

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Кваліфікаційна наукова
праця на правах рукопису

ТИМЧИШИН БОГДАН СТЕПАНОВИЧ

УДК: 519.876: 004.41

ДИСЕРТАЦІЯ

**ІНТЕЛЕКТУАЛІЗОВАНА ПРОГРАМНА СИСТЕМА ДЛЯ
ПРОАКТИВНОГО УПРАВЛІННЯ ЕКОСИСТЕМОЮ МІСТА**

Спеціальність 121 – Інженерія програмного забезпечення
Галузь знань 12 – Інформаційні технології

Подається на здобуття ступеня доктора філософії.

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

/  / Б.С. Тимчишин

Науковий керівник: Манжула Володимир Іванович, доктор технічних наук,
доцент

Тернопіль – 2026

АНОТАЦІЯ

Тимчишин Б.С. Інтелектуалізована програмна система для проактивного управління екосистемою міста. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття ступеня доктора філософії за спеціальністю 121 – Інженерія програмного забезпечення – Західноукраїнський національний університет, Тернопіль, 2026.

Підготовка здійснювалась на кафедрі комп'ютерних наук Західноукраїнського національного університету Міністерства освіти і науки України.

У дисертаційній роботі розглянуто науково-прикладне завдання підвищення ефективності управління екосистемою урбанізованих територій шляхом розробки семантичних моделей, методів та інтелектуалізованої програмної системи для проактивного сценарного аналізу та оптимізації розміщення джерел техногенного навантаження.

У першому розділі виконано теоретичне узагальнення та обґрунтування підходів до розроблення інтелектуалізованої програмної системи проактивного управління екосистемою міста за умов невизначеності в даних. Встановлено, що традиційне реактивне управління («post-factum») є неефективним в умовах нелінійних та нестаціонарних процесів та щільної забудови, оскільки характеризується часовою затримкою між інцидентом та реакцією. Доведено необхідність переходу до проактивної моделі, яка базується на екологічному моделюванні для аналізу станів екосистеми міста з метою підвищення екологічної безпеки населення.

Проведено компаративний аналіз класів математичних моделей прогнозування розсіювання домішок в атмосфері від точкових джерел. Визначено, що для задач оперативного сценарного аналізу, який вимагає багаторазових розрахунків у прийнятному часі, оптимальними є Гаусові моделі, зокрема обчислювальне ядро AERMOD. Вони забезпечують найкращий баланс

між точністю та обчислювальною складністю, на відміну від CFD та Ейлерових моделей, які є надмірно ресурсоемними для оптимізаційних задач. Проаналізовано структурну організацію модуля AERMOD та його складових (AERMET, AERMAP, BPIPPRIME). Показано, що його модульність та файловий інтерфейс дозволяють інтегрувати його як обчислювальне ядро в автоматизовані системи прийняття рішень.

Здійснено критичний огляд існуючих програмних засобів: AERMOD View, BREEZE AERMOD, SCREEN View. Виявлено критичний технологічний розрив – існуючі пакети є монолітними десктопними додатками, орієнтованими на статичний проектний аналіз та ручне введення даних. Вони не мають API для інтеграції з зовнішніми ресурсами та не підтримують автоматизований цикл «вхідні дані – моделювання – рішення». Обґрунтовано, що вирішення задач проактивного управління можливе шляхом розроблення інтелектуалізованої програмної системи. Вона має виступати як «оркестратор», що автоматизує роботу математичного ядра, забезпечує безшовний обмін даними та реалізує алгоритми сценарного аналізу, перетворюючи AERMOD на інструмент системи прийняття рішень. Здійснено постановку завдань дисертаційного дослідження.

У другому розділі обґрунтовано перехід від неструктурованих, файло-орієнтованих робочих процесів до формалізованої керованої знаннями моделі, що є необхідною передумовою для створення інтелектуалізованої системи проактивного управління. Проведений аналіз довів, що традиційні підходи до зберігання даних (файли, реляційні БД) є семантично бідними, негнучкими та нездатними ефективно моделювати складні, гетерогенні сутності міської екосистеми, що унеможлиблює автоматизацію сценарного аналізу.

Розроблено та обґрунтовано гібридну архітектуру збору та інтеграції даних. Дана архітектура використовує онтологію UESO як центральний семантичний хаб, що інкапсулює складність гетерогенних джерел (AERMET, BPIPPRM, кадастр об'єктів теплогенерації) та відокремлює логіку знань від зберігання артефактів (файлове сховище) та гео-даних (PostGIS).

Спроектовано доменну онтологію UESO. Розроблена ієрархія класів формалізує не лише фізичні об'єкти (джерела та рецептори), але й ключові управлінські та обчислювальні сутності, забезпечуючи при цьому інтероперабельність через наслідування від онтологій верхнього рівня (SWEET, GeoSPARQL).

Формалізовано основи сценарного аналізу. На базі онтології UESO надано формалізоване визначення двох типів проактивного аналізу: оцінювального та оптимізаційного сценаріїв. Оцінювальний сценарій формалізовано як повний, самодостатній кортеж, що забезпечує відтворюваність експериментів «що-якщо». Оптимізаційний сценарій формалізовано як задачу комбінаторної оптимізації, для якої необхідно застосовувати метаевристичні алгоритми.

У третьому розділі розроблено математичну постановку задачі комбінаторної оптимізації розміщення джерел антропогенного забруднення, зокрема об'єктів теплогенерації враховуючи забудову міста. На відміну від класичних геометричних задач розміщення, запропонований підхід враховує суперпозицію полів концентрацій в точках-рецепторах та використовує модель AERMOD для обчислення цільової функції. Для гарантування дотримання екологічних стандартів цільова функція, що враховує індекс якості повітря, модифікована методом штрафних функцій, що дозволяє мінімізувати максимальний ризик перевищення ГДК у точках-рецепторах. Перехід до дискретного простору пошуку дозволив уникнути проблем локальних мінімумів, характерних для неперервних функцій.

Адаптовано метаевристичні алгоритми (генетичний алгоритм, дискретний метод рою часток, дискретний алгоритм бджолої колонії) для вирішення задач дискретної оптимізації фіксованої розмірності. Розроблено та реалізовано у фреймворку DEAP три уніфіковані оператори: `Create_Random_Solution()`, `Random_Swap()` та `Targeted_Swap()`, що дозволило уніфікувати програмну реалізацію різних евристик та гарантувати валідність рішень.

Запропоновано метод інтелектуалізованого вибору стратегії оптимізації, який базується на аналізі складності задачі. Введення порогів складності та масштабу дозволяє системі автоматично перемикатися між повним перебором (для тривіальних задач), D-PSO (для задач тактичного рівня) та D-ABC (для стратегічних задач високої розмірності). Це забезпечує баланс між часом отримання рішення та гарантією його оптимальності в умовах високої обчислювальної вартості звернень до модуля AERMOD. Проведено апробацію методу на прикладі розміщення трьох когенераційних установок у м. Тернопіль. Експеримент показав, що центральна частина мікрорайону є екологічно перевантаженою через високий фоновий рівень забруднення. На основі методу визначено оптимальну стратегію децентралізації критичної інфраструктури теплогенерації, запропонувавши розміщення, що дозволяє мінімізувати вплив на житлову забудову та навчальні заклади.

У четвертому розділі спроектовано гнучку сервіс-орієнтовану архітектуру програмної системи, ключовим елементом якої є розробка спеціалізованих сервісів-обгорт. Це дозволило успішно інтегрувати ядро регуляторної моделі AERMOD у сучасне веб-середовище, забезпечивши автоматизацію повного циклу розрахунків без необхідності ручного втручання в консольні процеси. Розгортання системи в кластері Kubernetes гарантує динамічне масштабування обчислювальних ресурсів при зростанні навантаження.

Також програмно реалізовано підсистему збереження даних на основі гібридної моделі. Поєднання семантичного сховища (для онтології UESO), реляційної бази даних з розширенням PostGIS (для геометрії) та об'єктного файлового сховища забезпечило реалізацію концепції «єдиного джерела істини». Такий підхід забезпечує синхронізацію даних та дозволяє автоматично валідувати сценарії моделювання ще до початку ресурсоемних розрахунків.

Розроблено інтерактивний веб-інтерфейс користувача із застосуванням UX-патернів, спрямованих на зниження когнітивного навантаження на

користувача. Реалізовані інструменти візуалізації (теплові карти забруднення, ізолінії, інтерактивні шари забудови) дозволяють оперативно інтерпретувати результати моделювання та приймати обґрунтовані управлінські рішення без необхідності глибокого аналізу текстових звітів. Результати навантажувального тестування засвідчили суттєве підвищення продуктивності: завдяки автоматизації підготовки даних та розпаралеленню обчислень час отримання розрахункового сценарію скорочено з годин (при ручному режимі) до хвилин (у 24 рази), що робить систему придатною для оперативного та гнучкого управління.

У висновках сформульовано основні результати дисертаційної роботи.

Ключові слова: інтелектуалізована програмна система, архітектура програмного забезпечення, програмне забезпечення, математичне моделювання, математична модель, нелінійні нестационарні процеси, екологічне моделювання, прогнозування, нелінійні задачі, ідентифікація, числові методи, точкове джерело, обчислювальний модуль, онтологія, оптимізаційна задача, методи оптимізації, цільова функція, метаввристичні алгоритми, критична інфраструктура, теплогенеруючі об'єкти, проактивне управління, екосистема міста, забруднення атмосфера, невизначеність даних, система підтримки рішень.

ПЕРЕЛІК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Наукові праці, в яких опубліковано основні наукові результати дисертації:

1. Тимчишин Б., Манжула В. Математичне та алгоритмічне забезпечення інтелектуалізованої програмної системи для проактивного управління екосистемою міста. *Вимірювальна та обчислювальна техніка в технологічних процесах*. 2025. вип. 84, вип. 4, с. 35–48. (1,2 д.а. / 0,9 д.а.; особистий внесок: запропоновано математичне та алгоритмічне забезпечення інтелектуалізованої системи для сценарного аналізу в межах проактивного управління екосистемою міста, формалізовано задачу оптимізації розміщення

джерел техногенного навантаження (зокрема, котелень та когенераційних установок) в умовах щільної міської забудови, яка NP-складною задачею дискретної комбінаторної оптимізації, запропоновано метод адаптивної оптимізації, який на основі порогів складності та масштабу автоматично обирає найбільш ефективний алгоритм).

DOI: <https://doi.org/10.31891/2219-9365-2025-840-4>

2. В.І. Манжула, Б.С. Тимчишин. Семантичне моделювання гетерогенних джерел шкідливих викидів в атмосферу для проактивного управління екосистемою міста. *Наукові праці Донецького національного технічного університету. Серія: “Інформатика, кібернетика та обчислювальна техніка”*. 2025. №2 (41), с. 118–127. (1 д.а. / 0,8 д.а.; особистий внесок: розроблено доменну онтологію UESO, яка формалізує знання про гетерогенні міські джерела викидів та забезпечує семантичну інтероперабельність та логічну несуперечливість даних. Спроектовано та обґрунтовано сервісно-орієнтовану архітектуру інтелектуалізованої системи, де онтологія виступає «єдиним джерелом істини»).

DOI: <https://doi.org/10.31474/1996-1588-2025-2-41-118-127>

3. Тимчишин В., Мельник А., Дивак Т., Тимчишин Б., Файфура В., Момотюк Є., Костик Б. Інтегральний показник для оцінювання ефективності спеціалізованих моніторингових систем. *Оптико-електронні інформаційно-енергетичні технології*. 2024. 48(2). с. 234–241. (0,8 д.а. / 0,2 д.а.; особистий внесок: аналіз методів комплексної оцінки ефективності впровадження інтелектуалізованих систем моніторингу для підтвердження практичної цінності запропонованої системи).

DOI: <https://doi.org/10.31649/1681-7893-2024-48-2-234-241>

4. Tymchyshyn V., Tymchyshyn B., Melnyk A., Manzhula V., Faifura V., Romanets I. The System Architecture of the Software for Modeling Harmful Emissions in Soil. Proceedings International Conference on Advanced Computer Information Technologies ACIT. 2023, P. 58–62. (0,5 д.а. / 0,1 д.а.; особистий внесок: обґрунтовано принципи побудови сервіс-орієнтованої архітектури для

екологічного моделювання, апробовано підхід до розбиття складної системи на незалежні модулі, який трансформувався в архітектуру «Оркестратора» для керування ядром AERMOD).

DOI: <https://doi.org/10.1109/ACIT58437.2023.10275416> (Scopus)

Наукові праці, які засвідчують апробацію матеріалів дисертації:

5. Tymchyshyn V., Melnyk A., Tymchyshyn B., Faifura V., Mazur I.-S., Honchar L. Data Management Service Architecture of the Software for Modeling Harmful Emissions in Soil. Proceedings International Conference on Advanced Computer Information Technologies ACIT. 2024, P. 678–682. (0,5 д.а. / 0,1 д.а.; особистий внесок: розроблено механізми збору, валідації та маршрутизації гетерогенних даних для створення підсистеми збору даних, яка інтегрує Weather API та параметри джерел).

DOI: <https://doi.org/10.1109/ACIT62333.2024.10712526> (Scopus)

6. Tymchyshyn V., Porplytsya N., Melnyk A., Tymchyshyn B. Software for Modelling the Air Pollution by Vehicles. CEUR-WS: Proceedings of the International Conference Advanced Computer Information Technologies (ACIT 2018). 2018. Vol 2300. pp. 207-210. (0,5 д.а. / 0,1 д.а.; особистий внесок: аналіз методів моделювання фонового забруднення від автотранспорту, що використано як вхідний параметр для оптимізаційних задач розміщення теплогенеруючих об'єктів).

<https://ceur-ws.org/Vol-2300/Paper50.pdf> (Scopus)

7. Тимчишин В.С., Порплиця Н.П., Тимчишин Б.С. Програмний комплекс для моделювання забруднення атмосфери шкідливими викидами автотранспорту в часі. Сучасні комп'ютерні інформаційні технології: Матеріали Всеукраїнської конференції з міжнародною участю ACIT'2017. Тернопіль: ТНЕУ, 2017. с.233-234. (0,2 д.а. / 0,05 д.а.; особистий внесок: проведено аналіз програмних засобів для моделювання забруднення атмосфери шкідливими викидами).

<https://dspace.wunu.edu.ua/items/55add516-6724-4334-8e74-9f53b6d3bbcc>

ANNOTATION

Tymchyshyn B.S. Intelligent software system for proactive urban ecosystem management. – Scientific work on the rights of the manuscript.

Thesis for the degree of Doctor of Philosophy in the specialty 121 – Software Engineering – West Ukrainian National University, Ternopil, 2026.

The research was carried out at the Department of Computer Sciences of the West Ukrainian National University, Ministry of Education and Science of Ukraine.

The dissertation addresses the scientific and applied task of improving the efficiency of managing the ecosystem of urbanized territories by developing semantic models, methods, and an intellectualized software system for proactive scenario analysis and optimization of the placement of sources of technogenic load.

The first chapter provides a theoretical generalization and substantiation of approaches to developing an intellectualized software system for proactive urban ecosystem management under conditions of data uncertainty. It is established that traditional reactive management ("post-factum") is ineffective under conditions of non-linear and non-stationary processes and dense building development, as it is characterized by a time delay between the incident and the response. The necessity of transitioning to a proactive model, which is based on ecological modeling for analyzing the states of the city's ecosystem to enhance the environmental safety of the population, is proven.

A comparative analysis of classes of mathematical models for forecasting the dispersion of impurities in the atmosphere from point sources was conducted. It was determined that for tasks of operational scenario analysis, which require multiple calculations within an acceptable time frame, Gaussian models, particularly the AERMOD computational core, are optimal. They provide the best balance between accuracy and computational complexity, unlike CFD and Eulerian models, which are excessively resource-intensive for optimization tasks. The structural organization of the AERMOD module and its components (AERMET, AERMAP, BPIPPRIME) is analyzed. It is shown that its modularity and file interface allow for its integration as

a computational core into automated decision support systems.

A critical review of existing software tools (AERMOD View, BREEZE AERMOD, SCREEN View) was carried out. A critical technological gap was identified: existing packages are monolithic desktop applications oriented towards static project analysis and manual data entry. They lack APIs for integration with external resources and do not support an automated "input data – modeling – solution" cycle. It is substantiated that solving proactive management tasks is possible by developing an intellectualized software system. This system should act as an "orchestrator" that automates the operation of the mathematical kernel, ensures seamless data exchange, and implements scenario analysis algorithms, transforming AERMOD into a decision support system tool. The objectives of the dissertation research were formulated.

The second chapter substantiates the transition from unstructured, file-oriented workflows to a formalized knowledge-driven model, which is a necessary prerequisite for creating an intellectualized system of proactive management. The analysis proved that traditional approaches to data storage (files, relational databases) are semantically poor, inflexible, and incapable of effectively modeling the complex, heterogeneous entities of the urban ecosystem, making the automation of scenario analysis impossible.

