витягнутих з геоконтекста
Аналіз на вимогу	Ta (S)+T m (S)<C , C>0	Ta (S)+T m (S)+Te (S)<C , C >0
періодичний аналіз	Ta (S)+Tm (S)<T	Ta (S)+Tm (S)+Te (S)<T
Аналіз в реальному часі	Ta (S)+T m (S)≪Tt	Ta (S)+T m (S)+Te (S)≪Tt

Так як метою роботи є підвищення ефективності використання геоконтекста в задачах ІВ, то обмежимо подальший розгляд розв’язуваних задач описаними вище шістьма класами. Розгляд класифікації вихідних даних завдань будемо використовувати в контексті класів необхідних результатів.

1.7 Постановка завдання

Рішенням завдання використання геоданих в рамках програмних систем ІВ є формулювання нової моделі даних. Дана модель повинна базуватися на принципах пошарових моделей ГІС і инкапсулировать в них різні форми геоданих ІВ. При цьому модель також має враховувати слабоструктурірованность даних ІВ в широкому сенсі, реалізовувати операції геоконтекстной розмітки, а також надавати інструменти для адаптації до нових джерел даних. Дані вимоги можна формалізувати наступним чином:

- модель повинна описувати елементи геодані пошарово.
- Модель повинна містити операції зіставлення шарів.
- Модель повинна підтримувати довільну схему для описуваних об’єктів і їх атрибутів.

- Модель повинна підтримувати різні структурні одиниці опису місця розташування – точки, багатокутники, ламані і їх довільні сукупності.

- Модель повинна забезпечувати прив’язку об’єктів, що моделюються на часі.

- У моделі повинна забезпечуватися уніфікація різних форм геоданих, як у вигляді географічних координат, так і у вигляді адрес на природній мові.

- Модель повинна уніфікувати годинне з точки зору структурних одиниць опису місця розташування.

- Модель повинна формалізувати фільтрацію даних по тимчасовим мітках, місцем розташування, атрибутам і сукупності цих критеріїв.

- Модель повинна визначати класи операції геоконтекстной розмітки.

- Модель повинна забезпечувати гнучкі терміни моделювання для різних предметних областей.

- Формалізація моделі повинна дозволяти роботу не тільки з геоданих, а й зі слабоструктурованих даними ІВ в широкому сенсі.

- Формалізація моделі повинна мати на увазі можливість прозорого розширення для використання різних форматів і типів даних, а також методів геоконтекстной розмітки.

Описана модель даних повинна породжувати архітектуру проектування ІС, за допомогою якої будуть вирішуватися прикладні завдання ІР. Дана ІС буде являти собою інструмент геоконтекстной розмітки даних програмних систем ІВ. Для цього важливо щоб рішення на базі ІС були узгоджені з системою завдань і вимог, сформульованих в розділі «Класи задач». Як було показано раніше, таке узгодження можливо в разі організації ІС по схемі озера даних. Така ІС повинна бути розширюваної як з точки зору підтримуваних джерел даних, так механізмів доступу до них. Дана мета досягається реалізацією таких вимог:

- Архітектура ІС повинна втілювати концепцію озера даних зі збереженням вихідних форматів даних.

- Архітектура ІС повинна мати інтерфейсами для забезпечення роботи

з гетерогенними даними і протоколами доступу.

- Архітектура ІС повинна надавати зовнішні інтерфейси доступу для програмних систем ІВ, через які здійснюється завантаження даних в систему і вивантаження результатів геоконтекстной розмітки.

- Архітектура ІС повинна надавати інструменти для підключення додаткових коштів виділення геоконтекста.

- Архітектура ІС повинна забезпечувати ізоляцію даних окремих програмних систем ІВ.

Технічні реалізації ІС, які використовують архітектуру, визначену вище, повинні не тільки забезпечувати якісне відповідність вимогам вирішуваних завдань з точки зору продуктивності, але також забезпечувати технічну сумісність з існуючими програмними системами ІВ. Дана мета досягається при виконанні наступних вимог до технічної реалізації ІС:

Рішення на базі технічної реалізації ІС повинні задовольняти вимогам продуктивності відповідних завдань з розділу «Класи задач». ІС повинна реалізовувати інтерфейси доступу програмних систем ІВ на основі протоколів взаємодії ІВ.

ІС повинна забезпечувати інтерфейси для завантаження, читання, зміни, аналізу, візуалізації даних (в тому числі і не містять відомості про місцезнаходження) ІВ.

Оперування даними в рамках ІС повинно відбуватися в рамках форматів даних ІВ.

2 МОДЕЛЬ ОБРОБКИ ГЕОКОНТЕКСТНИХ ДАНИХ В ІНТЕРНЕТІ РЕЧЕЙ

2.1 Модель геоконтекстних даних для програмних систем ІВ

2.1.1 Формальний опис моделі

Для вирішення поставленого завдання пропонується використовувати модель геоданих ІВ, що складається з двох множин кортежів: P – безліч об’єктів, званими точки, і C – безліч об’єктів, званих канали. Дані безлічі є кінцевими і не порожніми.

Точка – окрема одиниця даних, що характеризується унікальним ідентифікатором, описом, інформацією про місцезнаходження (геоконтекст), часовою міткою і зазначенням на канал.

$$p = \langle id, description, location, datetime, channelid \rangle, p \in P. \quad (2.1)$$

Канал – анотація для безлічі точок. Канал характеризується ідентифікатором, назвою і описом:

$$з = \langle id, name, description \rangle, c \in C, \quad (2.2)$$

Зв’язок між каналом і точкою задається шляхом вказівки ідентифікатора каналу в атрибуті channelid.

Для розгляду атрибутів точок і каналів введемо такі множества:

$A = \{(a1, ..., am)\}, m > 0;$

$$K = \{(latitude, longitude, altitude)\} \subset \mathbb{R}3. \quad (2.3)$$

Безліч A являє собою безліч рядків довільної довжини на природній

мові, безліч K – безліч геодезичних координат в системі WGS-84 [13].

Id, channelid – ідентифікатори у вигляді натурального числа для точок і каналів.

$$id, channelid \in \mathbb{N} \quad (2.4)$$

Location – геоконтекст, що задає інформацію про місцезнаходження точки у вигляді кінцевого безлічі цифрових розташування (2.4):

$$location = \{l1 .. lm\}, li = \{li\ 1 .. lin\}, lij \in K, n, m \in \mathbb{N}. \quad (2.5)$$

При цьому l і може задавати як положення одного об’єкта на поверхні

Землі (m = 1), так і бути сукупністю положень одного або декількох протяжних об’єктів (m> 1). Потужність безлічі location для кожної точки є довільною.

Datetime – тимчасові мітки, які містять інформацію про дату і час, асоційованим з цією точкою. З причини необхідності підтримки історичних даних, тимчасові мітки можуть розташовуватися як до, так і після нашої ери. Також з огляду на обмежену роздільну здатність [14], можна вважати безліч таких міток рахунковим. Тому кожної тимчасової мітці можна зіставити ціле число.

$$datetime \in . \quad (2.6)$$

Description – опис точки або каналу у вигляді довільного кінцевого безлічі пар «ключ-значення», що дозволяє зберігати слабоструктуровані дані. «Ключі» і «значення» задаються у вигляді речових і цілих чисел, рядків.

$$description = \{(ai, bi)\}, i \in, a, b \in A., Name – назва каналу. \quad (2.7)$$

$$name \in A. \quad (2.8)$$

Точки і канали відповідають окремим елементам зовнішніх масивів

(джерел) геоконтекстних даних. Процедура перетворення різнорідних зовнішніх даних в дані пропонованої моделі описується в розділі «Метод обробки геоконтекстних даних в рамках концепції ІВ «. В рамках даної процедури відбувається виділення геоконтекста з даних природною мовою:

$$M: A \rightarrow K. \tag{2.9}$$

Для відповідності вимогам модель повинна реалізовувати ряд операцій на множинах точок і каналів (Р і С). Відповідно до поставленої задачі, дані операції повинні відповідати типовим діям, що застосовуються до даних програмних систем ІВ. Дані дії можна розділити на дві групи – читання і запис даних, аргументами яких виступають вказівки на конкретні елементи даних для осущетвелнія вибірки або зміни. Тому, операції моделі можна розділити на два класи – операції, які повертають елементи безлічі точок (ВР), і операції, які повертають елементи безлічі каналів (ОС):

$$Fp: Bp \rightarrow P, Fc: Bc \rightarrow C. \tag{2.10}$$

У класи операцій, які повертають точки, входять операції пошуку і зміни точок на підставі значень окремих атрибутів точки:

$$f\ p\ (location): K \rightarrow P, \tag{2.11}$$

$$f\ p\ (datetime): \mathbb{N} \rightarrow P, \tag{2.12}$$

$$f\ p\ (id): \mathbb{N} \rightarrow P, \tag{2.13}$$

$$f\ p\ (description): A \rightarrow P, \tag{2.14}$$

Прикладами подібних операцій може служить операція пошуку точок по безлічі ідентифікаторів і операція зміни атрибута datetime:

$$f\ p\ (id)\text{: findPointById (id)} = \{p \in P\text{: id} = (id)\}, \quad (2.15)$$

$$f\ p\ (datetime)\text{: changePointDatetime (datetime)} = p, \quad (2.16)$$

$$p = \langle id, description, location, datetime, channelid \rangle, \quad (2.17)$$

$$p = \langle id, description, location, datetime, channelid \rangle, p, p \in P. \quad (2.18)$$