A hybrid architecture for data collection and integration was developed and substantiated. This architecture uses the UESO ontology as a central semantic hub that encapsulates the complexity of heterogeneous sources (AERMET, BPIPFRM, and cadastral data of heat generation objects) and separates knowledge logic from artifact storage (file storage) and high-performance geodata (PostGIS).

The UESO domain ontology was designed. The developed class hierarchy formalizes not only physical objects (sources and receptors) but also key management and computational entities, while ensuring interoperability through inheritance from upper-level ontologies (SWEET, GeoSPARQL).

The foundations of scenario analysis were formalized. Based on the UESO ontology, a formalized definition of two types of proactive analysis was provided:

evaluation and optimization scenarios. The evaluation scenario is formalized as a complete, self-sufficient tuple ensuring the reproducibility of "what-if" experiments. The optimization scenario is formalized as a combinatorial optimization problem, which requires the application of metaheuristic algorithms.

The third chapter develops the mathematical formulation of the combinatorial optimization problem for placing sources of anthropogenic pollution, specifically heat generation objects, taking into account city buildings. Unlike classical geometric placement problems, the proposed approach considers the superposition of concentration fields at receptor points and uses the AERMOD model to calculate the objective function. To guarantee compliance with environmental standards, the objective function, which takes into account the air quality index, is modified by the penalty function method, allowing for the minimization of the maximum risk of exceeding the Maximum Permissible Concentration (MPC) at receptor points. The transition to a discrete search space allowed avoiding the problems of local minima characteristic of continuous functions.

Metaheuristic algorithms (Genetic Algorithm, Discrete Particle Swarm Optimization, Discrete Artificial Bee Colony algorithm) were adapted for solving discrete optimization problems of fixed dimensionality. Three unified operators were developed and implemented in the DEAP framework: `Create_Random_Solution()`, `Random_Swap()`, and `Targeted_Swap()`. This allowed for unifying the software implementation of various heuristics and guaranteeing the validity of solutions.

A method for the intellectualized selection of an optimization strategy based on problem complexity analysis is proposed. The introduction of complexity and scale thresholds allows the system to automatically switch between full enumeration (for trivial tasks), D-PSO (for tactical level tasks), and D-ABC (for strategic tasks of high dimensionality). This ensures a balance between the time to obtain a solution and the guarantee of its optimality under conditions of the high computational cost of calls to the AERMOD module. The method was tested on the example of placing three cogeneration units in Ternopil. The experiment showed that the central part of the microdistrict is environmentally overloaded due to a high background pollution level.

Based on the method, an optimal strategy for the decentralization of critical heat generation infrastructure was determined, proposing a placement that minimizes the impact on residential buildings and educational institutions.

The fourth chapter details the design of a flexible service-oriented architecture (SOA) for the software system, a key element of which is the development of specialized wrapper services. This allowed for the successful integration of the regulatory AERMOD kernel into a modern web environment, ensuring the automation of the full calculation cycle without the need for manual intervention in console processes. Deploying the system in a Kubernetes cluster guarantees the dynamic scaling of computational resources as the load increases.

A data storage subsystem based on a hybrid model was also software-implemented. The combination of a semantic repository (for UESO ontology), a relational database with PostGIS extension (for geometry), and object file storage ensured the realization of the "Single Source of Truth" concept. This approach ensures data synchronization and allows for the automatic validation of modeling scenarios even before the start of resource-intensive calculations.

An interactive web user interface was developed using UX patterns aimed at reducing the cognitive load on the user. Implemented visualization tools (pollution heatmaps, isolines, interactive building layers) allow for the operational interpretation of modeling results and the making of substantiated management decisions without the need for deep analysis of text reports. Load testing results demonstrated a significant increase in performance: thanks to the automation of data preparation and parallelization of calculations, the time to obtain a calculation scenario was reduced from hours (in manual mode) to minutes (by 24 times), making the system suitable for operational and flexible management.

The conclusions formulate the main results of the dissertation work.

Keywords: intelligent software system, software architecture, software, mathematical modeling, mathematical model, nonlinear non-stationary processes, environmental modeling, forecasting, nonlinear problems, identification, numerical methods, point source, computational module, ontology, optimization problem,

optimization methods, objective function, metaheuristic algorithms, critical infrastructure, heat-generating objects, proactive management, urban ecosystem, atmospheric pollution, data uncertainty, decision support system.

LIST OF PUBLISHED PAPERS BY THE TOPIC OF THESIS

Scientific papers in which the main scientific results of the dissertation were published:

1. Tymchyshyn B., Manzhula V. Mathematical and Algorithmic Support of an Intelligent Software System for Proactive Urban Ecosystem Management. Measurement and Computing Technology in Technological Processes. 2025. Issue 84, No. 4, pp. 35–48. (1,2 p.s. / 0,9 p.s.; personal contribution: a mathematical and algorithmic framework for an intelligent system for scenario analysis within proactive urban ecosystem management is proposed. The problem of optimizing the placement of technogenic load sources (in particular, boiler plants and cogeneration units) under conditions of dense urban development is formalized as an NP-hard discrete combinatorial optimization problem. An adaptive optimization method is proposed that automatically selects the most effective algorithm based on complexity and scale thresholds.

DOI: <https://doi.org/10.31891/2219-9365-2025-840-4>

2. Manzhula V. I., Tymchyshyn B. S. Semantic Modeling of Heterogeneous Sources of Harmful Atmospheric Emissions for Proactive Urban Ecosystem Management. Scientific Works of Donetsk National Technical University. Series: “Informatics, Cybernetics and Computer Engineering”. 2025. No. 2 (41), pp. 118–127. (1 p.s. / 0,8 p.s.; personal contribution: A UESO domain ontology has been developed, which formalizes knowledge about heterogeneous urban emission sources and ensures semantic interoperability and logical consistency of data. A service-oriented architecture of the intelligent system has been designed and substantiated, in which the ontology acts as a “single source of truth”).

DOI: <https://doi.org/10.31474/1996-1588-2025-2-41-118-127>

3. Tymchyshyn V., Melnyk A., Dyvak T., Tymchyshyn B., Faifura V.,

Momotiuk Ye., Kostyk B. An Integral Indicator for Evaluating the Efficiency of Specialized Monitoring Systems. *Opto-Electronic Information and Energy Technologies*. 2024. Vol. 48, No. 2, pp. 234–241. (0,8 p.s. / 0,2 p.s.; personal contribution: an analysis of methods for comprehensive evaluation of the effectiveness of implementing intelligent monitoring systems to substantiate the practical value of the proposed system).

DOI: <https://doi.org/10.31649/1681-7893-2024-48-2-234-241>

4. Tymchyshyn V., Tymchyshyn B., Melnyk A., Manzhula V., Faifura V., Romanets I. The System Architecture of the Software for Modeling Harmful Emissions in Soil. *Proceedings International Conference on Advanced Computer Information Technologies Acit*. 2023, P. 58–62. (0,5 p.s. / 0,1 p.s.; personal contribution: the principles of constructing a service-oriented architecture for environmental modeling are substantiated. An approach to decomposing a complex system into independent modules has been validated, which was further transformed into an “Orchestrator” architecture for managing the AERMOD core).

DOI: <https://doi.org/10.1109/ACIT58437.2023.10275416> (Scopus)

Scientific works certifying the approval of the dissertation materials:

5. Tymchyshyn V., Melnyk A., Tymchyshyn B., Faifura V., Mazur I.-S., Honchar L. Data Management Service Architecture of the Software for Modeling Harmful Emissions in Soil. *Proceedings International Conference on Advanced Computer Information Technologies Acit*. 2024, P. 678–682. (0,5 p.s. / 0,1 p.s.; personal contribution: mechanisms for collecting, validating, and routing heterogeneous data have been developed to create a data acquisition subsystem that integrates Weather API data and source parameters).

DOI: <https://doi.org/10.1109/ACIT62333.2024.10712526> (Scopus)

6. Tymchyshyn V., Porplytsya N., Melnyk A., Tymchyshyn B. Software for Modelling the Air Pollution by Vehicles. *CEUR-WS: Proceedings of the International Conference Advanced Computer Information Technologies (ACIT 2018)*. 2018. Vol 2300. pp. 207-210. (0,5 p.s. / 0,1 p.s.; personal contribution: an analysis of methods for modeling background pollution from motor vehicles has been carried out and used

as an input parameter for optimization problems related to the placement of heat-generating facilities).

<https://ceur-ws.org/Vol-2300/Paper50.pdf> (Scopus)

7. Tymchyshyn V. S., Porplytsia N. P., Tymchyshyn B. S. A Software System for Temporal Modeling of Atmospheric Pollution Caused by Motor Vehicle Emissions. Modern Computer Information Technologies: Proceedings of the All-Ukrainian Conference with International Participation ACIT'2017. Ternopil: TNEU, 2017, pp. 233–234. (0,2 p.s. / 0,05 p.s.; personal contribution: an analysis of software tools for modeling atmospheric pollution by harmful emissions has been conducted).

<https://dspace.wunu.edu.ua/items/55add516-6724-4334-8e74-9f53b6d3bbcc>

ЗМІСТ

АНОТАЦІЯ.....	2
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	18
ВСТУП.....	19
РОЗДІЛ 1. КОНЦЕПТУАЛЬНІ ЗАСАДИ ТА АНАЛІЗ ІНСТРУМЕНТАРІЮ ПРОАКТИВНОГО УПРАВЛІННЯ ЕКОСИСТЕМОЮ МІСТА.....	29
1.1. Концепція проактивного управління екосистемою міста: перехід від реагування до попередження.....	30
1.2. Критичний аналіз методів моделювання в контексті задач проактивного управління	33
1.3. Аналіз програмних засобів моделювання розсіювання на основі модуля AERMOD	46
1.4. Постановка задачі дисертаційного дослідження.....	55
Висновки до розділу 1	59
РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛІЗОВАНОЇ СИСТЕМИ НА ОСНОВІ ОНТОЛОГІЧНОГО ПІДХОДУ	61
2.1. Інтеграції гетерогенних даних в задачі екологічного моделювання	62
2.2. Математичні та логічні основи онтологічної моделі Urban Emission Source Ontology	69
2.3. Семантичне моделювання сценаріїв для проактивного управління	76
2.4. Оцінка ефективності онтології UESO на основі її структурної складності	85
Висновки до розділу 2	88
РОЗДІЛ 3. МАТЕМАТИЧНЕ ТА АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛІЗОВАНОЇ СИСТЕМИ.....	90
3.1. Математична модель дискретної оптимізації розміщення джерел забруднення в умовах екологічних обмежень.....	91
3.2. Уніфікація програмної реалізації операторів метаевристичних алгоритмів для задачі дискретної оптимізації.....	98

3.3. Метод інтелектуалізованого вибору стратегії оптимізації залежно від складності оптимізаційної задачі	107
3.4. Апробація методу адаптивного вибору стратегії оптимізації	112
Висновки до розділу 3	118
РОЗДІЛ 4. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛІЗОВАНОЇ СИСТЕМИ ПРОАКТИВНОГО УПРАВЛІННЯ ЕКОСИСТЕМОЮ МІСТА	120
4.1 Архітектура програмної системи для проактивного управління екосистемою міста	121
4.2 Програмна реалізація підсистем управління знаннями та даними.....	136
4.3 Організація графічного інтерфейсу інтелектуалізованої системи для проактивного управління екосистемою міста	150
4.4 Тестування системи проактивного управління екосистемою міста	154
Висновки до розділу 4	166
ВИСНОВКИ	168
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	171
ДОДАТОК А. ФРАГМЕНТ ЛІСТИНГУ КОДУ ОНТОЛОГІЇ UESO V2.0 У ФОРМАТІ TURTLE	184
ДОДАТОК Б. ФРАГМЕНТ ЛІСТИНГУ КОДУ СЕРВІСУ ОБГОРТКИ ДЛЯ ВЗАЄМОДІЇ З МОДУЛЕМ AERMOD.....	194
ДОДАТОК В. СПИСОК ПУБЛІКАЦІЙ ЗА ТЕМОЮ ДИСЕРТАЦІЇ.....	200
ДОДАТОК Д. АКТИ ВПРОВАДЖЕНЬ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ..	202

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – База даних

ГДК – Гранично допустима концентрація

ГІС – Геоінформаційна система

КГУ – Когенераційна установка

ОПР – Особа, що приймає рішення)

НМУ – Несприятливі метеорологічні умови

СЗЗ – Санітарно-захисна зона

СКБД – Система керування базами даних

ТЕЦ – Теплоелектроцентрально

AERMOD – AMS/EPA Regulatory Model

API – Application Programming Interface

CFD – Computational Fluid Dynamics

D-ABC – Discrete Artificial Bee Colony

D-PSO – Discrete Particle Swarm Optimization

DEAP – Distributed Evolutionary Algorithms in Python

DEM – Digital Elevation Model

EPA – Environmental Protection Agency

ETL – Extract, Transform, Load

GA – Genetic Algorithm

OGC – Open Geospatial Consortium

OWL – Web Ontology Language

SPARQL – SPARQL Protocol and RDF Query Language

UESO — Urban Emission Source Ontology

UML — Unified Modeling Language

WKT — Well-Known Text

ВСТУП

Актуальність теми дослідження. Актуальність дисертаційної роботи обумовлена загостренням проблеми забруднення атмосферного повітря в урбанізованих екосистемах, що є одним із ключових викликів сталого розвитку у XXI столітті. Зростання міст, ущільнення забудови та розвиток енергетичної інфраструктури локальних рівнів призводять до формування складних полів антропогенного навантаження, що безпосередньо впливає на здоров'я населення та стан довкілля. Особливої гостроти ця проблема набуває для України в контексті післявоєнної відбудови та стратегічного курсу на децентралізацію енергетичної системи. Розгортання мережі локальних котелень та когенераційних установок, що є ключовим елементом підвищення енергетичної безпеки міст, водночас створює нові виклики для систем екологічного моніторингу та управління. В умовах децентралізації енергетики України, яка передбачає перехід від вразливих централізованих схем до розподіленої генерації (зокрема, впровадження когенераційних установок малої та середньої потужності), важливим стає питання вибору оптимальних локацій для розміщення нових об'єктів техногенного навантаження.

Традиційні підходи до оцінки якості повітря, що базуються на даних стаціонарних постів спостереження, мають реактивний характер. Вони надають лише узагальнену картину, не дозволяючи виокремити внесок конкретних джерел та оперативно прогнозувати наслідки змін у міській інфраструктурі. Це формує нагальну потребу подолання розриву між наявними складними математичними моделями розсіювання, призначеними для вузькоспеціалізованих наукових розрахунків, та практичними потребами міських служб на основі розробки доступного та гнучкого інструменту для підтримки прийняття управлінських рішень.

Фундаментальні засади моделювання дисперсії забруднюючих речовин в атмосфері закладено у працях зарубіжних вчених, таких як F. Pasquill [72], G.A. Briggs [7], D.B. Turner [100], які розробили базові гаусові моделі. Вагомий

внесок у розвиток методології внесла Агенція з охорони довкілля США (U.S. EPA), яка розробила та стандартизувала регуляторні моделі, зокрема AERMOD, що сьогодні є світовим стандартом для оцінки впливу стаціонарних джерел [1, 104, 105, 107]. Розвитку та програмній реалізації цих моделей присвячені роботи таких компаній та дослідницьких груп, як Lakes Environmental Software, ORIS Solutions, що створили потужні, проте комерційно орієнтовані та складні у використанні програмні комплекси.

Дослідженням проблем якості повітря в містах України та адаптацією моделей займалися вітчизняні наукові школи під керівництвом Губернського О.М. [118], Гранберга І.Г. [117], Дідковського М.М. [119]. Значний внесок у розробку національних нормативних методик (ОНД-86) та відповідного програмного забезпечення (ПК "ЕОЛ+", "АТМОСфера") зробили українські науковці та інженери, сфокусувавши зусилля на вирішенні задач нормування викидів та розробки природоохоронної документації [127123].

Проте, незважаючи на значний науковий доробок у сферах математичного моделювання та екології, залишаються поза увагою питання інженерії програмного забезпечення, спрямовані на подолання бар'єру між складністю використання обчислювальних модулів та потребами кінцевих користувачів – спеціалістів міських служб. Існуючі програмні продукти функціонують як «чорна скринька», вимагаючи від користувача глибоких спеціалізованих знань, і не надають гнучких інструментів для інтерактивного прогностичного моделювання, який є критично важливим для переходу проактивного управління екосистемою міст. Крім того, ефективна автоматизація стримується проблемою семантичної гетерогенності даних міської екосистеми. Традиційні файло-орієнтовані підходи не здатні адекватно описувати складні зв'язки між різнорідними джерелами даних (метеорологія, кадастр, забудова), що вимагає застосування онтологічного підходу для створення «єдиного джерела істини».

Особливої ваги тема набуває в контексті повоєнного відновлення України. Масштабна відбудова енергетичної інфраструктури на принципах

«Відбудувати краще» вимагає впровадження інструментів, здатних вирішувати складні комбінаторні задачі оптимізації розміщення нових джерел генерації. Необхідно гарантувати, що нова децентралізована система теплопостачання буде не лише енергоефективною, а й екологічно безпечною для населення. Також, курс на євроінтеграцію вимагає гармонізації методів екологічного моніторингу з директивами ЄС, що передбачає перехід до інформаційних систем управління та використання верифікованих математичних моделей замість застарілих нормативних методик.

Таким чином, розроблення інтелектуалізованої програмної системи, яка б поєднувала точність регуляторних моделей розсіювання шкідливих речовин в атмосфері, семантичну інтероперабельність даних та методи проактивного сценарного аналізу для оперативного прийняття рішень, є актуальним науково-прикладним завданням, що має важливе значення для сталого розвитку та екологічної безпеки міст. Її вирішення дозволить перетворити складне екологічне моделювання на доступний та зрозумілий інструмент для щоденної роботи міських служб, що сприятиме прийняттю науково обґрунтованих рішень у процесах планування, модернізації та розвитку інфраструктури українських міст.

Мета та завдання дисертаційного дослідження. Метою дисертаційної роботи є підвищення ефективності управління екосистемою урбанізованих територій шляхом інтелектуалізації процесів підтримки прийняття рішень на основі розробки онтологічних моделей, методів адаптивної оптимізації та програмної системи для проактивного сценарного аналізу.

Для досягнення поставленої мети необхідно вирішити наступні завдання, що стосуються розробки інформаційного, математичного та програмного забезпечення системи:

- провести критичний аналіз методів та програмних засобів управління екосистемою міста. Обґрунтувати перехід від реактивної моделі управління до проактивної та доцільність використання Гаусових моделей (AERMOD) як

обчислювального ядра, а також виявити архітектурні обмеження існуючих систем, що унеможлиблюють їх автоматизоване використання;

- розробити онтологічну модель для семантичної інтеграції гетерогенних даних міської екосистеми та здійснити математичну формалізацію оцінювальних та оптимізаційних сценаріїв з метою розробки «єдиного джерела істини» та виключення людського фактору з рутинного контуру підготовки даних для моделювання;

- сформулювати задачу розміщення нових джерел забруднення як задачу дискретної комбінаторної оптимізації з мінімізацією цільової функції, яка відображає індекс забруднення у контрольних точках-рецепторах;

- розробити метод адаптивної оптимізації для сценарного аналізу безпечних локацій розміщення нових енергогенеруючих потужностей у щільній міській забудові;

- спроектувати архітектуру та здійснити програмну реалізацію інтелектуалізованої системи проактивного управління екосистемою міста;

- провести експериментальну апробацію запропонованих методів та програмної системи на задачі оптимізації розміщення когенераційних установок у реальному міському середовищі.

Об’єкт дослідження. Процеси інформаційної підтримки прийняття рішень при управлінні екосистемою урбанізованих територій.

Предмет дослідження. Моделі, методи та архітектура інтелектуалізованої програмної системи для проактивного сценарного аналізу та оптимізації розміщення джерел техногенного навантаження.

Методи дослідження. Для досягнення поставленої мети та вирішення завдань у дисертаційній роботі використано комплекс загальнонаукових та спеціальних методів дослідження. Теоретичною та методологічною основою роботи стали: методи системного аналізу та теорії інформаційних систем – для аналізу предметної області, декомпозиції науково-технічної задачі та критичної оцінки існуючих програмних рішень; методи математичного моделювання та теорії оптимізації, зокрема метаввристичні алгоритми, – для

розробки інтелектуального методу оптимального просторового розміщення джерел викидів на основі ітераційного використання зовнішнього обчислювального ядра; принципи об'єктно-орієнтованого аналізу та проєктування з використанням мови UML – для розробки інформаційної моделі предметної області та проєктування сервіс-орієнтованої архітектури програмної системи; методи сучасної інженерії програмного забезпечення – для програмної реалізації ключових компонентів системи, включаючи модуль-інтегратор та веб-інтерфейс; методи обчислювального експерименту та імітаційного моделювання – для проведення апробації розробленого програмного комплексу, перевірки працездатності оптимізаційного алгоритму та оцінки адекватності результатів моделювання на реальних даних.

Комплексне застосування зазначених методів дозволило забезпечити теоретичну обґрунтованість архітектурних рішень, достовірність отриманих експериментальних результатів та практичну значущість розробленого програмного забезпечення.

Наукова новизна одержаних результатів.

Вперше:

- розроблено доменну онтологічну модель для інформаційного забезпечення процесів екологічного моделювання, яка, на відміну від традиційних файлових або реляційних структур, забезпечує семантичну інтеграцію гетерогенних даних міської екосистеми, що дозволило підвищити ефективність та достовірність прийняття рішень на основі сценарного аналізу, усунувши семантичні колізії та мінімізувавши вплив людського фактору при підготовці сценаріїв моделювання;

- запропоновано адаптивний метод оптимізації розміщення джерел техногенного навантаження, який, на відміну від існуючих, базується на уніфікації програмної реалізації еволюційних операторів для різних метаевристик та здійснює керований онтологією вибір стратегії пошуку, що дозволило скоротити час пошуку рішення до прийнятних для оперативної

роботи меж в умовах високої обчислювальної складності оцінки цільової функції;

- запропоновано архітектуру інтелектуалізованої системи проактивного управління екосистемою міста із застосуванням сервіс-орієнтованого підходу для оркестрації обчислювального ядра регуляторної моделі AERMOD, яка, на відміну від існуючих монолітних програмних комплексів, забезпечує повну інкапсуляцію обчислювальної логіки, слабку зв'язність компонентів та надає стандартизований програмний інтерфейс для доступу до функціоналу інтелектуалізованої системи, що забезпечує високу пропускну здатність для виконання обчислювальних процедур адаптивної оптимізації та сценарного моделювання;

- *набула подальшого розвитку:*

- математична модель вибору локацій розміщення джерел теплогенерації, яка, на відміну від існуючих на основі геометричних обмежень у вигляді фіксованих санітарно-захисних зон, базується на формалізації задачі як дискретної комбінаторної оптимізації з цільовою функцією, що задана алгоритмічно на основі чисельної моделі розсіювання, що дозволило здійснити пошук оптимальних рішень за критерієм мінімізації індексу забруднення у точках-рецепторах з урахуванням міської забудови.

Особистий внесок здобувача. Усі результати, викладені в дисертаційній роботі, отримані автором самостійно. У друкованих працях (Додаток В), опублікованих у співавторстві, автору належать такі результати:

[1] – запропоновано математичне та алгоритмічне забезпечення інтелектуалізованої системи для сценарного аналізу в межах проактивного управління екосистемою міста, формалізовано задачу оптимізації розміщення джерел техногенного навантаження (зокрема, котелень та когенераційних установок) в умовах щільної міської забудови, яка NP-складною задачею дискретної комбінаторної оптимізації, запропоновано метод адаптивної оптимізації, який на основі порогів складності та масштабу автоматично обирає найбільш ефективний алгоритм;

[2] – розроблено доменну онтологію UESO, яка формалізує знання про гетерогенні міські джерела викидів та забезпечує семантичну інтероперабельність та логічну несуперечливість даних. Спроектовано та обґрунтовано сервісно-орієнтовану архітектуру інтелектуалізованої системи, де онтологія виступає «єдиним джерелом істини»;

[3] – аналіз методів комплексної оцінки ефективності впровадження інтелектуалізованих систем моніторингу для підтвердження практичної цінності запропонованої системи;

[4] – обґрунтовано принципи побудови сервіс-орієнтованої архітектури для екологічного моделювання, апробовано підхід до розбиття складної системи на незалежні модулі, який трансформувався в архітектуру «Оркестратора» для керування ядром AERMOD;

[5] – розроблено механізми збору, валідації та маршрутизації гетерогенних даних для створення підсистеми збору даних, яка інтегрує Weather API та параметри джерел;

[6] – аналіз методів моделювання фонового забруднення від автотранспорту, що використано як вхідний параметр для оптимізаційних задач розміщення теплогенеруючих об'єктів;

[7] – проведено аналіз програмних засобів для моделювання забруднення атмосфери шкідливими викидами;

Апробація результатів дисертації. Основні положення і результати дисертаційної роботи презентовано на 4 конференціях, зокрема:

- 14th International Conference on Advanced Computer Information Technologies (ACIT), Ceske Budejovice, Czech Republic, 2024;
- 13th International Conference on Advanced Computer Information Technologies (ACIT), Wrocław, Poland, 2023;
- International Conference Advanced Computer Information Technologies (ACIT 2018). Ceske Budejovice, Czech Republic, 2018;
- Всеукраїнська конференція з міжнародною участю «Сучасні комп'ютерні інформаційні технології» ACIT'2017, Тернопіль. 2017.

Публікації. За результатами дисертаційних досліджень опубліковано 7 наукових праць (Додаток В) загальним обсягом 46 сторінок, зокрема 3 статті у фахових наукових виданнях [1, 2, 3], 3 публікацій у матеріалах міжнародних конференцій, матеріали яких є періодичними виданнями та індексуються наукометричною базою Scopus [4, 5, 6] та 1 у матеріалах всеукраїнської конференції з міжнародною участю [7].

Структура та обсяг роботи. Дисертаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел із 128 найменувань та 4 додатки. Загальний обсяг роботи складає 205 сторінок друкарського тексту, з них 170 сторінок основного тексту. Робота містить 56 рисунків і 13 таблиць.

Зв'язок роботи з науковими програмами, планами, темами.

Дисертаційна робота виконувалася в рамках пріоритетного напрямку «Інформаційні та комунікаційні технології» відповідно до закону України «Про пріоритетні напрями розвитку науки і техніки» та пов'язана з науковими дослідженнями, які виконувалися за тематикою кафедри комп'ютерних наук Західноукраїнського національного університету.

Результати дисертаційного дослідження отримано в межах виконання молодіжного науково-дослідного дослідження на тему «Математичне та програмне забезпечення прототипу біогазової установки з підвищеною ефективністю функціонування» (державний реєстраційний номер 0124U000076).

Практичне значення одержаних результатів. Практична значущість одержаних результатів полягає у розробці архітектурних рішень та прототипу інтелектуалізованої системи підтримки прийняття рішень. для спеціалістів міських служб, що не мають глибоких знань у сфері програмування чи екологічного моделювання.

Практична цінність роботи визначається тим, що розроблене програмне та математичне забезпечення уможливорює: науково обґрунтовувати вибір оптимальних локацій для розміщення нових теплогенеруючих об'єктів

(котелень, когенераційних установок) з метою мінімізації їхнього впливу на житлові та рекреаційні зони завдяки використанню розробленого оптимізаційного методу; оперативно оцінювати екологічні наслідки модернізації або перепрофілювання існуючої мережі теплопостачання шляхом проведення інтерактивного прогностичного моделювання; підвищити якість та обґрунтованість управлінських рішень у сферах містобудування та охорони довкілля за рахунок надання візуалізованої та легко інтерпретованої аналітичної інформації. Результати моделювання, отримані за допомогою розробленої системи, можуть бути використані для підготовки вихідних даних при розробці містобудівної документації, схем теплопостачання та при проведенні процедур оцінки впливу на довкілля. Основні архітектурні рішення та програмні компоненти впроваджено у прототипі системи та апробовано на реальних даних мережі теплопостачання м. Тернопіль, що підтверджено відповідним актом впровадження.

Теоретичні та практичні результати дисертаційної роботи використано:

- у комунальному підприємстві теплових мереж «Тернопільміськтеплокомуненерго» Тернопільської міської ради при організації робіт з планування модернізації системи теплопостачання та екологічного моніторингу (акт про впровадження результатів дисертаційної роботи від 22 грудня 2025 р.);

- в НДЧ ЗУНУ при виконанні молодіжного науково-дослідного дослідження на тему «Математичне та програмне забезпечення прототипу біогазової установки з підвищеною ефективністю функціонування» (державний реєстраційний номер 0124U000076) (довідка від 26 грудня 2025 р.);

- в ІТ компанії «Crowdin», у проєктній діяльності компанії при розробці складних веб-орієнтованих систем для сталого розвитку міста, (акт про впровадження результатів дисертаційної роботи від 23 грудня 2025 р.);

- в освітньому процесі Західноукраїнського національного університету на кафедрі комп'ютерних наук при викладанні дисципліни «Архітектура та

проектування програмного забезпечення (акт про впровадження в освітній процес від 22 грудня 2025 р.)

У додатку Д до дисертації наведено акти, що засвідчують використання результатів дисертаційного дослідження.

РОЗДІЛ 1.

КОНЦЕПТУАЛЬНІ ЗАСАДИ ТА АНАЛІЗ ІНСТРУМЕНТАРІЮ ПРОАКТИВНОГО УПРАВЛІННЯ ЕКОСИСТЕМОЮ МІСТА

Сучасний розвиток урбанізованих територій характеризується стрімким зростанням енергоспоживання та ущільненням інфраструктури, що неминуче призводить до загострення екологічних проблем. В умовах децентралізації енергетики, яка передбачає впровадження розподіленої генерації (зокрема, когенераційних установок малої та середньої потужності), критично важливим стає питання вибору оптимальних локацій для розміщення нових об'єктів техногенного навантаження.

Традиційні підходи до управління екологічною безпекою міста здебільшого мають реактивний характер. Вони базуються на системах моніторингу, які фіксують перевищення гранично допустимих концентрацій (ГДК) забруднюючих речовин вже за фактом їх виникнення. Такий підхід («post-factum») дозволяє лише констатувати погіршення стану довкілля, але не надає інструментів для його попередження на етапі планування. Більше того, існуючі методи проектування часто спираються на спрощені геометричні обмеження (санітарно-захисні зони (СЗЗ) фіксованого радіусу) або інтуїтивні рішення, ігноруючи складну фізику розсіювання викидів, вплив метеорологічних умов, рельєфу місцевості та ефект суперпозиції полів від багатьох джерел. Це призводить до прийняття субоптимальних управлінських рішень, які в довгостроковій перспективі створюють ризики для здоров'я населення [127116, 128123].

У даному розділі досліджено сучасний стан та проблематику управління екологічною безпекою урбанізованих територій. Розглянуто необхідність зміни управлінської парадигми з реактивної, яка фіксує порушення постфактум, на проактивну, що дозволяє попереджати екологічні ризики. Проведено критичний аналіз існуючих математичних моделей розсіювання викидів та спеціалізованого програмного забезпечення. Метою розділу є

обґрунтування вибору математичного апарату та визначення функціональних і архітектурних обмежень існуючих інструментальних засобів, що обумовлює необхідність створення нової інтелектуалізованої програмної системи.

Основні результати отримані в даному розділі опубліковані в працях [124].

1.1. Концепція проактивного управління екосистемою міста: перехід від реагування до попередження

Розвиток урбанізованих систем характеризується стрімким зростанням техногенного навантаження, що вимагає докорінного перегляду підходів до забезпечення екологічної безпеки. Традиційні методи управління, що домінували в останні десятиліття, вичерпали свій потенціал ефективності в умовах динамічної зміни клімату та ущільнення міської інфраструктури. Виникає нагальна потреба у зміні управлінської парадигми: переході від реактивної моделі (реагування на інциденти) до проактивної (попередження інцидентів) (рис. 1.1).

Реактивне управління базується на принципі зворотного зв'язку із запізненням. У такій системі керуючий вплив формується лише після фіксації відхилення параметра стану екосистеми від нормативного значення (наприклад, перевищення ГДК шкідливих речовин). Критичним недоліком реактивної моделі є часова затримка, між моментом виникнення пікового забруднення та моментом прийняття рішення проходить час, необхідний для вимірювання, передачі даних, аналізу та реалізації заходів. У цей проміжок часу населення зазнає негативного впливу. Ліквідація екологічних інцидентів або лікування захворювань, спричинених забрудненням, потребує значно більше ресурсів, ніж превентивні заходи. Реактивні системи не враховують динаміку зовнішніх факторів (зміну напрямку вітру, температурні інверсії), що робить управлінські рішення застарілими ще до моменту їх виконання [127116, 128123].