Прикладом операції моделі, що використовує годинне, є операція пошуку точок, чиє місце розташування розташовується в колі з центром (latitude, longitude) і радіусом r.

$$f\ p\ (location)\text{: findByDistance (P, latitude, longitude, r)} = \{P \in P \vee \forall\ l \in location\text{: dist (l, (latitude, longitude))} < r\}, \quad (2.19)$$

де distance – функція обчислення відстані між двома трійками географічних координат.

Для ряду випадків необхідні також операції пошуку і перетворення точок на підставі конкретних точок і каналів:

$$f\ p\ (p)\text{: } P \rightarrow P,$$

$$f\ p\ (c)\text{: } C \rightarrow P. \quad (2.20)$$

Дані операції можуть бути зведені до операцій, що використовують значення окремих атрибутів.

Класи операцій для пошуку і зміни каналів визначаються аналогічним чином:

$$f\ c\ (p)\text{: } P \rightarrow C,\ f\ c\ (id)\text{: } \mathbb{N} \rightarrow C,\ f\ c\ (description)\text{: } A \rightarrow C. \quad (2.21)$$

Як приклад можна визначити операцій для каналів:

$$f\ c\ (p)\text{: pointChannel } (p) = \{c \in C | p.\ channelid = c.\ id\} \quad (2.22)$$

$$f\ c\ (id)\text{: channelById } (id) = \{c \in C\text{: } id = (id)\}, \quad (2.23)$$

$$f\ c\ (description)\text{: channelByDescription } (description) = \{C \in C | description \cap description \neq \emptyset\}. \quad (2.24)$$

Описана вище структура класів операцій є узагальненою, так як для різних предметних областей можуть вимагатися різні реалізації самих відображень. Однак подібна структура дозволяє забезпечити замкнутість моделі щодо її понять точки і каналу, що дозволяє домогтися часткової суперпозиції операцій.

Пропонована модель має переважаючими оцінками швидкодії в порівнянні з OGC і OOM, так як в ній реалізується спільне зберігання як атрибутів об’єкта, так і його геоконтекста. Крім того, обмеження розглянутих систем координат тільки системою WGS-84 як однієї з найбільш популярних також спрощує елементарні операції. Оцінки представлені в Таблиці 2.1.

2.1.2 Відповідність моделі вимогам

Розглянемо, наскільки сформульована модель відповідає пред’явленим раніше вимогам:

Збереження вихідного слабоструктурованого формату даних досягається за рахунок використання атрибута description, що реалізує довільний набір пар «ключ-значення».

Зв’язування даних з джерелом здійснюється завданням назви і опису джерела в атрибутах каналу name і description, а також за рахунок вказівки атрибута channelid для кожної точки, відповідної елементу даних джерела.

Зберігання тимчасових відміток досягається за рахунок атрибута точок datetime.

Фільтрація по тимчасовій мітці, розташування і опису може бути визначена як операція моделі.

Уніфікація різних форм геоданих, зокрема представлених у вигляді географічних координат і у вигляді адрес на природній мові і уніфікація геоданих зі структурної точки зору досягається

Таблиця 2.1– Оцінка швидкодії пропонованої моделі

Операція	Модель геоконтекстних даних для програмних систем ІВ
Додавання нового елемента в модель.	CI (N)
Вибірка елементів, які мають певну парою «атрібут-значення».	CS (N)
Розширення кількості атрибутів елемента в рамках моделі.	CI (N)
Вибірка елементів, чиє місце розташування знаходиться в рамках кола, визначеного радіусу.	CS (N)

Можливість комбінованої фільтрації досягається тим, що всі операції моделі визначаються на підмножині каналів або точок і своїм результатом також мають такі підмножини. Це робить можливим використання в якості аргументу в одних операціях результати виконання інших операцій, тим самим комбінуючи фільтрацію.

Крім виконання вимог, важливою перевагою пропонованої моделі є поєднання даних різних джерел за рахунок узагальнення поняття елемента

геоданих, збереження тимчасової позначки і слабоструктуроване представлення даних як набору пар «ключ-значення», що дозволяє об’єднати різнорідні масиви даних набором уніфікованих і взаімосочетаемих операцій.

2.2 Метод обробки геоконтекстних даних в рамках концепції Інтернету речей

Якісний і кількісний склад відкритих даних як правило дуже сильно відрізняється від джерела до джерела. При цьому, найчастіше, велика частина програмних систем ІВ рідко покладається на один набір даних. Для задач, пов’язаних з використанням геоданих в програмних системах ІВ кількість наборів даних найчастіше більше двох, так як потрібні як дані широкого призначення від електронних пристроїв (сенсори, журнали обчислювальних пристроїв), так і відомості про місцезнаходження. Тому для уніфікації використання геоданих в програмних системах ІВ необхідно враховувати гетерогенність відкритих даних і реалізовувати методи, що забезпечують максимальне абстрагування від формату представлення наборів даних. Класифікація вихідних даних для розглянутих в даній роботі завдань дана в розділі «Класи задач».

У даній роботі створено метод обробки геоконтекстних даних для програмних систем ІВ, завданнями якого є перетворення різнорідних даних з декількох різних джерел в єдине уявлення і витяг геоконтекста з цих наборів. Метод буде розглянуто для найбільш загального випадку з точки зору вихідних даних (кілька наборів даних з необхідністю зіставлення, годинне в змішаній формі, використання повної вибірки), так як при необхідності він може бути зведений до інших класів вихідних даних.

Пропонований метод реалізує поставлені завдання спільно і послідовно. Для цього виконується три етапи роботи з даними (рисунок 2.1): перетворення даних, індивідуальна геоконтекстная розмітка, групова геоконтекстная розмітка. Кожному етапу відповідає однойменний алгоритм.

Рисунок 2.1 – Етапи методу обробки геоконтекстних даних

Перетворення даних – процес приведення відкритих даних в формат, який не залежить від їх вихідного уявлення. Вхідними даними для процесу є елементи джерел відкритих даних, в тому числі і програмних систем ІВ, описані в рамках машиночитаємих форматів (a). Власне перетворення даних включається два етапи: приведення форматів опису до єдиної формі (f 11 (a)) і виділення існуючої в явному вигляді інформації про місцезнаходження і тимчасової оцінки для подальшого зіставлення об’єктів (f 12 (a)). Виконання перетворення даних уможливлює всі подальші операції геоконтекстной розмітки.

$$f\ 11\ (a)\colon A \rightarrow P,\ f\ 12\ (a) \rightarrow C,\ a \in A \qquad (2.25)$$

Індивідуальна геоконтекстная розмітка – етап, на якому для окремих елементів імпортованих даних виробляється перетворення існуючої інформації про місцезнаходження (f 21 (p)) в формат географічних координат і / або його вилучення з існуючих даних, а також відбувається зіставлення різних наборів даних (f 22 (p)) на підставі тимчасових відміток і опису . В результаті виконання даного методу окремі елементи наборів отримують відсутні геоконтексти, що дозволяє в подальшому отримувати геоконтексти для самих наборів.

$$f\ 21\ (p)\colon P \rightarrow P,\ p \in P \qquad (2.26)$$

$$f\ 22\ (c1, c2)\colon C \rightarrow C,\ c2 \in C,\ c1 \in C \qquad (2.27)$$

Групова геоконтекстная розмітка полягає в обчисленні статистичних характеристик цілими розділами даних, а також їх підмножини, і їх зіставлення з індивідуальними географічними контекстами, обчисленими на попередньому етапі.

$$f\ 31\ (c, S\ (p))\colon C \rightarrow P,\ S\ (p)\colon P \rightarrow P,\ c \in C, \qquad (2.28)$$

$$\text{де } f\ 32\ (c, p)\ :(\ C, P) \rightarrow P,\ c \in C,\ p \in P,\ f\ 31\ (c, S\ (p)) \qquad (2.29)$$

- процедура обчислення статистичних характеристик для обраного безлічі, *S (p)* – функція обчислення предметно-залежною статистичної характеристики, *f 32 (c, p)* – процедура зіставлення значення

S (p) з точками моделі.