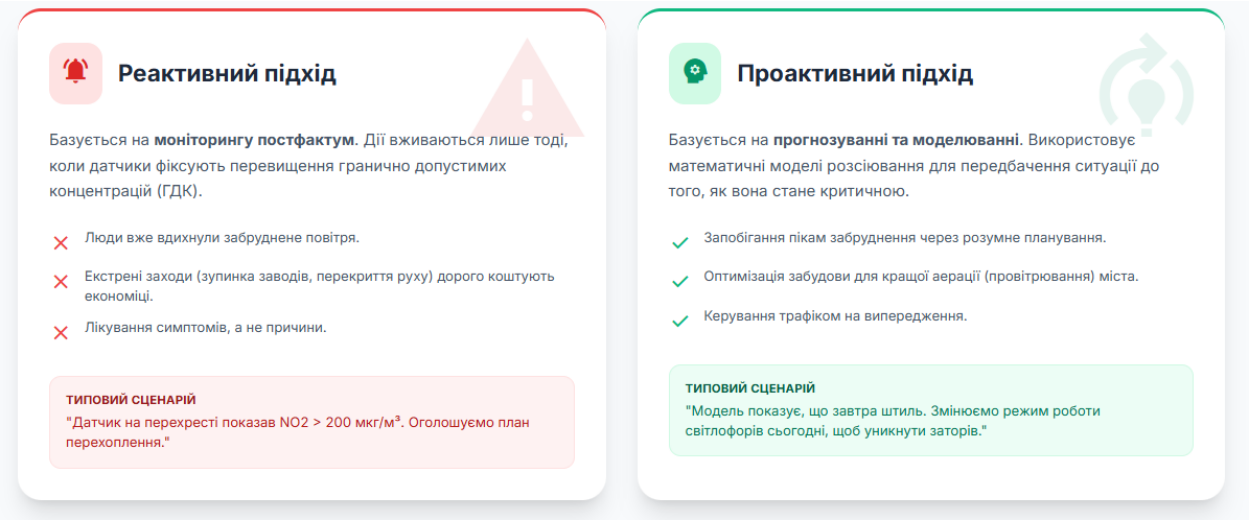


Рис. 1.1. Дві стратегії управління екосистемою міста

Проактивне управління – це стратегія управління екосистемою, яка базується на випереджувальному аналізі майбутніх станів системи. Ключовим інструментом тут має виступати інтелектуалізована програмна система, здатна прогнозувати поведінку об'єкта управління. В основі концепції лежить використання прогнозних моделей, які дозволяють розрахувати вектор стану системи на майбутній горизонт часу та обрати такий керуючий вплив, що не допустить виходу системи за межі допустимих параметрів [22, 38, 6465].

Таблиця 1.1

Переваги проактивного підходу над реактивним

Критерій порівняння	Реактивне управління (As-Is)	Проактивне управління (To-Be)
Момент прийняття рішення	Після інциденту ($t > t_{event}$)	До настання інциденту ($t < t_{event}$)
Інформаційний базис	Історичні дані та поточні виміри	Прогнозні моделі та сценарний аналіз
Характер дій	Коригувальні, штрафні, обмежувальні	Запобіжні, оптимізаційні, планові
Вплив на інфраструктуру	Стресові навантаження (екстрені зупинки)	Планування та плавне регулювання режимів роботи
Екологічна безпека	Мінімізація шкоди	Запобігання шкоді

Реалізація проактивного управління є складним інженерним завданням, що вимагає інтеграції різномірних технологій у єдину програмну систему.

Для здійснення переходу необхідно забезпечити наявність наступних функціональних компонентів (рис. 1.2):

1. Підсистема моніторингу в реальному часі. Перехід неможливий без щільної мережі датчиків, які постачають «сирі» дані про викиди та метеоумови. Проактивна система вимагає високої частоти дискретизації даних для налаштування математичних моделей.

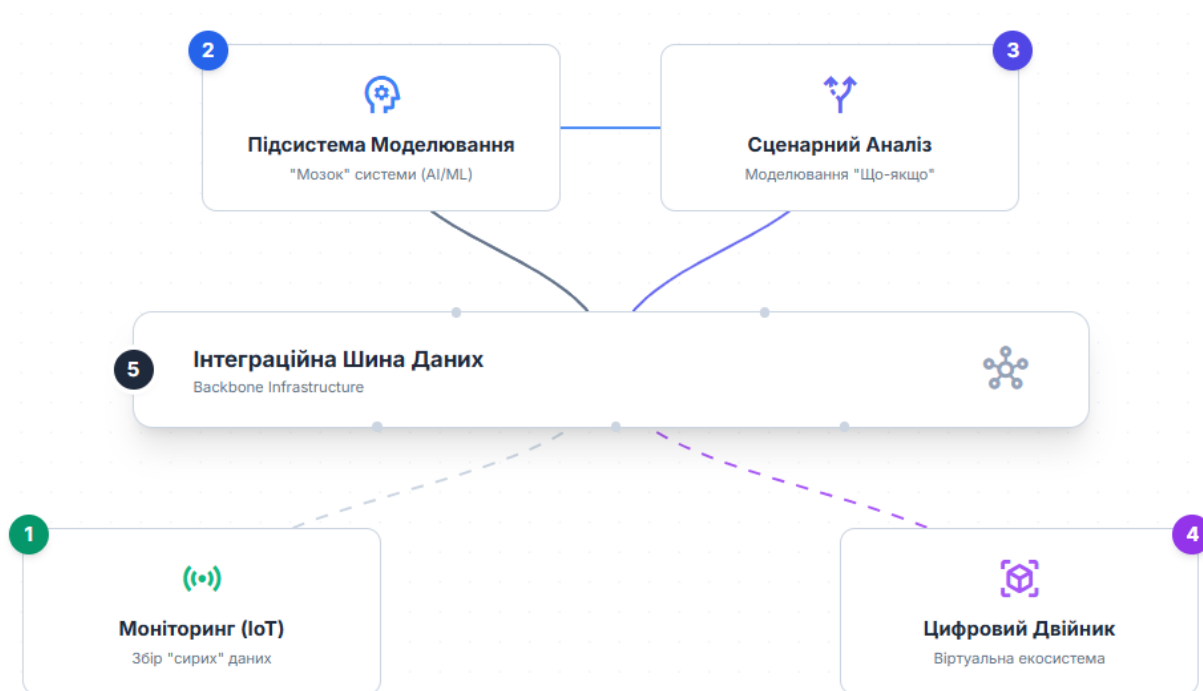


Рис. 1.2. Складові проактивного управління екосистемою міста

2. Підсистема моделювання та прогнозування. Це «мозок» інтелектуалізованої системи. Вона повинна містити: математичні моделі дисперсії: Алгоритми \, що описують фізику розповсюдження домішок, моделі машинного навчання для прогнозування фонових забруднень та метеорологічних умов, які важко описати детермінованими методами.

3. Модуль сценарного аналізу Програмний компонент, що дозволяє моделювати сценарії «що-якщо». Він має автоматично генерувати альтернативні варіанти управлінських рішень (наприклад, перерозподіл навантаження між джерелами тепла) та оцінювати їх ефективність до реального впровадження.

4. Цифровий двійник міста – віртуальна репрезентація міської екосистеми, яка об'єднує статичні дані (карта забудови, рельєф) та динамічні дані (потoki викидів, погода). Це середовище, в якому проактивна система «тестує» свої рішення.

5 Інтеграційна шина даних. Програмна інфраструктура, що забезпечує безшовний обмін даними між джерелами генерації (ТЕЦ, котельні), метеослужбами та муніципальними органами управління.

Отже, перехід від реактивного до проактивного управління є безальтернативним шляхом розвитку екологічної безпеки сучасного міста. Він стає можливим лише завдяки розробці спеціалізованих інтелектуалізованих програмних систем, які поєднують методи математичного моделювання, аналізу великих даних та теорії прийняття рішень. Саме розробці архітектури та алгоритмів такої системи присвячена дана дисертаційна робота.

1.2. Критичний аналіз методів моделювання в контексті задач проактивного управління

Для реалізації функцій прогнозування та сценарного аналізу в інтелектуалізованій системі необхідно обрати математичний апарат, що описує транспорт та дифузію забруднюючих речовин. Вибір конкретного типу моделі для програмної реалізації залежить від масштабу задачі, мети дослідження та наявних обчислювальних ресурсів. На сьогоднішній день розглядають чотири основні класи математичних моделей, що використовуються для аналізу якості атмосферного повітря:

- Гаусові,

- Ейлерові,
- Лагранжеві
- CFD-моделі.

Гаусові моделі – це найбільш поширений тип моделей для регуляторних цілей. Принцип роботи даних моделей базується на моделюванні стаціонарного «факелу» викидів від кожного джерела, при цьому припускають, що розподіл концентрації в перпендикулярному до вітру перерізі підпорядковується нормальному закону (розподілу Гауса). Сферою застосування таких моделей є локальний масштаб (до 50 км). Вони ідеально підходять для оцінки впливу конкретних точкових джерел [3, 37, 46].

Базове рівняння для концентрації $C(x, y, z)$ у точці з координатами (x, y, z) від безперервного джерела викидів має вигляд:

$$C(x, y, z) = \frac{Q}{2\pi u \sigma_y \sigma_z} \exp\left(-\frac{y^2}{2\sigma_y^2}\right) \times \\ \times \left(\exp\left(-\frac{(z-H)^2}{2\sigma_z^2}\right) + \exp\left(-\frac{(z+H)^2}{2\sigma_z^2}\right) \right), \quad (1.1)$$

де,

Q – інтенсивність викиду (г/с);

u – швидкість вітру на висоті викиду (м/с);

$\sigma_y(x), \sigma_z(x)$ – коефіцієнти дисперсії (стандартні відхилення), які є функціями від відстані x та класу стійкості атмосфери;

H – ефективна висота джерела викиду.

Сучасні моделі цього класу (наприклад, AERMOD) використовують не дискретні класи стійкості, а параметри приграничного шару атмосфери (PBL – Planetary Boundary Layer), такі як масштаб довжини Моніна-Обухова (L) та швидкість тертя (u_*), що значно підвищує точність [1104, 108].

Ейлерові моделі – описують процес розсіювання відносно фіксованої системи координат (сітки), що покриває досліджувану територію. Місто розбивається на комірки, і для кожної розв'язується рівняння балансу маси [86, 112, 113].

Розв'язок базуються на чисельному розв'язанні рівняння адвекції-дифузії в частинних похідних:

$$\begin{aligned} \frac{dC(x, y, z, t)}{dt} + v_x \cdot \frac{dC(x, y, z, t)}{dx} + v_y \cdot \frac{dC(x, y, z, t)}{dy} + (v_z) \cdot \frac{dC(x, y, z, t)}{dz} - 3 \\ - D_x \cdot \frac{d^2 C(x, y, z, t)}{d^2 x} - D_y \cdot \frac{d^2 C(x, y, z, t)}{d^2 y} - D_z \cdot \frac{d^2 C(x, y, z, t)}{d^2 z} + \\ + \lambda \cdot C(x, y, z, t) = S(x, y, z, t), \end{aligned} \quad (1.2)$$

де

$C(x, y, z, t)$ – концентрація забруднюючої речовини в точці з координатами x, y, z в момент t , г/м³,

v_x, v_y, v_z – компоненти швидкості вітру в напрямках x, y, z , м/с,

D_x, D_y, D_z – коефіцієнти турбулентної дифузії у відповідних напрямках, [м²/с],

λ – коефіцієнт реакції, [г/с];

$S(x, y, z, t)$ – деяка функція, що описує джерело викидів.

Цей метод ідеальний для моделювання фотохімічних реакцій (смог), але страждає від ефекту «чисельної дифузії» (штучного розмивання шлейфу).

Лагранжеві моделі – це підхід в якому викид розглядається як потік дискретних частинок (віртуальних пакетів повітря), які рухаються за вітром. Траєкторія кожної частинки моделюється окремо як стохастичний процес [3, 37].

Рух частинки описується рівнянням Ланжевена (стохастичне диференціальне рівняння):

$$dX(t) = (u_{mean} + u'(t))dt, \quad (1.3)$$

де швидкісна флуктуація u' часто моделюється як Марковський процес:

$$du' = -\frac{u'}{T_L}dt + \sqrt{\frac{2\sigma_u^2}{T_L}} d\xi, \quad (1.4)$$

де T_L – часовий масштаб Лагранжа, ξ – випадкова величина (білий шум). При цьому концентрація розраховується шляхом підрахунку щільності частинок у заданому об'ємі.

Обчислювальна гідродинаміка (CFD – Computational Fluid Dynamics). CFD-моделі вирішують повну систему рівнянь гідродинаміки для опису потоків повітря, враховуючи складну геометрію будівель (мікрорівень) [62108].

Основою є система рівнянь Нав'є-Стокса (рівняння збереження імпульсу):

$$\rho \left(\frac{\partial u}{\partial t} + u \cdot \nabla u \right) = -\nabla p + \mu \nabla^2 u + f, \quad (1.5)$$

де ρ – густина, p – тиск, μ – в'язкість.

Для турбулентності зазвичай використовують RANS (Reynolds-Averaged Navier-Stokes) з моделями замикання (наприклад, $k - \epsilon$) або LES (Large Eddy Simulation).

На основі проведеного порівняльного аналізу (Таблиця 1.2) можна зробити висновок, що вибір математичного ядра для інтелектуалізованої системи проактивного управління є проблемою пошуку оптимуму між фізичною достовірністю та обчислювальною ефективністю.

Критичний аналіз наведених даних дозволяє виділити наступні ключові аспекти, що визначають непридатність більшості методів для поставленої задачі та переваги Гаусових моделей [1107, 109]:

Таблиця 1.2.

Компаративний аналіз типів моделей розсіювання викидів в атмосфері

Критерій	Гаусові моделі	Ейлерові моделі	Лагранжеві моделі	CFD-моделі
Принцип роботи	Моделює стаціонарний "факел" викидів від кожного джерела на основі статистичного розподілу Гауса.	Ділить територію на 3D-сітку (контрольні об'єми) і розраховує баланс маси та хімічні реакції в кожній комірці.	Відстежує траєкторію руху окремих "частинок" або "порцій" забруднення в нестационарному полі вітру.	Розв'язує фундаментальні рівняння руху рідини (Нав'є-Стокса) для детального моделювання потоків повітря та турбулентності
Масштаб	Локальний (до 50 км). Оптимально для оцінки впливу конкретних джерел (завод, котельня)	Регіональний/Континентальний (100+ та 1000+ км). Оптимально для аналізу фону, смогу, трансграничного переносу.	Від локального до регіонального. Добре працює в умовах складного рельєфу та мінливих метеумов.	Мікромасштаб (від кількох метрів до ~1-2 км). Оптимально для моделювання обтікання будівель, вулиць, стадіонів.
Обчислювальна складність	Низька.	Надзвичайно висока	Висока	Екстремально висока
Вимоги до даних	Помірні (параметри джерел, дані метеополі)	Дуже високі (детальні 3D метеополі, повні емісійні інвентаризації).	Високі (3D метеополі, параметри джерел).	Дуже високі (точна 3D-геометрія всіх об'єктів, граничні умови).
Врахування хімічних процесів	Обмежене (прості реакції першого порядку).	Повноцінне (складні фотохімічні процеси).	Можливе, але ускладнює розрахунки.	Можливе, але значно збільшує складність.
Застосування	Сценарний аналіз «що-якщо», нормування викидів, ОВД, планування.	Аналіз формування озону, кислотних дощів, регіонального фону.	Моделювання аварійних викидів, переносу на великі відстані, пилових бурь.	Детальний аналіз забруднення у «вуличних каньйонах», біля будівель.
Програмні засоби	AERMOD, AERSCREEN, CALINE	CMAQ, WRF-Chem	CALPUFF, HYSPLIT	OpenFOAM, Ansys Fluent

1. Фактор обчислювальної складності. Головною вимогою до системи проактивного управління є здатність працювати в режимі, наближеному до реального часу, або здійснювати багаторазовий сценарний аналіз за короткий проміжок часу. CFD-моделі та Ейлерові моделі, згідно з таблицею, характеризуються високою та «екстремально» високою обчислювальною складністю. Розв'язання рівнянь Нав'є-Стокса для однієї ітерації може займати години, що робить неможливим оперативне перебирання сотень сценаріїв.

Наприклад для вибору оптимального розміщення декількох об'єктів теплогенерації. Гаусові моделі (зокрема AERMOD) мають низьку складність, дозволяючи отримувати результати за секунди або хвилини. Це єдиний клас моделей, який технічно дозволяє реалізувати цикл зворотного зв'язку в системі управління без критичних часових затримок.

2. Масштабність та цільове призначення. Ейлерові моделі орієнтовані на регіональний масштаб і врахування хімічних трансформацій (смог, озон). Для задач управління конкретним теплогенеруючим об'єктом, вплив якого локалізований в межах міста (зона 10-20 км), використання глобальної сітки є надлишковим і ресурсозатратним. CFD-моделі працюють на мікромасштабі («вуличний каньйон»). Хоча вони дають найвищу точність біля конкретних будівель, вони не здатні ефективно оцінити сумарний вплив на все місто через обмеженість розрахункової області. Гаусові моделі займають нішу локального масштабу (до 50 км), що ідеально співпадає з зоною відповідальності муніципальних систем екологічного моніторингу.