2.2.1 Перетворення даних

Завданням етапу перетворення даних є переклад імпортованих даних із зовнішніх джерел даних в універсальний формат. В якості універсального формату, що інкапсулює різні джерела даних пропонується наступний набір атрибутів, що спирається на поняття точки, визначене в попередньому розділі:

- дані про місцезнаходження (можуть бути відсутніми).
- Тимчасова мітка.
- Ідентифікатор.
- Опис у вигляді довільного набору пар «ключ-значення».
- Вказівка на джерело.
- Можливість відсутності даних про місцезнаходження можна

пояснити тим, що багато набори даних можуть не містять вказівки на місце розташування, або містити цю інформацію в неявному вигляді. До першої групи належать вимірювання різних сенсорів, які як правило складаються тільки з вимірювань і тимчасової позначки. До другої групи належать тексти на природній мові, в яких містяться вказівки на місця розташування – адреси, назви регіонів, країн, населених пунктів і т.д.

На рисунку 221 наводиться псевдокод для алгоритму перетворення даних. Для досягнення цієї мети виконуються наступні дії:

- для кожного джерела даних створюється канал, що містить опис
- джерела;
- для кожного елемента даних кожного джерела:
- виконується синтаксичний розбір вихідних даних згідно зі схемою даних і / або правилами перетворення,
- створюється точка, прив’язана до каналу джерела,
- атрибути точки заповнюються відпо даними елемента (дані про місцезнаходження, тимчасова мітка, опис).

Data: $\bar{A} = \{a_{ij}\}$, i ∈ [1,n], j ∈ [1,m]
Result: C, P
for *i=1; i ≤ n; i++* **do**
    n_c, d_c ← i-ый источник;
    создать c_i с названием n_c и описанием d_c;
    for *j=1; j ≤ m; j++* **do**
        n_p, d_p, t_p, l_p ← a_{ij};
        создать p_{ij} с атрибутами n_p, d_p, t_p, l_p;
        P ← p_{ij} включить p_{ij} в c_i;
    end
end
C = $\{c_i\}$;
P = $\{p_{ij}\}$;
return C, P

Рисунок 2.2 – Алгоритм перетворення даних

Необхідно відзначити, що виділення атрибутів джерел і окремих елементів наборів даних може бути здійснено за допомогою схем даних, так як для найбільш популярних форматів відкритих даних вже розроблені стандарти опису структури. Однак на практиці частина наборів даних може бути описана в рамках форматів, які не мають на увазі подібної технології. Тому програмні реалізації даного методу повинні передбачати не тільки підтримку найбільш поширених форматів, але також і інтерфейси для розширення.

Результатом алгоритму перетворення даних є уніфіковане представлення елементів даних (точки), а також слабоструктурированное мета- інформація про наборах даних (канали). Завдяки універсальності своєї моделі безліч одержуваних об’єктів можна використовувати для аналізу за допомогою однакових операцій над точками і каналами, визначених у моделі.

2.2.2 Індивідуальна геоконтекстная розмітка

Завданнями етапу індивідуальної геоконтекстной розмітки є геоконтекстная розмітка окремих точок і зіставлення розмічених каналів між собою. При цьому, геоконтекстная розмітка здійснюється за рахунок перетворення описів місця розташування на природній мові в географічні координати алгоритмами геокодування. На рисунку 2.3 представлений алгоритм даного етапу.

Зіставлення розмічених каналів між собою можна уявити як суміщення між собою точок двох каналів в один канал. Спрощено цю операцію можна розуміти як аналог операції з’єднання з реляційної алгебри. Поєднання виконується тільки для тих точок, атрибути яких в сукупності відповідають певним умовам. Таким умовою може бути близькість тимчасових міток для точок, або наявність однакових значень атрибута в описах точок. При цьому правило сполучення крапок формулюється як набір наступних обмежень на

атрибути точки- суміщення:

Місцезнаходження (location) і опис (description) представляють собою об’єднання відповідних атрибутів для першої і другої точки.

Тимчасова мітка вибирається рівною тимчасової мітки першої точки. Тимчасова мітка другий точки включається в опис точки- суміщення.

Data: P, $C_{match} \subset C$
Result: $\bar{P}, \bar{C}$
for $p \in P$ **do**
    d_p = p.description;
    $l_p = \text{geocode}(d_p)$;
    p.location = l_p
end
for (c_i, c_k) *in* C_{match} **do**
    создать $\bar{c_{ik}}$;
    for $j=1; j \leq m; j{+}{+}$ **do**
        $P_{sim} = \text{find\_similar\_points}(p_{ij}, c_k)$;
        $\bar{p} = \text{merge\_attr}(p_{ij}, P_{sim})$;
        включить $\bar{p}$ в $\bar{c_{ik}}$;
    end
end
return $\bar{P}, \bar{C}$

Рисунок 2.3 - Алгоритм перетворення даних

2.2.3 Групова геоконтекстная розмітка

Завданням етапу групової геоконтекстной розмітки є визначення геоконтекстов для груп каналів і / або сукупності кількох підмножин їх точок. Передбачається, що всі канали і підмножини пройшли етап індвідуально геоконтекстной розмітки. Обчислення групових контекстів пропонується робити на підставі статистичних характеристик оброблюваних наборів даних. При цьому весь алгоритм розмітки можна розділити на два етапи:

Обчислення статистики для множин точок на основі метоположенія.

Зіставлення статистики та інформації про місцезнаходження.

Для визначеності позначимо об’єкти (канали і підмножини точок), що підлягають статистичній обробці як D, а канал, з яким буде відбуватися зіставлення по місцю розташування як E. Оскільки не можна сформулювати універсальну величину, що характеризує дані для всіх прикладних задач, то і конкретний вид статистичної характеристики, використовуваної в якості основи побудови геоконтекстной розмітки залежить від розглянутої задачі. Відзначимо, що в загальному випадку процедура для обчислення такої характеристики буде використовувати безліч точок в якості вхідних даних, а набір числових характеристик – як вихідних даних. Також можна стверджувати, що подібна статистика може бути обчислена з використанням наступних характеристик:

- на підставі підрахунку кількості точок в уже згадуваному підмножині,
- на підставі аналізу їх описів та тимчасових міток (кількість або частка точок з певним атрибутом в описі),
- розташування точок в просторі (кількість точок певній галузі).

Для того, щоб результати обчислення статистики могли бути використані в рамках пропонованої моделі, вони повинні мати вигляд точки. Псевдокод алгоритму групової геоконтекстной розмітки представлений на рисунку 2.4.

Зіставлення статистики та інформації про місцезнаходження виконується за алгоритмом, визначеним у попередньому розділі, де зіставляються каналами є cS і E.

Як приклад операції груповий геоконтекстной розмітки можна розвинути приклад з вимірами на автоматизованому заводі. Володіючи набором вимірювань, розширеному інформацією про місцезнаходження сенсорів, можна визначити динаміку показників в рамках окремих приміщень, ділянок і цехів. Для цього буде потрібно обчислити як статистики середні значення показників по окремим тимчасових інтервалах.

Data: P, C, $C_{match} \subset C$, S: $C \rightarrow P$
Result: P, C
$c_s = \emptyset$;
for $c \in C$ **do**
  i = c.id;
  p_i = S(c);
  включить p_i в c_s;
end
for *c in* C_{match} **do**
  i = c.id;
  $\bar{p}$ = get_point(c_s, i);
  for *j=1; j* $\leq$ *m; j++* **do**
    p_{ij} = merge_attr(p_{ij}, $\bar{p}$);
  end
end
return P, C

Рисунок 2.4 – Алгоритм групової геоконтекстной розмітки

Отриманий канал, який містить тимчасову статистику окремих сенсорів, можна зіставити з каналом, що містить координати будівель заводів. В результаті буде отриманий канал, який містить

Результатом етапу групової геоконтекстной розмітки є статистика, прив’язана до географічних координат. При цьому, статистика являє собою груповий геоконтекст для аналізованих джерел і дозволяє отримати для них додаткову інформацію, яку не містять вихідні дані дані.

Висновки.