3. Вимоги до вхідних даних. Успішне впровадження програмної системи залежить від доступності вхідних даних. Як зазначено в таблиці, CFD вимагає точної 3D-геометрії всіх об'єктів, а Ейлерові/Лагранжеві методи потребують детальних 3D-полів вітру. В умовах реального міста підтримка актуальної бази 3D-геометрії забудови та отримання даних про вітер на різних висотах у кожній точці простору є технічно недосяжним або економічно недоцільним завданням. Гаусові моделі висувають «помірні» вимоги, працюючи зі

стандартними метеоданими та параметрами джерел, які реально отримати з наявних автоматизованих систем.

4. Підтримка сценарного підходу. Таблиця демонструє, що саме Гаусові моделі ідеально підходять для сценарного аналізу «що-якщо». Їх математична структура (суперпозиція полів концентрацій від окремих джерел) дозволяє легко моделювати зміну навантаження окремого котла без необхідності перераховувати всю аеродинаміку міста, як це було б у випадку з CFD.

Отже, незважаючи на те, що CFD та Лагранжеві моделі забезпечують вищу фізичну деталізацію процесів, їх використання в контурі оперативного управління міською екосистемою є неможливим через «прокляття розмірності» та надмірні вимоги до вхідних даних. Таким чином, Гаусові моделі (типу AERMOD) є безальтернативним вибором для розробки інтелектуалізованої програмної системи, оскільки вони забезпечують оптимальний баланс між точністю (достатньою для прийняття регуляторних рішень), швидкістю (необхідною для сценарного моделювання) та вимогами до інформаційного забезпечення.

1.2.2. Структурна організація та особливості функціонування комплексу AERMOD

Для реалізації функцій моделювання розсіювання викидів в інтелектуалізованій системі використовується комплекс AERMOD (AMS/EPA Regulatory Model). Архітектурно цей комплекс являє собою систему взаємопов'язаних модулів (препроцесорів та основного процесора), які працюють послідовно (рис. 1.3.). Така модульність дозволяє гнучко налаштовувати процес моделювання, розділяючи обробку метеоданих, географічних даних та безпосередньо розрахунок концентрацій [1, 104, 105107].

Структурна схема комплексу включає три основні компоненти:

1. Метеорологічний препроцесор AERMET. Призначення – підготовка та адаптація «сирих» метеорологічних даних для використання в дисперсійній

моделі. AERMET обробляє погодинні дані приземних спостережень та дані вертикального зондування атмосфери. Ключовою особливістю є те, що він не просто форматує дані, а розраховує параметри турбулентності планетарного пограничного шару (ППШ), які не вимірюються напряму.

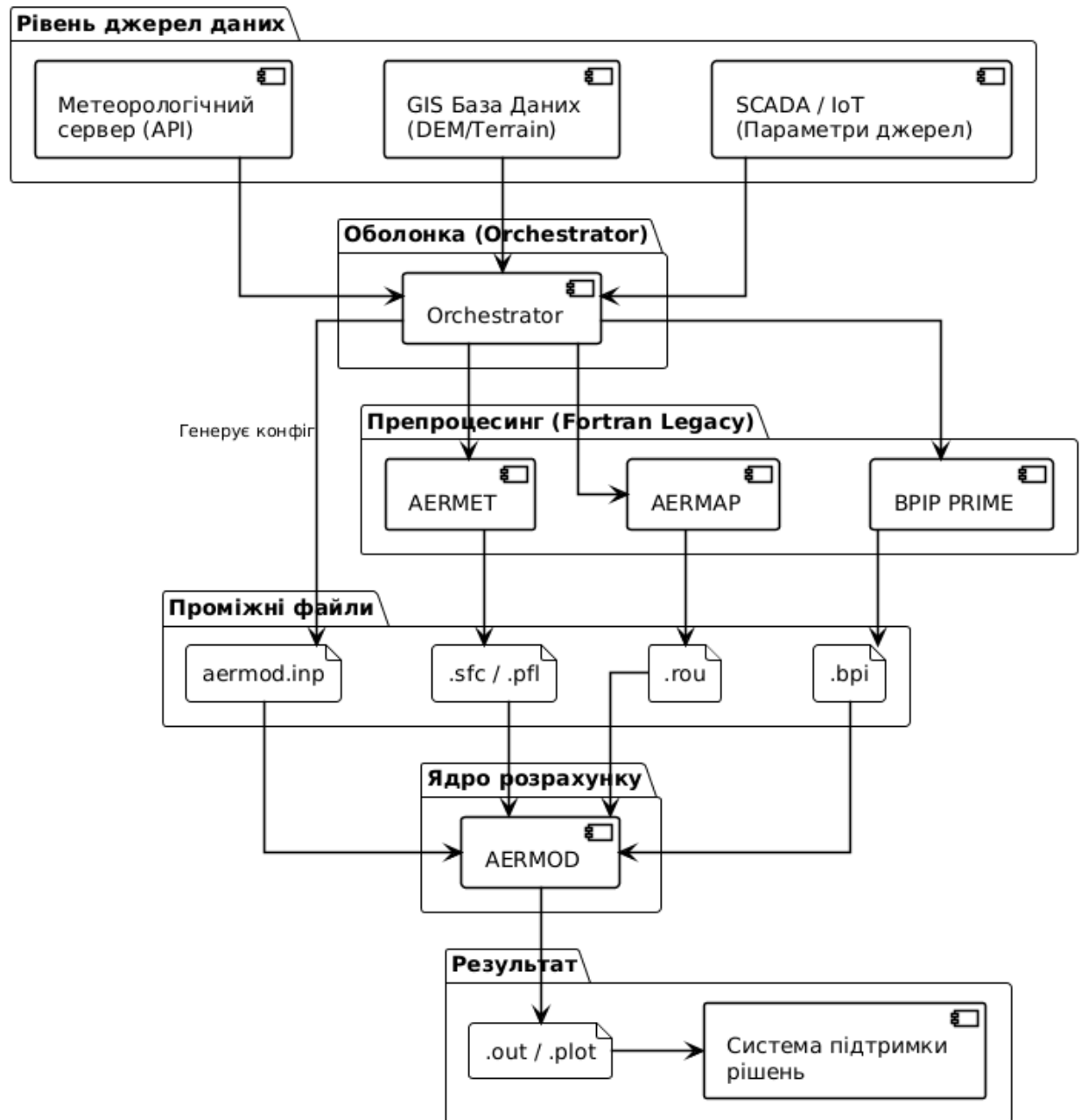


Рис. 1.3. Архітектура обчислювальних модулів AERMOD

Фундаментальною відмінністю AERMOD від застарілих методик є використання теорії подібності для опису структури приземного шару

атмосфери. Замість дискретних класів стійкості Пасквілла-Гіффорда, модель оперує безперервними параметрами турбулентності [104].

Масштаб довжини Моніна-Обухова (L). Цей параметр характеризує співвідношення між механічною (за рахунок тертя вітру) та термічною (за рахунок нагріву/охолодження поверхні) турбулентністю:

$$L = - \frac{u_*^3 \cdot \bar{T}}{k \cdot g \cdot \overline{w'\theta'_0}}, \quad (1.6)$$

де,

u_* – швидкість тертя (м/с);

$\bar{T}$ – середня температура повітря біля поверхні (К);

k – стала фон Кармана (≈ 0.4);

g – прискорення вільного падіння (9.81 м/с²);

$\overline{w'\theta'_0}$ – кінематичний турбулентний потік явного тепла біля поверхні (характеризує теплообмін між землею та повітрям).

Фізичний зміст параметра:

якщо $L > 0$ (і мале значення) – стійка стратифікація (ніч, інверсія, слабке розсіювання);

якщо $L < 0$ (і мале за модулем) – нестійка конвективна атмосфера (день, інтенсивне перемішування);

якщо $|L| \rightarrow \infty$ – нейтральна стратифікація (сильний вітер, похмуро).

Швидкість тертя (u_*). Параметр, що визначає інтенсивність механічної турбулентності, викликані шорсткістю підстиляючої поверхні. Розраховується на основі логарифмічного профілю вітру:

$$u(z) = \frac{u_*}{k} \left[\ln \left(\frac{z}{z_0} \right) - \Psi_m \left(\frac{z}{L} \right) \right]. \quad (1.7)$$

Звідки u_* можна виразити (ітеративно):

$$u_* = \frac{k \cdot u(z_{ref})}{\ln\left(\frac{z_{ref}}{z_0}\right) - \Psi_m\left(\frac{z_{ref}}{L}\right)}, \quad (1.8)$$

де,

$u(z_{ref})$ – швидкість вітру, виміряна на референсній висоті (зазвичай 10 м);

z_0 – параметр шорсткості поверхні (залежить від типу забудови, для міста $z_0 \approx 1.0$ м);

Ψ_m – універсальна функція подібності для імпульсу (поправка на стійкість атмосфери).

Масштаб конвективної швидкості (w_*) визначає інтенсивність теплової турбулентності (важливо для денного часу). Використовується лише в конвективних умовах (день, $L < 0$) для опису вертикального перемішування у всьому шарі змішування:

$$w_* = \left(\frac{g \cdot \overline{w'\theta'_0} \cdot z_{mix}}{\bar{T}} \right)^{\frac{1}{3}}, \quad (1.9)$$

де, z_{mix} – висота шару перемішування, яку AERMET розраховує на основі профілю температури (радіозондування).

У формулі Гауса використовується не геометрична висота труби, а ефективна висота, яка враховує підйом факела за рахунок початкового імпульсу та температури газів:

$$H_{eff} = h_s + \Delta h. \quad (1.10)$$

Підйом факела Δh розраховується за формулами Бріггса. Для гарячих викидів (теплова плавучість домінує):

$$\Delta h = \frac{1.6 \cdot F_b^{\frac{1}{3}} \cdot x^{\frac{2}{3}}}{u(h_s)}, \quad (1.11)$$

де, F_b – параметр потоку плавучості:

$$F_b = g \cdot v_s \cdot \frac{d_s^2}{4} \cdot \frac{T_s - T_a}{T_s}, \quad (1.12)$$

де,

v_s – швидкість виходу газів;

d_s – діаметр гирла труби;

T_s, T_a – температура газів і навколишнього повітря відповідно.

Модуль формує два вихідні файли: файл опису поверхні (.sfc) та профільний файл (.pfl), які слугують вхідними даними для ядра AERMOD.

Використання вказаного математичного апарату дозволяє інтелектуалізованій системі трансформувати технічні параметри роботи теплогенеруючого об'єкта (T_s, v_s , які впливають на F_b і Δh) та прогнози метеоданих (які визначають L та u_*) у поле концентрацій забруднюючих речовин (рис. 1.4.). Це обґрунтовує необхідність включення модулів попередньої обробки даних (AERMET) до архітектури системи.

2. Географічний препроцесор AERMAP. Призначення, якого обробка даних про рельєф місцевості та генерація координат рецепторів (точок розрахунку). Оскільки поширення домішок у місті залежить від рельєфу (обтікання пагорбів, застійні зони в низинах), AERMAP аналізує цифрові моделі рельєфу (DEM – Digital Elevation Model). Для кожного рецептора він розраховує не лише його висоту над рівнем моря, а й так звану «шкалу висоти пагорба». Це дозволяє основній моделі визначати, чи буде шлейф викидів огинати перешкоду, чи перелетить через неї (концепція розділення лінії току).

3. BPIP PRIME (Building Profile Input Program). Для умов міської забудови критично важливим є модуль BPIP PRIME. Він розраховує ефект аеродинамічної тіні. Високі будівлі поблизу джерел викидів (труб котелень) створюють турбулентність, яка може «притиснути» шлейф диму до землі одразу за будівлею, створюючи зони екстремально високої концентрації. Ігнорування цього ефекту в місті призводить до грубих помилок.

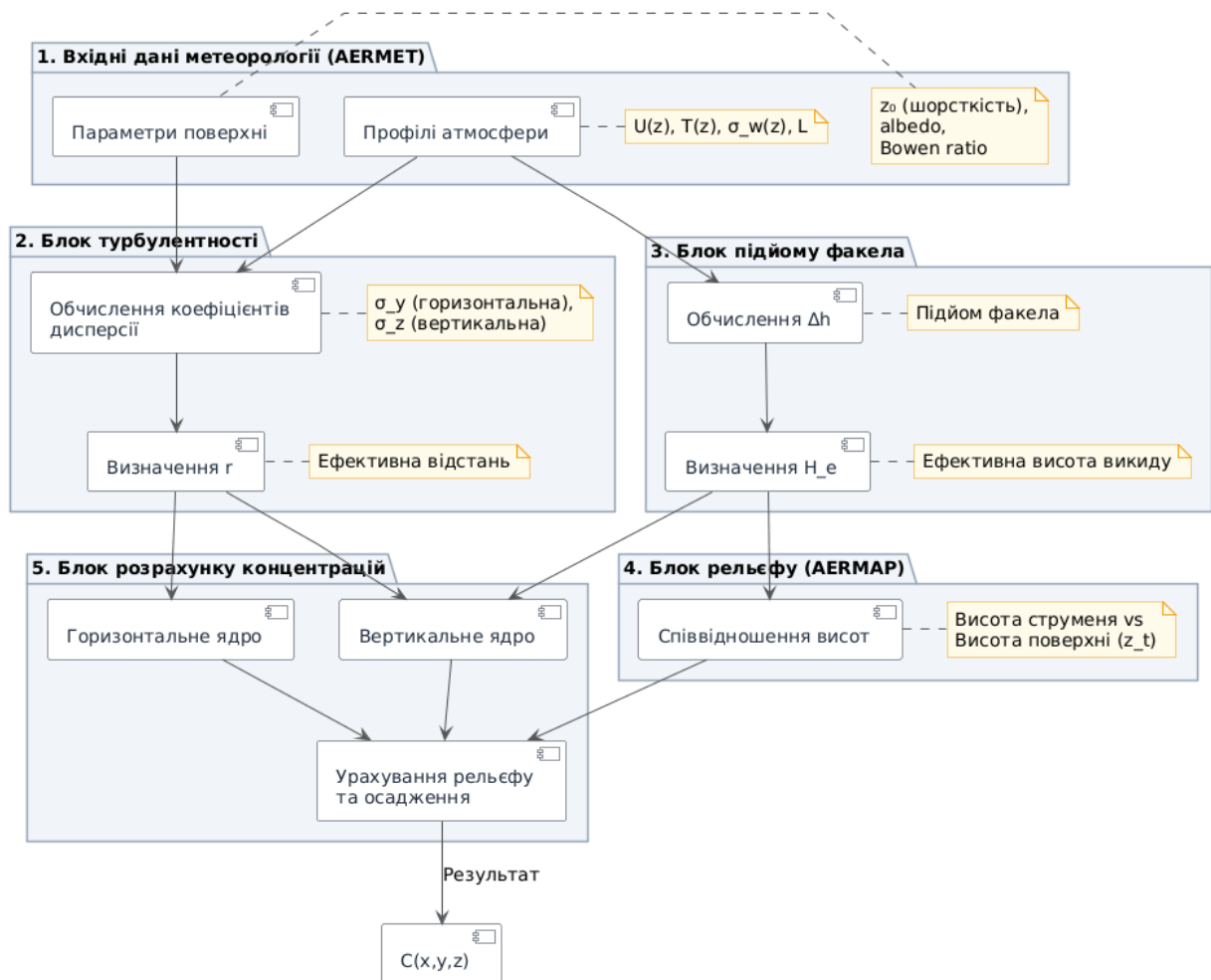


Рис. 1.4. Структурна схема математичних блоків AERMOD

4. Центральний обчислювальний модуль AERMOD здійснює безпосередній розрахунок концентрації забруднюючих речовин. Це ядро системи, яке об'єднує дані від AERMET, AERMAP та параметри джерел викидів. AERMOD використовує гібридний підхід: у стабільних умовах (ніч, інверсія) використовується гаусовий розподіл по вертикалі та горизонталі, у конвективних умовах (сонячний день) використовується бі-гаусовий розподіл для вертикальної компоненти, що дозволяє моделювати ефект відбиття шлейфу від землі та верхньої межі шару перемішування.

Використання AERMOD як компонента (Backend) автоматизованої системи проактивного управління має низку специфічних особливостей, які необхідно врахувати при проектуванні ПЗ:

1. Файловий інтерфейс керування. AERMOD не має API у звичному розумінні (DLL або бібліотеки класів). Керування здійснюється через текстовий файл конфігурації `aermod.inp`. Необхідна автоматична генерація файлу `aermod.inp` на основі обраного сценарію. Система повинна динамічно змінювати параметри джерел та рецепторів, підставляючи нові значення навантаження котелень чи когенераційної установки, не чіпаючи блоки метеорології та рецепторів.

2. Обробка результатів. AERMOD генерує вихідні файли у текстовому форматі (звіти) або двійковому форматі. Для прийняття управлінських рішень системі не потрібен весь масив даних. Необхідно розробити парсер, який зчитує матрицю концентрацій, знаходить локальні максимуми та порівнює їх з ГДК. Час роботи моделі залежить від кількості рецепторів. Для оперативного управління («що буде через годину?») доцільно використовувати не повну сітку міста, а набір «контрольних точок у найбільш вразливих зонах, що скорочує час розрахунку з хвилин до секунд.