В цьому розділі пояснювальної записки наведено формальний опис моделі даних для виконання геоконтекстной розмітки даних IP. Описано основні сутності і структури даних. Введено безлічі точок і каналів. Визначено операції на даних множини. Наведені приклади використання різних видів операцій для основних сценаріїв роботи з геоконтекстнимі даними IB. Описана особливість моделі – гнучкість як щодо формату самих джерел, так і щодо структури геоконтекстних даних. Показано, що введена модель задовольняє вимогам поставленого завдання на прикладі аналізу її

операцій і їх зіставлення з вимогами ІР.

Описано метод автоматизованої обробки геоконтекстних даних ІВ. Описано основні етапи обробки – перетворення даних, індивідуальна геоконтекстная розмітка, групова геоконтекстная розмітка. Особливостями запропонованого методу є розширюваність з точки зору нізкоуровнених методів геоконтекстной розмітки, а також наявність етапу групової геоконтекстной розмітки, що дозволяє витягувати нові знання з вже змодельованих геоконтекстов.

3 АРХІТЕКТУРА ІНФОРМАЦІЙНОЇ СИСТЕМИ

У цьому розділі описується архітектура ІС обробки геоконтекстних даних для ІР, яка використовує запропоновану в даній роботі модель і метод роботи з геоконтекснимі даними.

На базі запропонованої архітектури розроблено програмне забезпечення.

Програмне забезпечення, що відповідає пропонованій архітектурі може бути використано для як для широкого спектра наборів даних, що містять відомості про місцезнаходження, так і для різних програмних системах ІР.

3.1 Архітектура програмної системи

3.1.1 Методологія опису архітектури

На сьогоднішній день існує безліч підходів і методологій для опису архітектури програмних систем. Серед найбільш відомих можна виділити такі підходи як «4 + 1», Open Group Architecture Framework, а також стандарт ISO / IEC / IEEE 42010 . У роботах [23, 34] дається порівняльний аналіз даних підходів, на підставі якого робиться висновок про те, що загальною властивістю всіх методик опису архітектури є опис системи з різних точок зору, а також відзначається, що доцільність використання тих чи інших підходів диктується особливостями самої системи. Як узагальнення підходу в роботі [34] пропонується підхід «Views and Beyound», що характеризує роботу системи з точки зору підсистем і їх зв’язків, а також способи взаємодії з системою. В силу узагальненого характеру архітектури, її опис буде грунтуватися на «Views and Beyound», але на увазі важливості способів взаємодії програмних систем ІВ з системою, опис підсистем і зв’язків буде поєднане з описом середовище програмного продукту та сценаріїв застосування.

3.1.2 Підсистеми і модулі

Для опису архітектури розглянемо і охарактеризуємо підсистеми і модулі ІС обробки геоконтекстних даних в порядку взаємодії з ними прикладних програм. Схема архітектури зображена на рисунку 3.1 і представляється як набір шарів, що реалізують різні етапи обробки запитів клієнтських програмних систем.

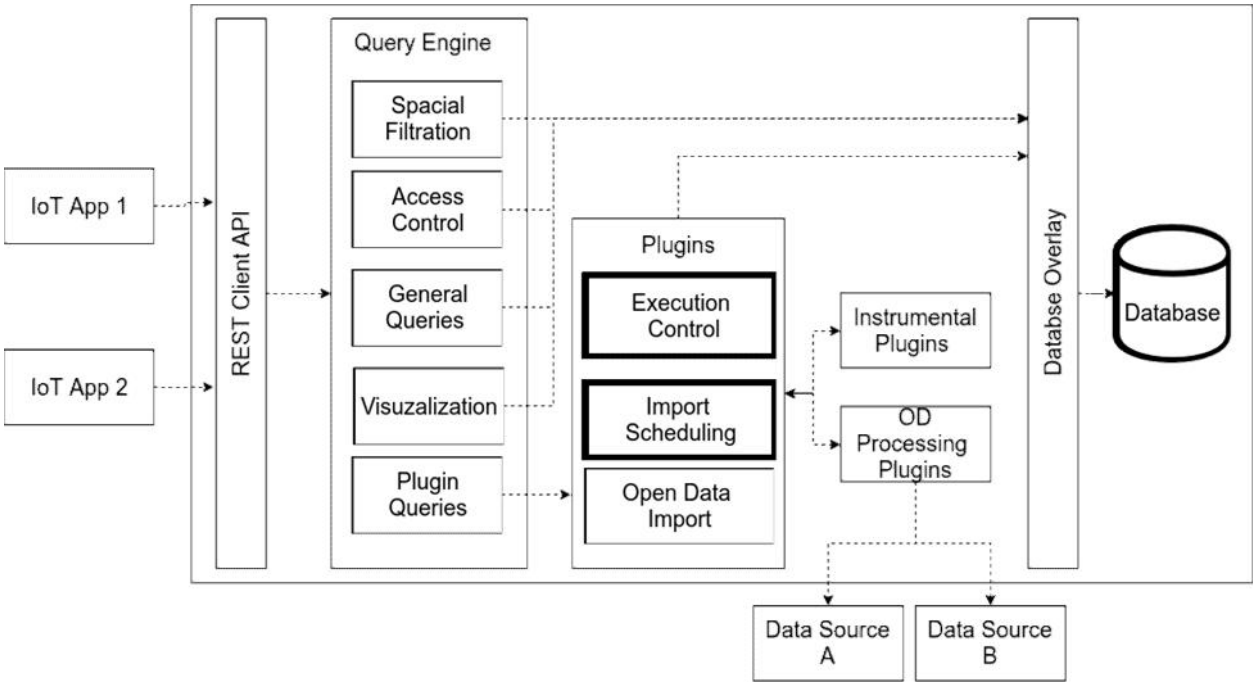

Рисунок 3.1 – Архітектура програмної системи

В процесі взаємодії програмних систем ІВ і ІВ використовується модуль REST [80] Client API, який реалізує обробку спеціального мережевого протоколу прикладного рівня [15, 74]. В рамках протоколу взаємодія є синхронним і будується за принципом «запит клієнта – відповідь системи». Протокол взаємодії инкапсулирует основні функції системи:

Додавання, видалення, зміна і пошук даних в БД системи (General Queries).

Просторова фільтрація даних в БД системи (підсистема Spacial Filtration).

Управління доповненнями (Plugin Queries).

Візуалізація даних (Visualization).

Розмежування доступу користувачів до даних програмних систем ІВ (Access Control).

При цьому протокол реалізує ізоляцію даних різних програмних систем ІВ один від одного, що є актуальним для коректної одночасної роботи програмних систем ІВ як з точки зору приватності даних, так і з точки зору продуктивності. Тому основним завданням модуля REST Client API Web є низькорівневий контроль доступу, що полягає у визначенні наявності необхідних прав користувача на здійснення запитуваної операції.

Інтерпретація запитів в рамках протоколу здійснюється модулем Query Engine. В рамках його роботи відбувається переклад тексту запиту, а також виклик відповідних семантиці запиту процедур для формування відповіді. Процедурами можуть бути або звернення до існуючих призначеним для користувача доповнень, або безпосередньо звернення до модуля взаємодії з БД (Database Overlay).

Plugins є модуль, який реалізує можливість створення, підключення і управління роботою користувальницьких доповнень (плагінів). Для цього підсистемою надаються інтерфейси, що дозволяють:

- підключати та відключати призначені для користувача доповнення від системи в процесі її роботи (підсистема Execution Control),

- управляти обчислювальної навантаженням доповнень (підсистема Import Sheduling),

- надавати інтерфейси для однакової обробки відкритих даних (підсистема Open Data Import).

В контексті описуваної архітектури призначені для користувача доповнення є підпрограми, що реалізують додаткову функціональність системи, і що доповнюють протокол взаємодії з нею новими командами для звернення до даної функціональності. Їх робота відбувається незалежно від основної бізнес-логіки системи. При цьому вони мають опосередкований

доступ до вмісту БД за рахунок інтерфейсів модуля Plugins. Залежно від реалізації і використовуваних прикладних інтерфейсів, призначені для користувача доповнення діляться на дві групи: доповнення імпорту та обробки відкритих даних (OD Import Plugins) і доповнення загального призначення (Instrumental Plugins). До першої групи входять доповнення, завданням яких є реалізація повного циклу завантаження і індивідуальної геоконтекстной розмітки відкритих даних із зовнішніх джерел в рамках методу імпорту та обробки з попередньої глави. Методика реалізації методу обробки геоконтекстних даних в рамках призначених для користувача доповнень імпорту описується в далі. Призначеними для користувача доповненнями загального призначення є всі інші доповнення, в завдання яких не входить завантаження даних із зовнішніх джерел.

Database Overlay являє собою модуль, який реалізує функції взаємодії з прикладної СКБД (Database). Даний шар здійснює перетворення даних з внутрішнього представлення програмної системи ІВ в формат СКБД, а також виконує трансляцію результатів операцій над даними в СКБД.

Необхідно відзначити, що дана архітектура забезпечує можливість створення рішень для всіх класів задач, сформульованих в 1 розділі за рахунок структури підсистем Database, Import Sheduling і Execution Control. Тому конкретні архітектурні виконання цих модулів (в тому числі і тип використовуваної СКБД) визначаються класом завдання.

3.1.3 Підсистеми і зв’язку

На рисунку 3.2 зображена схема взаємодії підсистем, а також показані прикладні проргаммние інтерфейси підключення клієнтських програмних систем ІР і джерел даних до ІС обробки геоконтекстних даних.

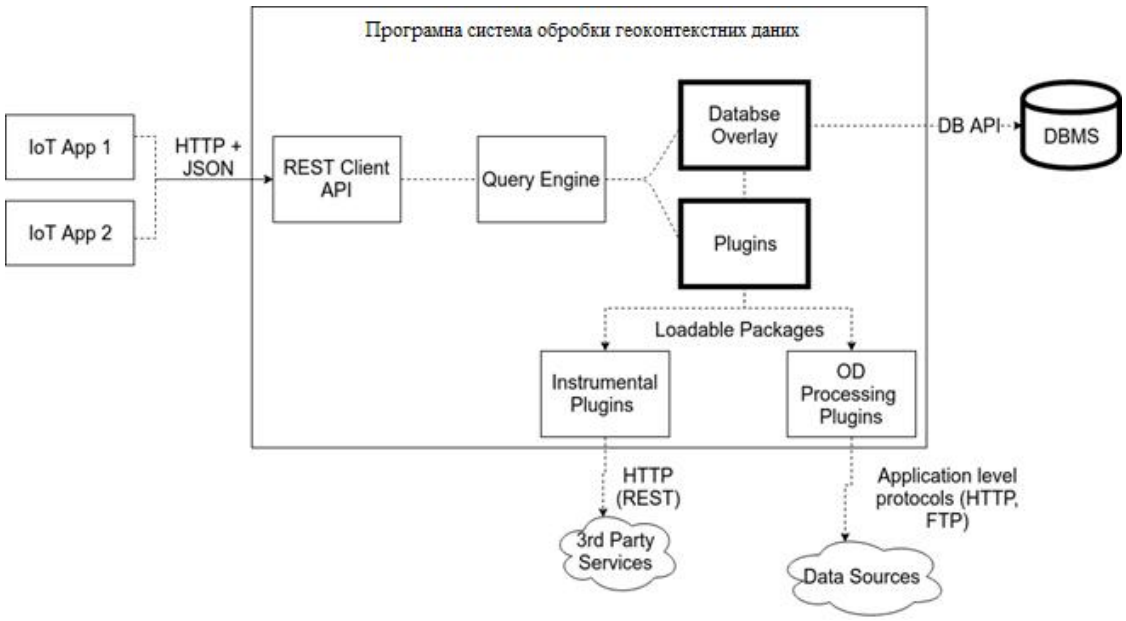

Рисунок 3.2 – Підсистеми і зв’язку

ІС обробка геоконтекстних даних виконується у вигляді монолітної програми, тому описані раніше підсистеми виступають в ній в якості окремих програмних модулів.

Взаємодія з СУБД здійснюється за допомогою клієнтської бібліотеки для мови високого рівня, використовуваного для написання системи обробки геоконтекстних даних. Підсистема Database Overlay реалізує шаблон проектування «Фасад», приховуючи деталі взаємодії і низькорівневі примітиви.

Підключення програмних систем ІВ до ІС обробки геоконтекстних даних здійснюється за допомогою протоколу HTTP і проходить до в режимі

«питання відповідь». Для передачі даних запитів клієнтів і відповідей системи використовується формат JSON.

Призначені для користувача доповнення ІС обробки геоконтекстних даних виконуються у вигляді окремих програм на тій же мові програмування, що і сама система для уникнення проблем сумісності і підвищення продуктивності. Дані програми повинні відповідати вимогам прикладних програмних інтерфейсів підсистеми Plugins і реалізовувати патерн розробки

«Стратегія». Запуск і підключення користувальницьких доповнень здійснюється за допомогою явних REST-запитів користувачів систем, обробкою яких займається REST Client API.

Призначені для користувача доповнення імпорту відкритих даних здійснюють підключення до джерел за допомогою мережевих протоколів прикладного рівня пропонуються джерелами. Інструментальні доповнення можуть використовувати прикладні протоколи для доступу до зовнішніх сервісів, необхідним для виконання операцій геоконтекстной розмітки.

3.2 Реалізація моделі геоконтекстних даних в рамках БД

На рівні СКБД реалізується зберігання і ізоляція даних ІС обробки геоконтекстних даних від даних програмних систем ІВ і даних програмних систем між собою за рахунок зберігання даних системи і програмних систем ІВ в окремих БД (рисунок 3.3). Метадані примірника ІС зберігатися в головній БД (MasterDB), дані окремих програмних систем ІВ зберігаються в окремих БД.

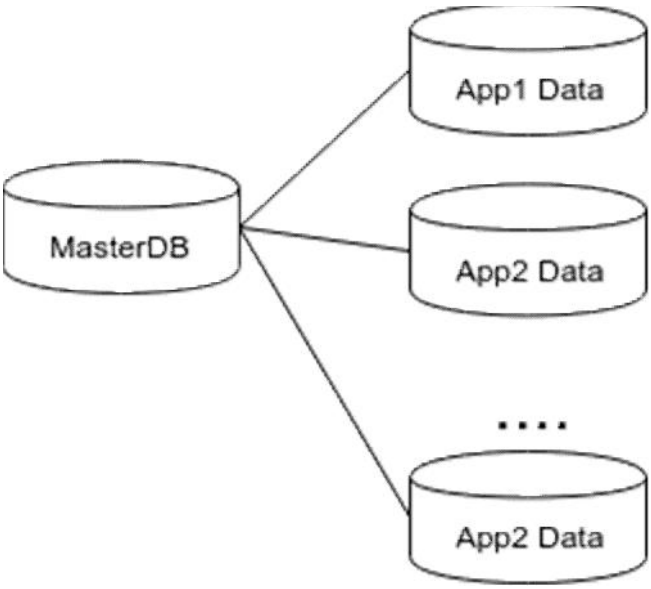

Рисунок 3.3 – БД системи

Для забезпечення уніфікованого використання довільних джерел даних і їх геоконтекстной розмітки архітектура ІС обробки геоконтекстних даних має на увазі використання моделі даних і операцій, визначених у Главі 2, реалізованих на рівні СКБД (модель) і прикладного ПЗ (метод). Основними сутностями моделі є користувачі системи (як кінцеві користувачі програмних систем ІВ, так і адміністратори і власники програмних систем ІВ), програмні системи ІВ, а також канали і точки, еквівалентні відповідним поняттям моделі геоданих в ІВ. Відносини між сутностями моделі даних показані на рисунка 3.4 і 3.5

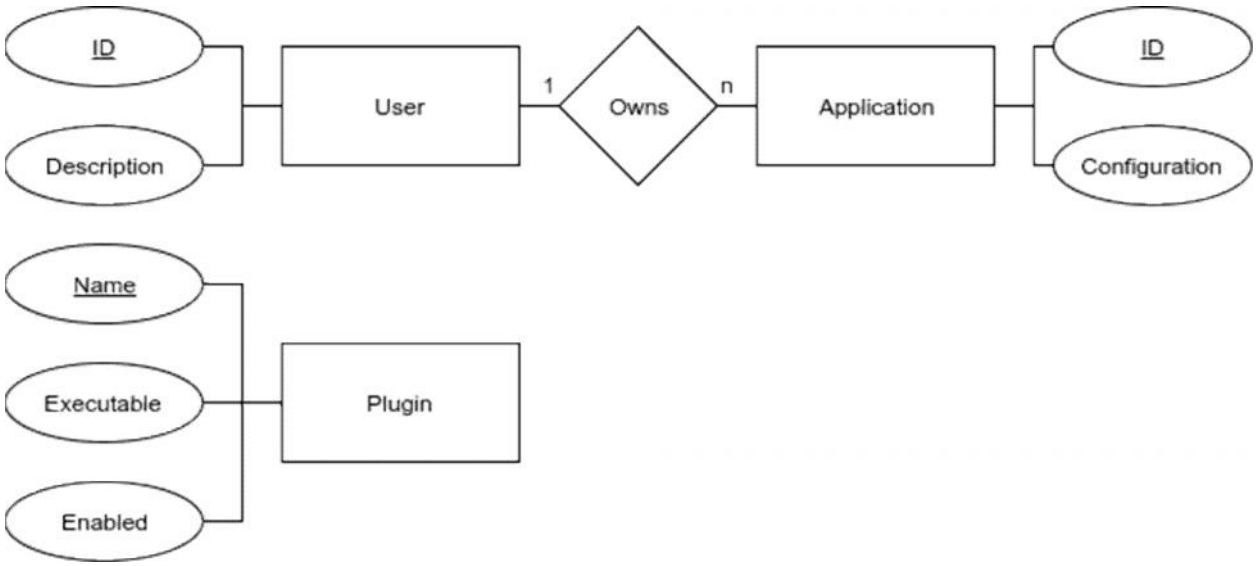

Рисунок 3.4 – ER-діаграма MasterDB

Для забезпечення ізоляції моделі даних БД окремих програмних систем ІВ і модель даних головною БД мають різну будову. Крім цього, додаткова ізоляція забезпечується за рахунок абстрагування користувача від низькорівневих запитів до системи шляхом надання спеціальної мови запитів (3.3).