3. Вимоги до циклічності. Проактивна система працює в режимі ковзного горизонту. AERMOD розроблений для розрахунку історичних періодів (рік, місяць). Для проактивного управління необхідно запускати модель у режимі короткострокового прогнозу (наприклад, на 24 години вперед), щоразу оновлюючи вхідний метеофайл прогнозними даними. Це вимагає розробки скриптів автоматизації, які керують життєвим циклом запуску модулів: Weather API -> AERMET -> AERMOD -> Decision Parsing.

4. Врахування фонового забруднення. AERMOD моделює лише внесок конкретних джерел, закладених у модель. Для коректної оцінки загальної екологічної ситуації інтелектуалізована система повинна додавати до результатів AERMOD значення фонові концентрації (від транспорту та інших неврахованих джерел), яке може прогнозуватися окремим ML-модулем.

Структурна декомпозиція AERMOD на незалежні модулі дозволяє ефективно інтегрувати його в сучасні програмні архітектури. Використання BPIP PRIME та AERMAP забезпечує необхідну точність для урбанізованого

ландшафту, а текстовий інтерфейс вводу-виводу дозволяє легко автоматизувати процес сценарного аналізу, що є фундаментом проактивного управління.

1.3. Аналіз програмних засобів моделювання розсіювання на основі модуля AERMOD

Оскільки математична модель AERMOD є затвердженим регуляторним стандартом (US EPA) та міжнародним еталоном для розрахунків розсіювання викидів, на світовому ринку екологічного інжинірингу сформувався сегмент спеціалізованого програмного забезпечення, побудованого на її основі. Більшість існуючих рішень являють собою графічні оболонки, розроблені для спрощення взаємодії користувача з оригінальним обчислювальним ядром, написаним на Fortran, а також для візуалізації вхідних даних та результатів моделювання [1, 107].

У цьому підрозділі проведено критичний аналіз найбільш поширених програмних комплексів сімейства AERMOD (зокрема AERMOD View, BREEZE AERMOD та інструменту скринінгу SCREEN View). Метою аналізу є визначення придатності цих рішень для вирішення специфічних задач проактивного управління екосистемою міста.

Оцінювання програмних засобів здійснюється не за критерієм точності моделювання (оскільки всі вони використовують ідентичне математичне ядро EPA), а за архітектурними та функціональними показниками, критичними для автоматизованих систем прийняття рішень. Результати цього аналізу дозволять обґрунтувати необхідність розробки власної архітектурної надбудови над моделлю AERMOD.

1.3.1. Програмний комплекс AERMOD View

AERMOD View – це інтегроване середовище розробки (IDE) для моделювання розсіювання викидів, яке виступає графічною надбудовою (GUI) над консольними модулями US EPA (AERMOD, AERMET, AERMAP, BPIP) [57].

Інтерфейс побудований за модульним принципом, що відображає логіку проходження даних (рис. 1.5). Основні блоки:

– блок джерел – дозволяє графічно розставляти джерела викидів на карті. Підтримує типи джерел: Point (труба), Area (кар'єр, відстійник), Volume (цех), Line (дорога), Open Pit. Дозволяє задавати змінні викиди через табличні коефіцієнти (погодинно, посезонно), але це статичні профілі, а не динамічні дані з датчиків;

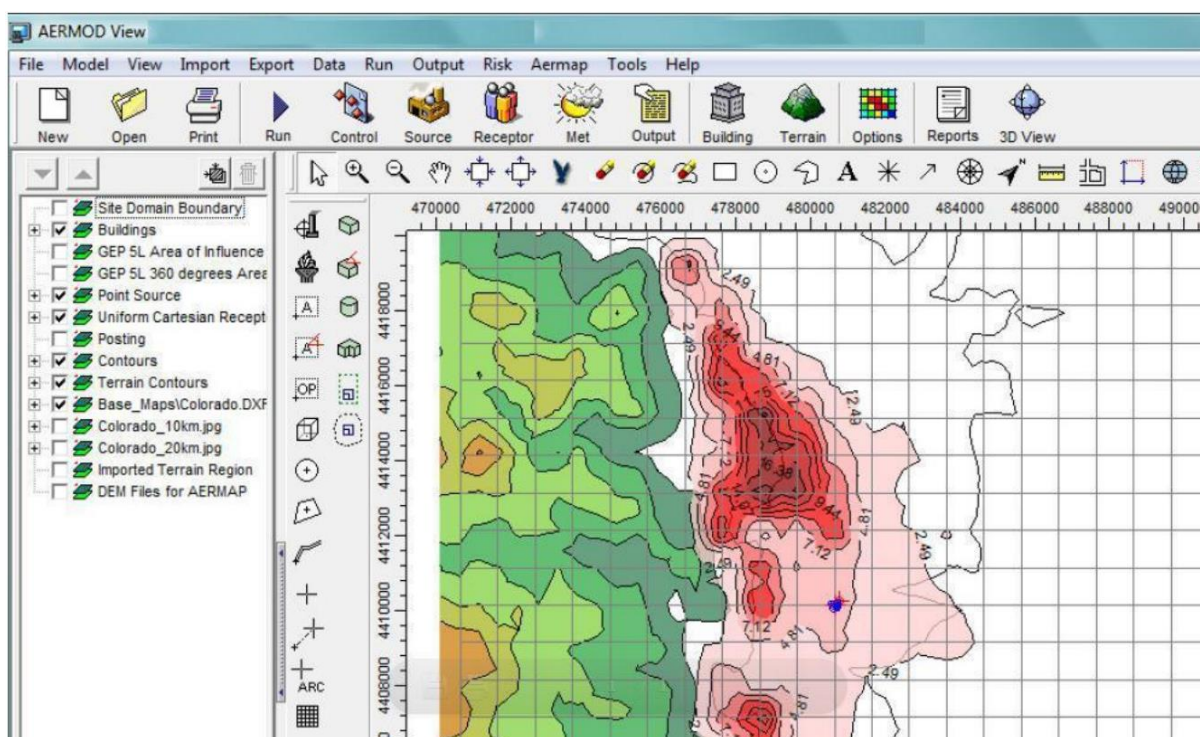


Рис. 1.5. Фрагмент вікна інтерфейсу AERMOD View

– блок рецепторів – Автоматична генерація сіток розрахунку (Uniform Cartesian Grid, Polar Grid). Функція «Plant Boundary» – автоматичне створення

рецепторів по периметру підприємства (санітарно-захисна зона). Імпорт цифрових моделей рельєфу (стандарти DEM/SRTM) для врахування висот;

– метеорологія – вбудовані інструменти AERMET View та AERMINUTE для обробки «сирих» метеоданих. Можливість завантажувати історичні дані з метеостанцій аеропортів (формат ISH/ISHD) безпосередньо з серверів NOAA.

– забудова – графічний редактор для малювання 3D-моделей будівель. Автоматичний запуск BPIP PRIME для розрахунку параметрів аеродинамічної тіні. Це одна з найсильніших сторін продукту, оскільки вручну рахувати проекції будівель на вітер майже неможливо.

AERMOD View використовує власний постпроцесор, який забезпечує побудову кольорових ліній рівних концентрацій накладених на карту (OpenStreetMap, Satellite); перегляд шлейфу викидів у тривимірному просторі відносно будівель та рельєфу; пряму інтеграцію з Google Earth Pro (.kmz файли), що є стандартом для презентації результатів замовникам; автоматичне порівняння отриманих концентрацій з нормативами (NAAQS, EU Standards) та виділення зон перевищення червоним кольором.

Основними перевагами є зручність, оскільки перетворює роботу з сотнями рядків текстового коду (aermod.inp) на кліки мишкою. Знижує поріг входження для еколога; візуалізація вхідних даних, дозволяє побачити помилки (наприклад, труба стоїть посеред будівлі) ще до запуску розрахунку; підтримує формати CAD (.dxf), GIS (.shp) та Excel, що спрощує перенесення даних з проектної документації; містить вбудовані перевірки на логічні помилки (QA/QC) перед запуску моделі.

Незважаючи на потужність, AERMOD View є інструментом для інженерного консалтингу (проектування), а не для оперативного управління. Також можна відмітити відсутність автоматизації. Це класичний десктопний додаток. Він не має API для зовнішнього виклику для підключення його, наприклад до SCADA-системи, щоб вона автоматично запускала розрахунок щогодини. Основні процеси, такі як «Завантажити погоду -> Запустити модель -> Подивитися карту» вимагає участі людини на кожному етапі. Це суперечить

концепції автоматизованої системи прийняття рішень. Інструмент спрямований на обробку архівних метеоданих за минулі роки (для звітів ОВД), а не на роботу з прогнозом погоди на завтра у реальному часі. Висока вартість ліцензії та прив'язка до апаратного забезпечення на основі USB-ключа ускладнюють масштабування системи на рівні муніципалітету.

1.3.2. Програмний комплекс BREEZE AERMOD

BREEZE – це лінійка програмних продуктів для моделювання якості повітря, пожеж та вибухів, розроблена компанією Trinity Consultants (США). Це один із найстаріших та найбільш авторитетних гравців на ринку екологічного консалтингу. Як і AERMOD View, BREEZE AERMOD є потужною графічною оболонкою для стандартних математичних моделей ЕРА, але має свої архітектурні та функціональні особливості, орієнтовані на великі промислові корпорації [74].

Інтерфейс BREEZE суттєво відрізняється від продуктів Lakes Environmental, орієнтуючись на стиль Microsoft Office. Програма використовує вкладки та панелі інструментів, аналогічні MS Word/Excel. Це спрощує навчання персоналу, який звик до офісних пакетів (рис. 1.6).

BREEZE має потужне вбудоване GIS-ядро. Воно дозволяє не просто підкладати карти, а й працювати з векторними шарами на професійному рівні, редагувати атрибути об'єктів безпосередньо в карті. Однією з головних технічних якостей BREEZE є оптимізована версія executable-файлу AERMOD, яка підтримує багатопотоковість. Версія від Trinity розбиває задачу на потоки, що пришвидшує розрахунок у 4-8 разів на сучасних ПК.

BREEZE пропонує кілька унікальних інструментів, яких немає в базовій версії ЕРА:

- BREEZE 3D Analyst – це окремий потужний модуль для пост-обробки та візуалізації. Він дозволяє будувати ізоповерхні – тривимірні «хмари» забруднення певної концентрації;

– Downwash Analyst – інструмент для глибокого аналізу впливу будівель. Він дозволяє візуалізувати вектори вітру навколо будівель і зрозуміти, яка саме стіна створює проблему «затягування» диму;

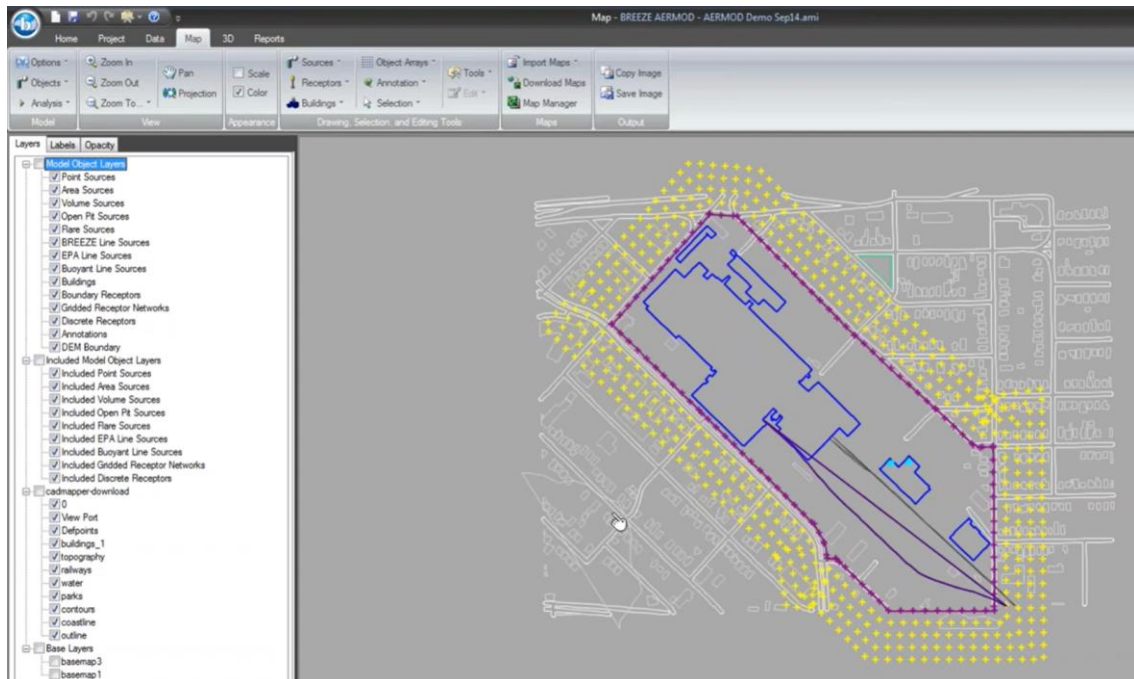


Рис. 1.6. Фрагмент вікна інтерфейсу BREEZE AERMOD

– Sensitivity Analysis – інструмент, що дозволяє автоматично варіювати вхідні параметри (наприклад, швидкість вітру) в певному діапазоні, щоб зрозуміти, наскільки стійким є модельоване рішення.

Завдяки підтримці багатоядерності, BREEZE краще підходить для великих проектів (міста-мільйонники, великі промзони), де кількість джерел вимірюється сотнями. Краща сумісність з базами даних та можливість імпорту даних з корпоративних систем обліку викидів. Оскільки розробник (Trinity Consultants) сам є великим консультантом, ПЗ часто оновлюється під найсвіжіші зміни в законодавстві США.

Незважаючи на «промисловий» статус, BREEZE має ті ж концептуальні обмеження для нашої задачі, що й інші інженерні пакети. Це класична «десктопна» програма, яка встановлюється на потужну робочу станцію інженера. Вона не призначена для роботи як серверний бекенд у хмарі (Cloud-

native). BREEZE підходить для створення звіту раз на рік. За допомогою нього не можливо отримувати прогноз погоди кожні 3 години, автоматично перераховувати дисперсію. Код є закритим, що унеможливорює інтеграцію у свою систему без порушення ліцензії та складних технічних маніпуляцій. Також, це одне з найдорожчих рішень на ринку.

1.3.3. Програмний комплекс SCREEN View

SCREEN View – це графічний інтерфейс для моделі SCREEN3 (або її новішої версії AERSCREEN), розробленої US EPA (рис. 1.7). Це інструмент для експрес-аналізу [58]. На відміну від AERMOD, який намагається змодельовати реальність, SCREEN View намагається знайти теоретичний максимум забруднення. Він перебирає всі можливі комбінації погоди (від штилю до бурі), щоб знайти умови, за яких концентрація буде найвищою («найгірший сценарій»).

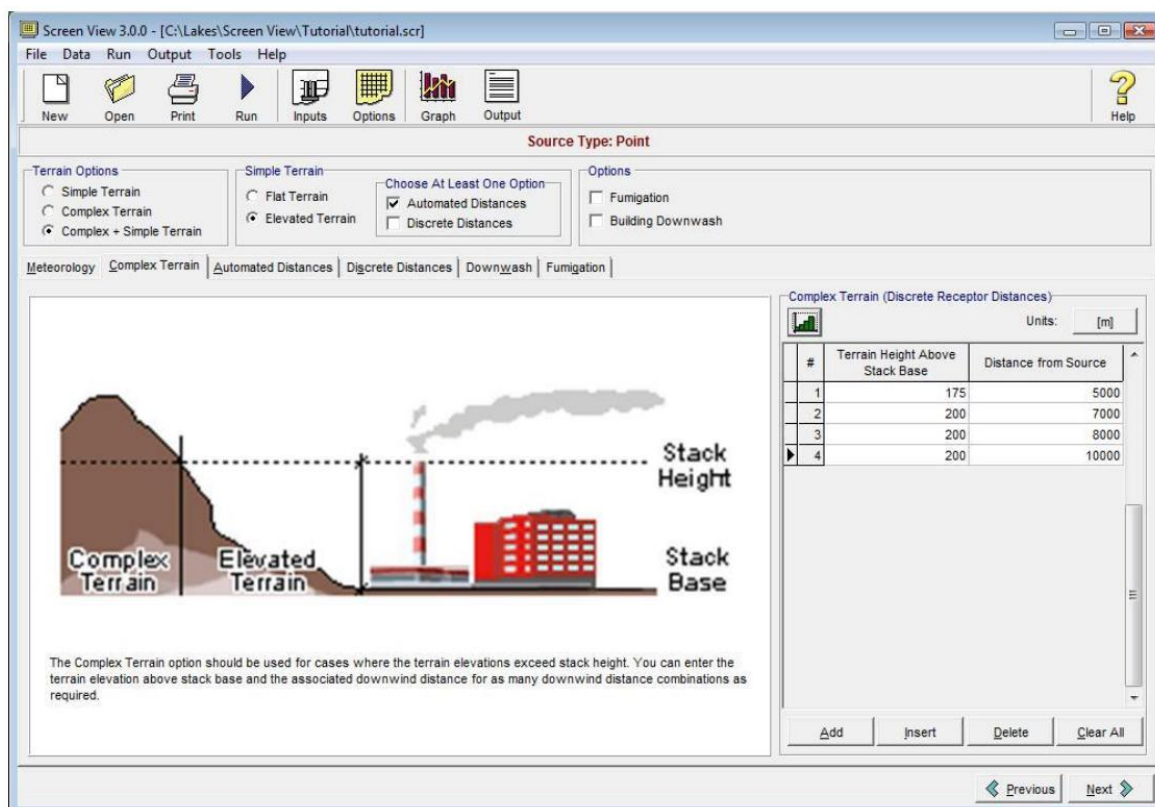


Рис. 1.7. Вікно інтерфейсу SCREEN View

Відсутність реальної метеорології – є головною відмінністю. Програмі не потрібен файл погоди. Вона використовує вбудовану матрицю з 54 стандартних комбінацій метеоумов (швидкість вітру + клас стійкості). Зазвичай моделює лише одне джерело або просту групу. Не призначена для моделювання складних мульти-джерельних систем (наприклад, цілого промвузла). Видає лише одне число – максимальну 1-годинну концентрацію на осі факела. Вона не скаже коли це станеться, вона каже, що це може статися.