Вміст головною БД розбите на три колекції – User, Plugin і Application. Колекція User містить основну інформацію про користувачів ІС обробки геоконтекстних даних, а також їх розширений опис (ім’я, прізвище, адресу електронної пошти). Важливо відзначити що в категорію користувачів

потрапляють як власники програмних систем ІВ, так і в свою чергу користувачі програмних систем. У колекції «Application» міститься інформація про програмні системах ІВ, зміни їх роботи, а також про користувача-власника. У колекції «Plugin» міститься інформація про встановлені в системі призначених для користувача додатках – назва (name), шлях до вихідного коду (executable) і вказівка на прапор дозволу виконання (enabled).

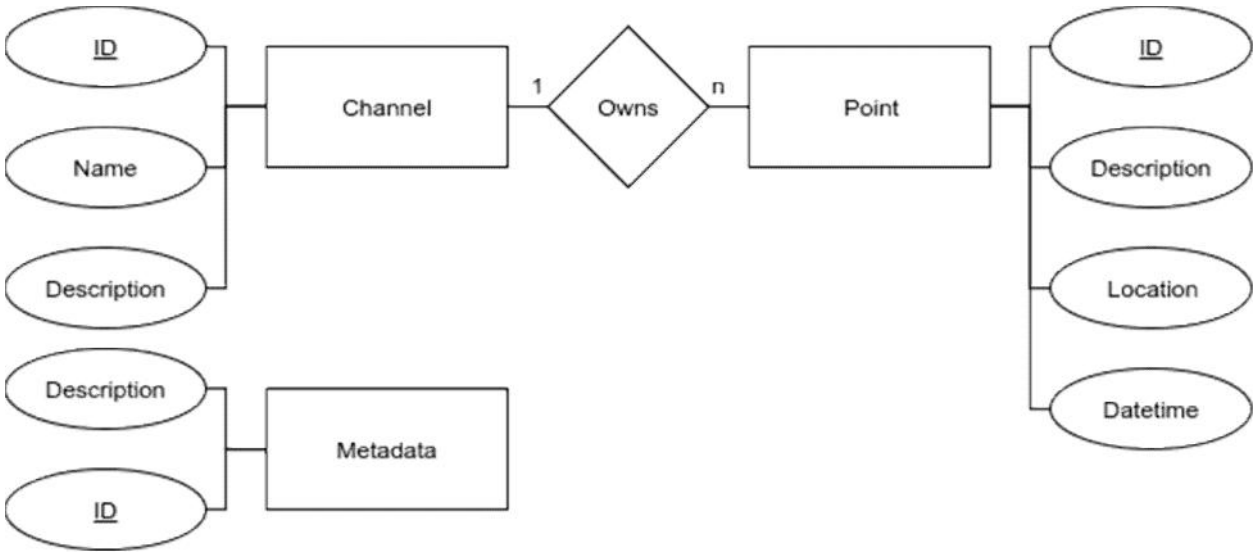

Рисунок 3.5 – ER-діаграма БД програмної системи ІВ

БД програмної системи ІВ в рамках розробленої архітектури ІС обробки геоконтекстних даних являє собою сукупність колекцій, що містять дані точок і каналів, а також метадані. Завданням колекція «Metadata» є зберігання будь-яких даних програмної системи ІВ, які не мають ні тимчасової мітки, ні інформації про місцезнаходження. Окремі елементи даних програмної системи структуруються в якості документів колекції «Point» з атрибутами:

- Опис – набір довільних пар «ключ-значення».
- Місцезнаходження – не порожній набір географічних координат точок.
- Дата і час.

Колекція «Channel» в БД програмної ситеми описується трьома атрибутами, де атрибут «Description» являє собою набір довільних пар ключ-значення. Детальний опис структури окремих колекцій викладено в розділі «Додаток 3. Схеми баз даних».

3.3 Використовувані операції моделі

Для реалізації моделі крім архітектури потрібно також визначити операції для роботи з точками і каналами. Спираючись на запропоновану архітектуру, визначимо операції над даними моделі.

Аналоги функцій distance, in і intersects для аргументів, які приймають тільки значення географічних координат існують і можуть бути визначені алгоритмічно [28]. Але, так як першим аргументом функцій distance, in і intersects можуть бути вказівки місця розташування на природній мові, то дані операції будуть мати сенс тільки при наявності перетворення таких вказівок в географічні координати.

Важливо відзначити, що операції №1 і №2 незважаючи на схожість, є різними і не можуть бути зведені до однієї операції. У задачі знаходження об’єкта в колі найважливішим є саме відстань об’єкта від центру. Подібні завдання виникають, наприклад в сценаріях використання, пов’язаних з автономними роботами, для яких критично важливо порівнювати дальність від зарядної станції з поточним рівнем заряду акумулятора. У той же час, в задачі пошуку об’єктів, що лежать в багатокутнику, головним є саме межа області, яка часто має досить складну форму. До подібних завдань відноситься підрахунок числа статичних об’єктів (записів в соціальних мережах, фотографій з геотеги, будівель) в межах окремих адміністративних одиниць – районів, міст, областей та країн.

3.4 Протокол взаємодії з програмними системами ІВ

Протокол взаємодії програмних систем ІВ з ІС обробки геоконтекстних даних базується на концепції REST і реалізує інтерфейс «питання відповідь». Для відправлення запиту, клієнтам необхідно виконати HTTP- запит до системи по URL, який містить ідентифікатор або декількох ресурсів системи. Ресурси є уявленнями окремих сутностей моделі даних:

- програмні системи ІВ / service / <serviceName>,
- точки / service / <serviceName> / point,
- канали / service / <serviceName> / channel,
- користувачі / service / <serviceName> / user,
- призначені для користувача доповнення / plugin /,
- завдання імпорту відкритих даних
- plugin / <plugin_name> / service / <serviceName> / job.

При цьому, згідно з рекомендаціями реалізації REST-інтерфейсів, спосіб скоєння запиту визначає семантику здійснюється дії:

- GET – читання ресурсу, в тому числі пошукові операції;
- PUT – зміна існуючого ресурсу;
- POST – створення нового ресурсу, завантаження даних в платформу;
- DELETE – видалення ресурсу.

Дані запиту клієнта і інформація про відповідь сервера передається в форматі JSON. Геоконтекст кодується за допомогою формату GeoJSON.

3.5 Методика реалізації методу обробки геоконтекстних даних

Реалізація методу обробки геоконтекстних даних IP здійснюється підсистемою OD Processing Plugins як за рахунок вже реалізованих функцій самої підсистеми, так і за рахунок призначених для користувача додатків, що враховують специфіку джерела даних і розв’язуваної задачі.

Для застосування методу до певного джерела даних необхідно

розробити нове призначене для користувача додаток до ІС, або адаптувати існуючий. При цьому необхідно реалізувати три етапи методу – перетворення даних, індивідуальну і групову геоконтекстную розмітки. Реалізація доповнення виконується за допомогою допоміжних бібліотек підсистеми OD Processing Plugins:

- job – клас для представлення однієї задачі обробки зовнішніх даних;
- job_list_resource_factory – фабрика класів для подання списків примірників job;
- job_manager – клас для пріоритезації примірників job;
- job_resource – клас для надання REST-інтерфейсу однієї job;
- od_import_parser – клас для параметрів обробки;
- open_data_object_address_getter – клас для завдання правила вилучення слабоструктурованої інформації про місцезнаходження;
- open_data_object_to_point_translator – клас для гармонізації формату і приведення його до типу Point;
- open_data_objects_loader – клас, що виконує завантаження даних з
- джерела через мережу передачі даних;
- open_data_objects_parser – клас для поділу даних джерела на окремі елементи;
- open_data_to_points_loader – клас, що виконує запис
- гармонізованих даних в БД;
- perform_import_actions – функція узгодження роботи вищеописаних класів.

3.5.1 Перетворення даних

Процедура розробки користувальницького доповнення вимагає попереднього дослідження властивостей джерела даних для отримання інформації про протоколі доступу; використовуваному форматі даних; тимчасової мітці: формат, поле, що містить мітку; місце розташування: форма запису (цифрове або на природній мові), поля, що містять місце

розташування.

Наступним кроком є написання вихідних кодів для користувача доповнення. Для цього необхідно виконати успадкування від базових класів OD Processing Plugins за наступною схемою (Рисунок 3.6): на підставі особливостей процедури доступу до даних джерела сформувати клас-спадкоємець open_data_import_parser; на підставі схеми даних джерела, яких правил доступу до окремих

Атрибути реалізувати класи спадкоємці для open_data_object_address_getter, open_data_object_to_point_translator, open_data_objects_parser; на підставі вимог до продуктивності рішення реалізувати дисципліни завантаження даних в БД і способи подання до спадкоємців open_data_objects_loader і thread_job відповідно.

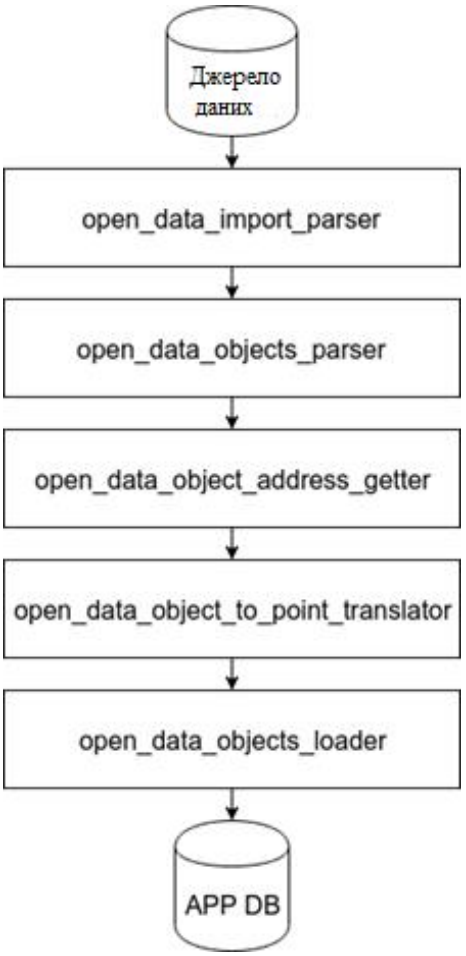

Рисунок 3.6 – Реалізація етапу перетворення даних

3.6.Індивідуальна і групова геоконтекстная розмітка

Розглянемо, яким чином реалізуються етапи індивідуальної і групової геоконтекстной розмітки. Для цього охарактеризуємо складність реалізації окремих підетапів даних процедур:

- зіставлення каналів,
- геокодування,
- обчислення статистичної функції S (n).

Виконання зіставлення в рамках індивідуальної та групової геоконтекстной розмітки здійснюється за рахунок реалізації додаткової бізнес-логіки в класах спадкоємців open_data_objects_loader для випадку, коли заздалегідь відомо, які саме канали необхідно зіставляти. У ситуації, коли пари зіставляються каналів визначаються динамічно, реалізація даної процедури здійснюється за рахунок створення додаткових REST- інтерфейсів користувача доповнення.

Виконання геокодування в рамках індивідуальної геоконтекстной розмітки здійснюється за рахунок використання geocoder, або за рахунок реалізації аналогічного доповнення.

Обчислення статистичної функції S (C), що використовується на етапі групової геоконтекстной розмітки, може бути виконано кількома способами. У разі, коли її обчислення не вимагає наявності всіх каналів в моделі, S (C) може бути обчислена в рамках роботи класу-спадкоємця open_data_objects_loader. Прикладами таких операцій можуть бути статистичні підрахунки в рамках одного каналу: сума окремих полів всіх точок, кількість точок і т. Д. В інших випадках обчислення S (C) може зажадати реалізації як окремих REST-інтерфейсів в рамках поточного користувача доповнення ІС, так і створення окремого доповнення.

3.7 Розробка програмної системи

Метою даної апробації є дослідження практичного застосування запропонованої архітектури. Для цього необхідно перевірити наступні гіпотези:

- можливість паралельної обробки даних.
- Можливість одночасного використання різнорідних джерел геоконтекстних даних.
- Відповідність швидкості роботи системи вимогам продуктивності.

Оскільки архітектура дозволяє створювати ІС, яка відповідає різним класам завдань обробки геоконтекстних даних, то в цілях дослідження необхідно розробити систему для роботи з самим «складним» класом з точки зору вхідних даних: слабоструктуровані неузгоджені геоконтекстні дані, набори даних відповідають критеріям великих даних. З точки зору результатів буде досліджений клас завдань, що включає використання знань, витягнутих з геоконтекста. Клас вимог до продуктивності рішення буде визначено експериментально.

Для апробації пропонованої моделі, методу і архітектури була обрана LBS-платформа Geo2Tag. Дана платформа являє собою ПО, що має відкритий вихідний код. Geo2Tag реалізує модель «Платформа як сервіс» (Platform as a Service) і виступає в якості основи для створення прикладних програм, що використовують геоконтекстні дані. Платформа виконує функції адміністрування та поділу доступу окремих прикладних програм, зберігання і аналізу геоданих.

З точки зору технічної реалізації, Geo2Tag це WSGI-додаток, який використовує інструменти та бібліотеки з відкритим вихідним кодом. Базовою технологією Geo2Tag є Python-фреймворк Flask, який використовує СУБД MongoDB [22] в якості сховища і веб-сервер Apache. Взаємодія клієнтів з платформою здійснюється за допомогою REST-інтерфейсів

Система також підтримує авторизацію користувачів по протоколу OAut, підтримувані провайдери авторизації Facebook і Google.

Для повної реалізації пропонованої архітектури ІС обробки геоконтекстних даних ІВ на базі Geo2Tag були додані наступні компоненти:

- модель геокотекстних даних для ІВ;
- підсистема користувальницьких додатків:
- інструментальні доповнення,
- інструменти контролю виконання доповнень,
- прикладні програмні інтерфейси для створення доповнень імпорту відкритих даних,
- інструменти пріоритизації доповнень імпорту відкритих даних.

Модель геоконтекстних даних для програмних систем ІВ була реалізована в рамках СУБД MongoDB згідно запропонованої архітектури у вигляді структури MasterDB і БД окремих програмних систем ІВ. Сутності, означені в розділі 3.2, були змодельовані у вигляді окремих колекцій відповідних БД, а зв’язки задані зовнішнім ключем. Принципи побудови моделі даних узгоджуються з прийнятими підходами моделювання даних в MongoDB.

Так як запропонований метод обробки геконтекстних даних ІВ базується на геоконтекстній розмітці і, оскільки індивідуальна геоконтекстна розмітка передбачає переклад строкових описів розташування в цифрову форму, то необхідна реалізація інструменту для геокодування строкових уявлень адрес.

Для експериментального дослідження продуктивності паралельної обробки геоконтекстних даних було реалізовано додаток ok_import LBSплатформи Geo2Tag. Дане доповнення реалізовувало послідовне застосування етапів методу для відкритих даних ІС: перетворення форматів до формату Geo2Tag, індивідуальна геоконтекстна розмітка шляхом зіставлення, групова геоконтекстная розмітка за рахунок обчислення середніх значень.

Проведений експеримент включав в себе вимір кількості оброблюваних об’єктів в секунду при зміні кількості паралельно працюючих завдань імпорту. В експерименті використовувався повний набір відкритих даних, що включає в себе 227 368 елементів. Для наочної ілюстрації використовувався ПК наступної конфігурації:

- процесор Intel (R) Core (TM) i3-2310M CPU @ 2.10GHz з двома ядрами,
- ОЗУ в обсязі 3 Гб,
- швидкість з’єднання 100 Мб / с.

В рамках експерименту була виконана низка однотипних процедур обробки при варіюванні кількості завдань в діапазоні 1-30. Подібний діапазон був обраний для перевірки випадків, коли кількість завдань більше кількості ядер, а також для ситуацій, коли можливості паралельної обробки перевищують навантаження. Зіставлення результатів експерименту з відомими оцінками продуктивності для OGC і OOM [10] показало перевагу в швидкості імпорту об’єктів для запропонованого методу. Візуалізація усереднених вимірювань для кожного значення кількості одночасних завдань показана на рисунку 3.7.