До переваг можна віднести швидкість обчислень. Розрахунок виконується за мілісекунди. Якщо SCREEN View показує, що викиди в нормі, то більш точний AERMOD гарантовано покаже норму. Це ідеальний фільтр: «Якщо пройшов скринінг, далі можна не рахувати». Для нашої задачі SCREEN View категорично не підходить як ядро системи через наступні причини: він майже завжди завищує концентрацію (іноді в рази). Розраховує концентрацію лише вздовж осі вітру. Не будує повноцінну карту забруднення міста.

Проведений порівняльний аналіз (Таблиця 1.3) дозволяє зробити висновок про існування критичного технологічного розриву між можливостями сучасного комерційного ПЗ та вимогами до систем проактивного управління екосистемою міста.

Жоден з розглянутих програмних комплексів не може бути використаний як готове рішення для оперативного прийняття управлінських рішень через наступні концептуальні обмеження:

- проблема «Людина в контурі» та відсутність автоматизації. Існуючі засоби реалізовані як інструменти для експертів-проектантів. Цикл роботи з ними передбачає ручну підготовку вхідних файлів, ручний запуск розрахунку та візуальну інтерпретацію результатів оператором. Це спричинює часову затримку між отриманням прогнозу погоди та отриманням результату моделювання в таких системах вимірюється годинами. Нова система повинна повністю автоматизувати роботу обчислювального ядра (AERMOD). Необхідна розробка програмного модуля-оркестратора, який без участі людини конвертує потік даних (Weather API) у вхідні формати моделі,

Таблиця 1.3

Порівняльна характеристика програмних засобів моделювання екосистеми

Критерій порівняння	AERMOD View (Lakes Env.)	BREEZE AERMOD (Trinity)	SCREEN View (Lakes Env.)	Необхідна розробка (ІПС проактивного управління)
Основне призначення	Екологічний консалтинг, ОВД, отримання дозволів.	Корпоративний моніторинг, звітування великих холдингів.	Експрес-скринінг. Оцінка "найгіршого сценарію" (Worst-case analysis).	Оперативне управління та підтримка прийняття рішень у прийнятному часі.
Об'єкт моделювання	Промислові джерела (Точкові, Площинні, Об'ємні).	Промислові джерела (Масштабовані проекти).	Поодинокі джерела або прості групи (спрощена геометрія).	Екосистема міста (Теплогенерація + оцінка фону).
Режим роботи	Офлайн / Проектний. Розрахунок виконується разово за історичний період.	Офлайн / Пакетний. Обробка великих масивів історичних даних.	Офлайн / Миттєвий. Пошук теоретичного максимуму без прив'язки до дати.	Онлайн / Циклічний. Безперервний моніторинг та прогноз на 24/48 год.
Вхідні дані (Метео)	Статичні файли за минулі роки (Historical Data).	Статичні файли + бази даних.	Штучні дані. Матриця з 54 стандартних погодних умов.	Динамічний потік прогнозних даних через API.
Вхідні дані (Джерела)	Вводяться вручну оператором (фіксовані значення).	Імпорт з Excel/CSV або корпоративних БД.	Вводяться вручну (зазвичай максимальна проектна потужність).	Автоматичний збір через IoT/БД (поточне навантаження).
Архітектура ПЗ	Desktop Monolith. "Товстий клієнт" для Windows.	Desktop / Enterprise. Оптимізована багатопоточність.	Desktop Utility. Легка графічна утиліта.	Microservices / Cloud. Модульна, серверна архітектура.
Сценарний аналіз	Ручний. Користувач сам створює копії проекту і змінює параметри.	Напівавтоматичний.	Відсутній. Модель сама перебирає умови, щоб знайти пік.	Повністю автоматичний. Генерація N сценаріїв та вибір оптимального.

Продовження таблиці 1.3.

Результат роботи	Карти, ізоплети, звіт у PDF/Word.	3D-візуалізація, звіти, експорт у GIS.	Одна цифра (Максимальна 1-годинна концентрація)	Управлінське рішення (рекомендація диспетчеру) + прогноз.
Вартість впровадження	Висока (Ліцензія на 1 ПК, \$1600).	Дуже висока (Корпоративна ліцензія).	Середня (дешевше за повний пакет AERMOD).	Гнучка (Open Source ядро + власна розробка).

запускає розрахунок та парсить результати. Це дозволить скоротити цикл реакції до хвилин;

– комерційні пакети орієнтовані на статичний аналіз – перевірку відповідності нормам за історичний період або при найгірших умовах (як у SCREEN View). Вони не пристосовані до пошуку оптимуму в прийнятному часі. Система повинна підтримувати автоматизований сценарний аналіз. Вона має самостійно генерувати множину альтернативних сценаріїв управління, прораховувати їх паралельно і ранжувати за критерієм «Екологічна ефективність / Економічна доцільність»;

– розглянуті аналоги є монолітними десктопними додатками замкненими на локальному ПК. Їх неможливо інтегрувати в єдиний інформаційний простір «Розумного міста». Вони не мають API для отримання даних з IoT-датчиків моніторингу або передачі команд на пульт диспетчера.

Необхідна розробка гнучкої, сервіс-орієнтованої архітектури. Система повинна бути реалізована як набір незалежних модулів, що дозволяє легко масштабувати її, замінювати джерела метеоданих та адаптувати під специфіку будь-якого міста без переписування ядра.

Таким чином, розробка інтелектуалізованої програмної системи є не дублюванням існуючих функцій AERMOD View чи BREEZE, а створенням нового класу програмного забезпечення – системи оперативного екологічного диспетчерування та планування. Наукова та практична цінність роботи полягає у створенні програмної надбудови, яка трансформуватиме перевірене, але «повільне» та «ручне» математичне ядро AERMOD у швидкий, повністю

автоматизований інструмент проактивного управління, здатний працювати в режимі прийнятного часу.

1.4. Постановка задачі дисертаційного дослідження

На основі проведеного аналізу стану проблеми в першому розділі встановлено, що існуючі підходи до управління техногенним навантаженням на екосистему міста мають суттєві обмеження, пов'язані з реактивним характером прийняття рішень та відсутністю автоматизованих засобів сценарного моделювання в прийнятному часі. Виявлено, що наявні програмні продукти (AERMOD View, BREEZE, SCREEN View) є інструментами інженерного проектування, а не оперативного управління, оскільки не підтримують динамічну роботу з потоками даних та потребують ручного керування.

Проведений критичний аналіз існуючих інструментальних засобів (AERMOD View, BREEZE, SCREEN View) дозволив встановити, що жоден з них не може бути використаний як готове рішення для задач оперативного проактивного управління. Незважаючи на спільне математичне ядро (модель AERMOD), існуючі комерційні пакети займають нішу інструментів для статичного проектування та екологічної звітності, тоді як задача дисертаційного дослідження вимагає створення інструменту для динамічного диспетчерування.

Необхідність розробки власної системи обумовлена наявністю критичних концептуальних та технологічних розривів, які неможливо подолати шляхом адаптації існуючого ПЗ. Проблема «Людина в контурі» та необхідність семантичної автоматизації. Існуючі засоби реалізовані як інструменти для експертів-проектантів, де цикл роботи передбачає ручну підготовку вхідних файлів, запуск розрахунку та візуальну інтерпретацію результатів. Це створює неприпустиму для оперативних задач часову затримку, яка вимірюється годинами.

Нова система повинна повністю автоматизувати роботу обчислювального ядра. Однак, проста скриптова автоматизація є недостатньою через гетерогенність даних (AERMET, BPIPPRM, кадастрові дані). Необхідна розробка модуля-оркестратора, що базується на онтологічному підході (UESO). Це дозволить створити «єдине джерело істини», яке забезпечує не лише автоматичну конвертацію даних, а й їх логічну валідацію та семантичну узгодженість без участі людини.

Комерційні пакети орієнтовані на перевірку відповідності нормам за історичний період («чи порушував завод норми минулого року?») або при найгірших умовах (як у SCREEN View). Вони працюють як «калькулятори»: користувач вводить сценарій – система видає результат. Вони не здатні самостійно пропонувати рішення.

Система повинна підтримувати автоматизований оптимізаційний сценарний аналіз. Задача пошуку оптимального розміщення джерел є NP-складною задачею комбінаторної оптимізації. Розробка власного рішення дозволить інтегрувати метаевристичні алгоритми безпосередньо з моделлю розсіювання, реалізуючи адаптивний вибір стратегії пошуку оптимуму, яким керуватиме онтологія. Це дозволяє системі самостійно генерувати та ранжувати сценарії за критерієм «Екологічна безпека / Економічна доцільність», що недоступно в аналогах.

Розглянуті аналоги є монолітними десктопними додатками («товстими клієнтами»), замкненими на локальному ПК. Вони не мають API для отримання даних у реальному часі (Weather API, IoT-моніторинг) або передачі команд на пульт диспетчера. Тому необхідна розробка гнучкої, сервіс-орієнтованої архітектури. Система повинна реалізовуватись як набір незалежних модулів (мікросервісів), що взаємодіють через онтологічний хаб. Це дозволяє масштабувати систему, замінювати джерела даних та адаптувати її під специфіку будь-якого міста без переписування ядра, а також використовувати гібридну модель зберігання даних (PostGIS + Файлове сховище + Онтологія) для ефективної роботи з геопросторовою інформацією.

Таким чином, розробка інтелектуалізованої програмної системи є не дублюванням функцій AERMOD View чи BREEZE, а створенням нового класу програмного забезпечення – системи проактивного управління екосистемою міста на основі сценарного аналізу.

Метою роботи є підвищення ефективності управління екосистемою урбанізованих територій шляхом розробки моделей, методів та інтелектуалізованої програмної системи для проактивного сценарного аналізу та оптимізації розміщення джерел техногенного навантаження.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести критичний аналіз методів та програмних засобів управління якістю атмосферного повітря. Обґрунтувати перехід від реактивної моделі управління до проактивної та доцільність використання Гаусових моделей (AERMOD) як обчислювального ядра, а також виявити архітектурні обмеження існуючих систем, що унеможливають їх автоматизоване використання;
- розробити онтологічну модель (UESO) для семантичної інтеграції гетерогенних даних міської екосистеми та здійснити математичну формалізацію оцінювальних та оптимізаційних сценаріїв з метою розробки «єдиного джерела істини» та виключення людського фактору з рутинного контуру підготовки даних для моделювання;
- сформулювати задачу розміщення нових джерел забруднення як задачу дискретної комбінаторної оптимізації з цільовою функцією типу «чорна скринька» (на базі AERMOD). Адаптувати метаевристичні алгоритми для роботи в дискретному просторі рішень з використанням розроблених уніфікованих операторів.
- розробити метод інтелектуалізованого вибору стратегії оптимізації, який, на основі аналізу складності простору пошуку та обчислювальної вартості моделювання, автоматично адаптує алгоритм пошуку (від повного перебору до метаевристик) для забезпечення балансу між часом реакції та якістю рішення.

- розробити архітектуру та програмну реалізацію інтелектуалізованої системи проактивного управління
- провести експериментальну апробацію на задачі оптимізації розміщення когенераційних установок у реальному міському середовищі.

Отже, сучасна практика управління екосистемою міст страждає від реактивності та низької оперативності через «часткову» автоматизацію та гетерогенність даних. Інженерні розрахунки розсіювання викидів часто виконуються вручну та ізольовано від задач міського планування, що унеможлиблює оперативне реагування на зміни. Дослідження спрямовано на вирішення цієї проблеми шляхом створення інтелектуалізованої системи проактивного управління.

Практична цінність роботи полягає у створенні програмного інструментарію, який трансформує процес прийняття рішень: замість пасивної фіксації забруднення система дозволяє моделювати майбутнє («що-якщо») та автоматично знаходити оптимальні конфігурації розміщення теплогенеруючих об'єктів. Використання семантичного моделювання забезпечує автоматичну інтеграцію різнорідних даних (метеопрогноз, кадастр, параметри котлів) у «єдине джерело істини», а методи сценарного аналізу дозволяють гарантувати дотримання санітарних норм ще на етапі планування інфраструктури та для його модернізації.

В умовах масового руйнування енергетичної інфраструктури, стратегія відбудови повинна базуватися на відмові від застарілих централізованих радянських систем на користь децентралізованої розподіленої генерації (когенераційні установки, міні-ТЕЦ). Дослідження також має значення для цього процесу, оскільки: забезпечує принцип «Відбудувати краще», дозволяючи ще на етапі проєктування знайти такі локації для нових енергооб'єктів, які мінімізують вплив на здоров'я населення, уникаючи помилок минулого планування; підвищує енергетичну стійкість, дозволяючи безпечно інтегрувати велику кількість малих джерел генерації (КГУ) в межі

житлових масивів, що є необхідним для енергонезалежності громад, гарантуючи при цьому дотримання санітарних норм завдяки сценарному моделюванню.

Також, набуття Україною статусу кандидата в члени ЄС вимагає імплементації європейських екологічних директив (зокрема Директиви 2008/50/ЄС про якість атмосферного повітря) та долучення до «Європейського зеленого курсу». Використання міжнародно визнаного ядра AERMOD замість застарілих методик ОНД-86 забезпечує верифікованість та визнання результатів екологічного моделювання європейськими партнерами та інвесторами. Запропонований онтологічний підхід створює прозорий цифровий слід кожного управлінського рішення, що відповідає європейським стандартам публічного управління та боротьби з корупцією при видачі дозволів на викиди.

Висновки до розділу 1

У першому розділі виконано теоретичне узагальнення та обґрунтування підходів до створення інтелектуалізованої програмної системи проактивного управління екосистемою міста.

1. Встановлено, що традиційне реактивне управління («post-factum») є неефективним в умовах щільної забудови та динамічних змін метеоумов, оскільки характеризується часовою затримкою між інцидентом та реакцією. Доведено необхідність переходу до проактивної моделі, яка базується на сценарному аналізі станів системи та запобіганні екологічним ризикам.

2. Проведено компаративний аналіз класів математичних моделей розсіювання домішок в атмосфері. Визначено, що для задач оперативного сценарного аналізу, який вимагає багаторазових розрахунків у прийнятному часі, оптимальними є Гаусові моделі, зокрема обчислювальне ядро AERMOD. Вони забезпечують найкращий баланс між точністю та обчислювальною

складністю, на відміну від CFD та Ейлерових моделей, які є надмірно ресурсоемними для оперативних задач.

3. Проаналізовано структурну організацію комплексу AERMOD та його складових (AERMET, AERMAP, BPIP PRIME). Показано, що його модульність та файловий інтерфейс дозволяють інтегрувати його як обчислювальне ядро (Backend) в автоматизовані системи управління.

4. Здійснено критичний огляд існуючих програмних засобів: AERMOD View, BREEZE AERMOD, SCREEN View. Виявлено критичний технологічний розрив: існуючі пакети є монолітними десктопними додатками, орієнтованими на статичний проектний аналіз та ручне введення даних. Вони не мають API для інтеграції з зовнішніми ресурсами та не підтримують автоматизований цикл «вхідні дані – моделювання – рішення».

5. Обґрунтовано, що вирішення задач проактивного управління можливе лише шляхом розроблення інтелектуалізованої програмної системи. Вона має виступати як «оркестратор», що автоматизує роботу математичного ядра, забезпечує безшовний обмін даними та реалізує алгоритми сценарного аналізу, перетворюючи AERMOD з інструменту проектування на інструмент оперативного диспетчерування. Здійснено постановку завдань дисертаційного дослідження.