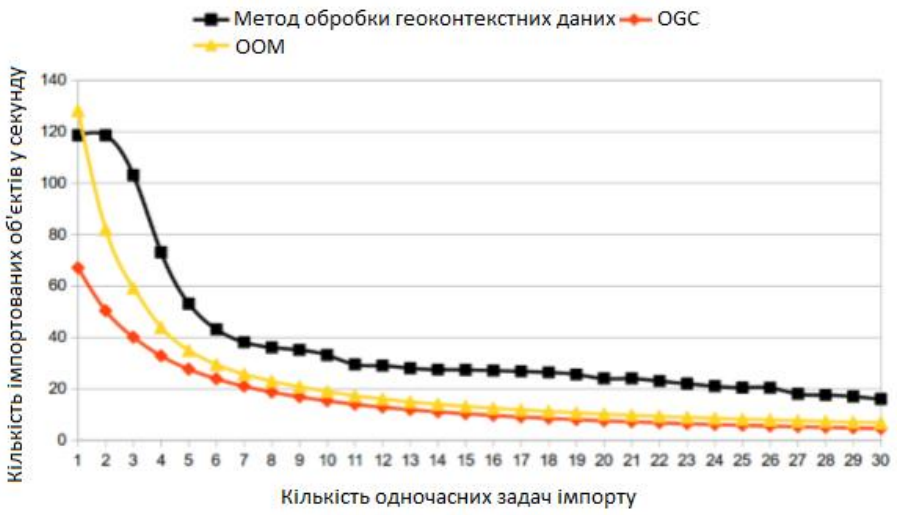

Рисунок 3.7 – Продуктивність паралельної обробки геоконтекстних даних

При перевищенні кількістю одночасних завдань кількості ядер відбувається різке зниження кількості одночасно імпортованих об’єктів. Дане явище викликається тим, що в проведеному експерименті були використані додаткові інструменти диспетчеризації завдань, крім вбудованих інтерфейсів Apache, що призводить до неефективної роботи в разі перевищення кількістю завдань кількості ядер. Однак, отримана в ході експерименту залежність кількості оброблюваних об’єктів показують, що, не дивлячись на обмеження кількості потоків виконання і на технологію Global Interpreter Lock (GIL) [68], вбудовану в реалізацію Python, кількість оброблюваних об’єктів знижується з повільною швидкістю. При збільшенні кількості одночасних завдань імпорту до 30, кількість одночасно імпортованих об’єктів знижується в шість разів. Крива зниження має близький до лінійного характеру залежності на ділянці 6-30, для опису якого можна використовувати наступне рівняння, отримане за методом лінійної регресії:

$$y=-0.9\,x+38.0,$$

де y – швидкість обробки, x – кількість одночасних завдань імпорту. Тому можна стверджувати, що динаміка швидкості імпорту при паралельному виконанні демонструє принципову стійкість експериментальної системи до підвищення навантаження.

ВИСНОВКИ

Досліджені існуючі моделі оброки геоконтекстних даних в IoT. Наведено формальний опис моделі даних для виконання геоконтекстної розмітки даних IoT. Описано основні сутності і структури даних. Наведені приклади використання різних видів операцій для основних сценаріїв роботи з геоконтекстнимі даними IoT. Описана особливість моделі – гнучкість як щодо формату самих джерел, так і щодо структури геоконтекстних даних. Показано, що введена модель задовольняє вимогам поставленого завдання на прикладі аналізу її операцій і їх зіставлення з вимогами IoT

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Дяченко В.О., Міхаль О.П., Соловйова А.С. Аналіз методів автоматизованої обробки даних в IOT // Сучасні напрями розвиткуінформаційно-комунікаційних технологій та засобів управління. Матеріали десятої міжнародної НТК. – Баку: ВА ЗС АР; Харків: НТУ «ХПІ»; Харків: ДП»ПДПРОНДІАВІАПРОМ»; Жиліна: УмЖ, 2020. – 9-10 квітня 2020. – С. 88.

2. Интеллектуальные географические информационные системы для мониторинга морской обстановки // Под общ. ред. чл.-кор. РАН Юсупова Р.М. и д-ра тех. наук Поповича В.В. - СПб.: Наука, 2013. – 284 с.

3. Клейнрок Л. Теория массового обслуживания. – М.: Машиностроение, 1979. – С. 432.

4. Марин В.А., Заславский М.М. Технология контейнеризации LBSплатформы Geo2Tag для использования в рамках модели PaaS // Международная конференция по мягким вычислениям и измерениям –2016. – Т. 2. – No Секции 4–7. – С. 85-88.

5. Система координат WGS 84 – основные параметры // Geostart. [Электронный ресурс]. URL: https://geostart.ru/wgs_84

6. Таненбаум Э. С. Архитектура компьютера:[пер. с англ.]. – Издательский дом» Питер», 2011. – 843 с

7. Таненбаум Э., Уэзеролл Д. Компьютерные сети. 5-е изд. – СПб.: Питер, 2012. – 960 с.: ил. ISBN 978-5-459-00342-0.

8. Технология обработки треков // Яндекс. [Электронный ресурс]. URL: https://yandex.ru/company/technologies/yaprobki/#h-2

9. A Tutorial on Clustering Algorithms. K-Means Clustering // Politecnico diMilano. [Электронный ресурс]. URLhttps://home.deib.polimi.it/ matteucc/Clustering/tutorial_html/kmeans.html

10. Amazon S3 – Amazon AWS // Amazon S3. [Электронный ресурс].

URL https://aws.amazon.com/s3/

11. Ashton K. The Internet of Things // RFiD Journal. – 2009. – №. 50. – С. 1.

12. Balandina E., Balandin S., Koucheryavy Y., Mouromtsev D. Innovative eTourism Services on top of Geo2Tag LBS Platform", the 11th InternationalConference on Signal Image Technology & Internet Systems (SITIS 2015), Bangkok, Thailand. 2015. – С. 752-759.

13. Balandina E.K., Balandin S.I., Koucheryavy Y.A., Zaslavskii M.M. Role ofMobile OS and LBS Platform in Design of e-Tourism SmartServices//UBICOMM 2016, The Tenth International Conference on MobileUbiquitous Computing, Systems, Services and Technologies, IET – 2016, С. 172-177.

14. Banker K. MongoDB in action. – Manning Publications Co., 2011. — 312 с.

15. Bass L., Clements P., Kazman R. Software architecture in practice. 3-е изд. –Addison-Wesley Professional, 2013. — 662 с. 24. Batra S., Sachdeva S., Bhalla S. Entity Attribute Value Style Modeling Approachfor Archetype Based Data //Information. – 2017. – Т. 9. – №. 1. – С. 2.

Beckmann N. et al. The R*-tree: an efficient and robust access method for points and rectangles //Acm Sigmod Record. – Acm, 1990. – Т. 19. – №. 2. – С. 322- 331. doi:10.1145/93597.98741. ISBN 0897913655.

16. Berlenko T.A., Krinkin K., Zaslavskiy M. Proximity points approach for building recommendation system for the "Open Karelia" information system // Proceedings of the 18th International Conference on Soft Computing and Measurements, SCM 2015 (St.Petersburg, May 19-21) – 2015, С. 176-178.

17. Berners-Lee T., Hendler J., Lassila O. The Semantic Web // Scientific American. – 2001. – Т. 284, №. 5. – С. 28–37.

18. Bessel, F. W. «The calculation of longitude and latitude from geodesic measurements». // Astronomische Nachrichten. 331 (8): 852–861. [Электронный ресурс]. URL: https://arxiv.org/pdf/0908.1824.pdf

19. Bizer C., Heath T., Berners-Lee T. Linked data-the story so far

//International journal on semantic web and information systems. – 2009. – Т. 5. – №. 3. – С. 1- 22.

20. Bizer C., Heath T., Berners-Lee T. Linked data: The story so far //Semantic services, interoperability and web applications: emerging concepts. – IGI Global, 2011. – С. 205-227.

21. Bonham-Carter G. F. Geographic information systems for geoscientists: modelling with GIS. – Elsevier, 2014. – Т. 13. – 416 с.

22. Boniface M., Nasser B., Papay J., Phillips S. C., Servin A., Yang X., Kousiouris G. Platform-as-a-service architecture for real-time quality of service management in clouds //Internet and Web Applications and Services (ICIW), 2010 Fifth International Conference on. – IEEE, 2010. – С. 155-160.

23. BSON // BSON (binary JSON) serialization. [Электронный ресурс]. URL: http://bsonspec.org/

.