РОЗДІЛ 2.

ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛІЗОВАНОЇ СИСТЕМИ НА ОСНОВІ ОНТОЛОГІЧНОГО ПІДХОДУ

Проведений у попередньому розділі аналіз дозволив встановити, що ключовим бар'єром на шляху до створення ефективних систем проактивного управління є не стільки відсутність обчислювальних потужностей, скільки фундаментальна проблема інформаційного забезпечення. Існуючі інструменти екологічного моделювання (напр., AERMOD) історично покладаються на файло-орієнтовані робочі процеси. Як було показано, такий підхід є фрагментованим, семантично бідним та негнучким. Він не здатний адекватно описувати складні, гетерогенні сутності міської екосистеми та їхні взаємозв'язки, що унеможлиблює автоматизацію робочих процесів, перевірку логічної несуперечливості даних та проведення ітеративного сценарного аналізу.

Метою даного розділу є вирішення цієї проблеми шляхом розробки та обґрунтування нової моделі інформаційного забезпечення, що базується на методології семантичного моделювання. Замість створення пасивного сховища даних (як у традиційних реляційних БД), пропонується розробка активного, керованого знаннями «єдиного джерела істини» (Single Source of Truth) у вигляді доменної онтології. Такий підхід дозволяє формально моделювати не просто дані, а знання про предметну область: її сутності, таксономічну ієрархію, атрибути та складні логічні взаємозв'язки. Це є необхідною концептуальною та технічною передумовою для створення будь-якої інтелектуалізованої системи підтримки прийняття рішень.

У розділі детально аналізується проблема інтеграції гетерогенних даних. Розглядаються різні за своєю природою джерела (AERMET, BIPPRM, AERMAP) та формати (текст, бінарні файли, GeoTIFF, CSV), що обґрунтовує необхідність гібридної архітектури збору даних, де онтологія виступає центральним семантичним хабом.

Запропоновано математичні та логічні основи онтології UESO для формалізації джерел, інфраструктурних елементів та управлінських сутностей, а також обґрунтовує інтеграцію з онтологіями верхнього рівня (SWEET, GeoSPARQL) [92, 93]. На основі розробленої онтології формалізуються математичні кортежі для двох фундаментальних типів аналізу: оцінювального та оптимізаційного

Проводиться кількісна оцінка ефективності розробленої онтології UESO шляхом розрахунку стандартизованих онтологічних метрик, яка об'єктивно підтверджує її семантичну насиченість та структурну збалансованість.

Основні результати отримані в другому розділі опубліковано в працях [115, 124].

2.1. Інтеграції гетерогенних даних в задачі екологічного моделювання

Проведений в попередньому розділі аналіз довів, що традиційні підходи до зберігання даних (файли, реляційні БД) є семантично бідними, негнучкими та нездатними ефективно моделювати складні, гетерогенні сутності міської екосистеми, що унеможлиблює автоматизацію сценарного аналізу [67, 93]. Оскільки, ефективність та достовірність будь-якої інтелектуалізованої системи безпосередньо залежить не лише від досконалості обчислювальних алгоритмів, але й від якості, повноти та семантичної узгодженості вхідних даних. Створення «єдиного джерела істини» є однією із умов такої ефективності, але його реалізація стикається з фундаментальною проблемою гетерогенності даних. Ця проблема має три основні виміри [19, 33, 34, 39]:

- гетерогенність екосистеми обчислювального модуля;
- гетерогенність джерел та форматів даних;
- гетерогенність семантики.

Обраний нами обчислювальний модуль для моделювання, AERMOD, не є монолітним застосунком. Це екосистема з кількох окремих програм

(препроцесорів), кожна з яких вимагає власних, унікальних за типом та форматом вхідних даних.

Зокрема, метеорологічний препроцесор AERMET вимагає сирих погодинних поверхневих даних та даних аерологічного зондування у специфічних текстових форматах (.dat), а на виході генерує власні бінарні файли (.SFC, .PFL). В свою чергу, препроцесор рельєфу AERMAP вимагає цифрових моделей рельєфу (DEM), зазвичай у растрових ГІС-форматах (напр., GeoTIFF). Також, препроцесор забудови BPIPRM вимагає векторних даних про контури та висоту будівель (напр., Shapefile, CSV, DWG).

Основний обчислювальний модуль AERMOD вимагає результати роботи всіх цих препроцесорів, а також детальний опис джерел та рецепторів у власному, строгому позиційному текстовому форматі .INP.

Гетерогенність джерел та форматів даних обумовлюється тим, що дані надходять з абсолютно різних, не пов'язаних між собою джерел: дані про рельєф – з Держгеокадастру, метеорологічні дані – з Держгідромету, дані про забудову – з місцевого управління архітектури або таких зовнішніх ресурсів, як OpenStreetMap, а кадастр джерел – з файлів самого підприємства (напр., CSV або Excel). Як наслідок, система має одночасно оперувати щонайменше п'ятьма різними форматами даних: текстовими файлами, бінарними файлами, растровими ГІС-даними, векторними ГІС-даними та таблицями.

Ключовою проблемою є гетерогенність семантики. Це найглибша проблема. Оскільки, традиційні підходи до зберігання даних (файли або реляційні БД) фіксують дані, але не їхнє значення [89].

Файло-орієнтований підхід (традиційна практика екологів) є крихким та невідтворюваним. Він покладається на те, що інженер вручну коректно збереже всі файли у потрібних теках, правильно їх назве та без помилок впише шляхи до них у головний керуючий файл. Будь-яка зміна (напр., оновлення метео-файлу) вимагає ручного редагування, що є джерелом помилок (рис. 2.1).

Реляційний підхід (напр., SQL-база даних) вирішує проблему структурованого зберігання, але страждає від семантичної бідності та

жорсткості схеми. База даних «знає», що в таблиці boilers є колонка `stack_height` типу `decimal(8,2)`. В той же час, вона не «розуміє», що це та сама сутність, яку AERMOD називає STKHEI, а онтологія SWEET – PhysicalProperty. Вона не може логічно перевірити, що BoilerHouse (котельня) повинна мати цей параметр, а AreaSource (площинне джерело) – ні. Ця логіка закладається в код застосунку, роблячи саму базу даних «пасивною».

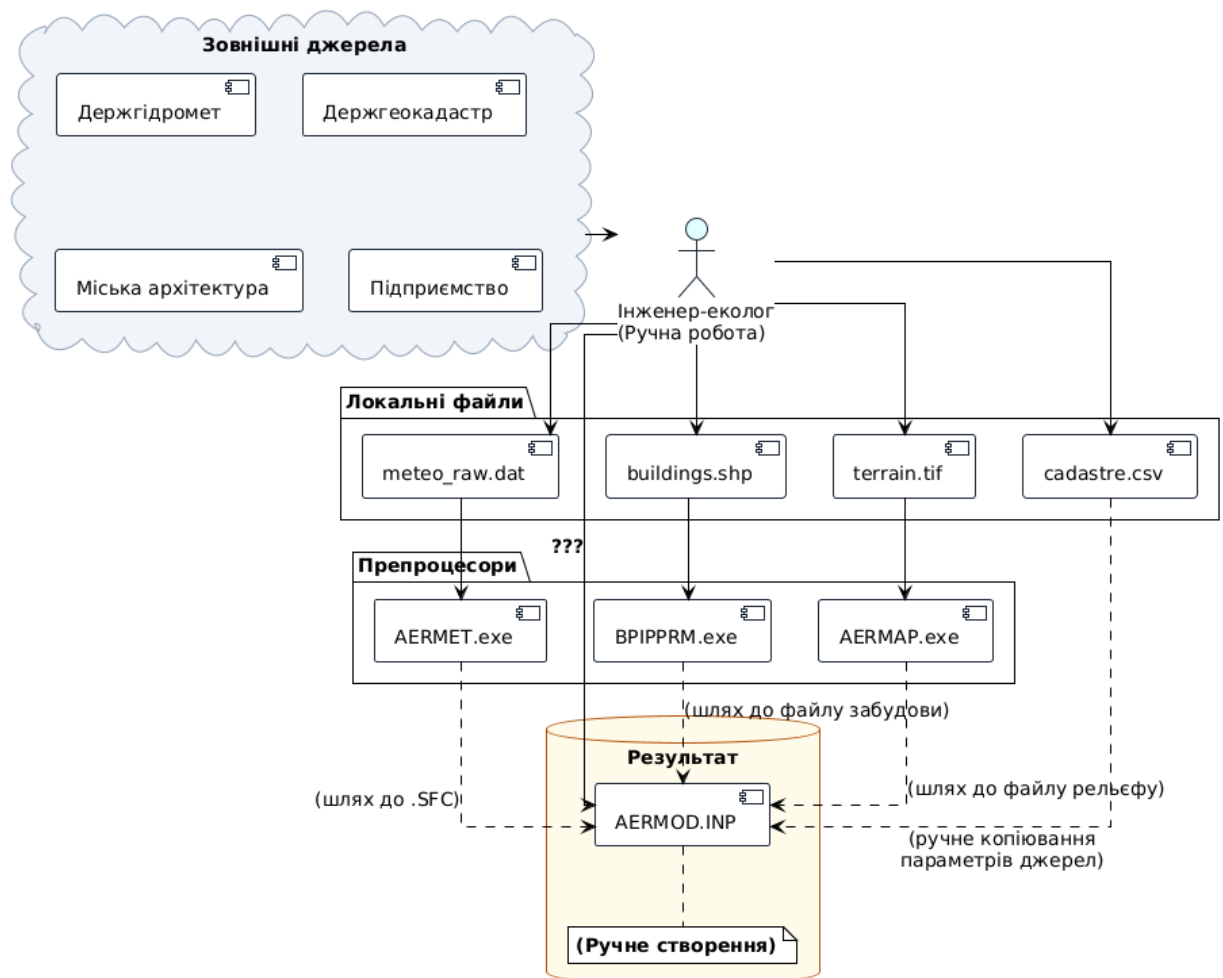


Рис. 2.1. Проблема гетерогенності даних в традиційному підході

Для подолання цих проблем та реалізації гнучкої, інтелектуалізованої системи її інформаційне забезпечення повинно відповідати наступним вимогам:

- семантична виразність. Модель даних повинна зберігати не лише значення (30.0), але й знання («Висота труби» є «Фізичною властивістю»);
- логічна валідація. Модель повинна мати вбудовані механізми перевірки несуперечливості (напр., «Джерело не може бути одночасно і точковим і поверхневим»);
- інтероперабельність. Модель повинна використовувати загальновизнані стандарти (напр., W3C, OGC) для легкої інтеграції з іншими міськими системами (транспорт, містобудування тощо);
- гнучкість та розширюваність. Модель повинна дозволяти легке додавання нових типів джерел (напр., EcoTransport) без необхідності руйнування існуючої структури.

Традиційні файло-орієнтовані та реляційні підходи не задовольняють цим вимогам. Вони створюють семантично розрізнені «набори» даних (data silos). Для створення справді інтегрованої системи, здатної автоматизувати складні робочі процеси (рис. 2.1), необхідний перехід до моделі, що керується знаннями. Це обґрунтовує розробку доменної онтології (UESO) як центрального семантичного інтеграційного хабу, що буде детально описано в наступному підрозділі.

Для вирішення фундаментальної проблеми гетерогенності даних та створення надійного «єдиного джерела істини» (Single Source of Truth), була розроблена спеціалізована архітектурна схема збору та інтеграції, представлена на рис. 2.2. Ця архітектура базується на двох ключових принципах: інкапсуляції (через препроцесорні сервіси) та централізації семантики (через онтологічний хаб).

Верхній рівень (“Зовнішні джерела”) представляє хаотичне середовище вхідних даних у їхніх “сирих” форматах (CSV, API, файли DEM, Shapefile, метеорологічні .dat файли). Пряма взаємодія ядра системи з цим середовищем призвела б до крихкості та надзвичайної складності коду.

Для запобігання цьому введено проміжний шар – “Препроцесорні Сервіси (ETL)”. Кожен сервіс у цьому шарі є незалежним мікросервісом-

адаптером, який реалізує принцип інкапсуляції: він “ховає” всю складність роботи з одним конкретним типом гетерогенних даних і надає єдиний, стандартизований інтерфейс для ядра системи.

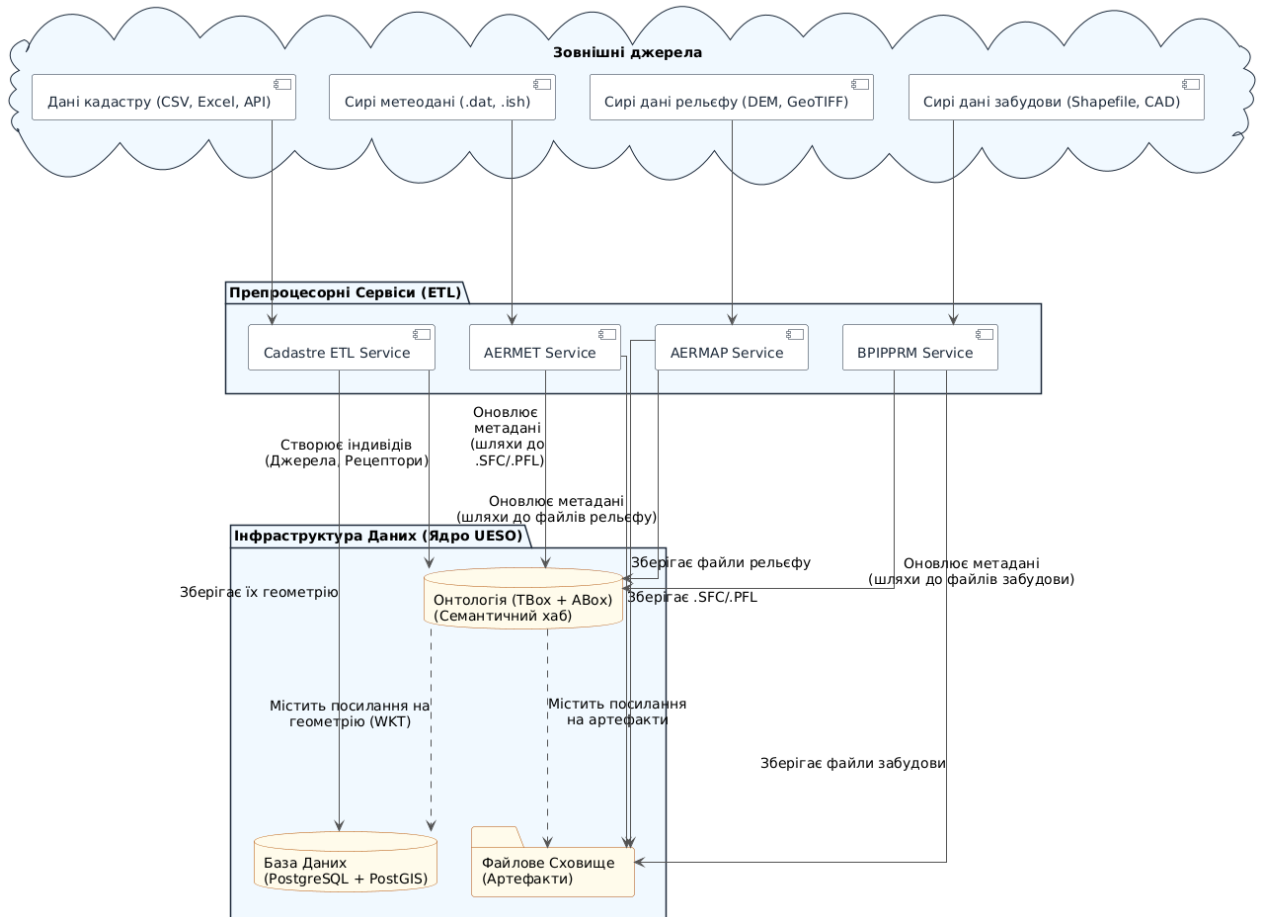


Рис. 2.2. Архітектурна схема збору та інтеграції гетерогенних даних

AERMET Service спеціалізується на обробці «сирих» метеорологічних даних. Він автоматично завантажує їх, запускає обчислювальне ядро AERMET.exe і зберігає результуючі бінарні артефакти (.SFC/.PFL). AERMAP Service та BPIPPRM Service виконують аналогічні функції для даних рельєфу та забудови, запускаючи відповідні препроцесори.

Cadastre ETL Service відповідає за найскладнішу частину – перетворення бізнес-даних (напр., з Excel-таблиць) у формалізовані об'єкти системи.

Нижній рівень (“Інфраструктура Даних (Ядро UESO)”) реалізує “єдине джерело істини”.

Аналіз повного спектру даних, необхідних для функціонування інтелектуалізованої системи (від «сирих» метеорологічних файлів до семантики управлінських рішень), показує, що жодна монолітна технологія зберігання не здатна ефективно впоратися з усіма трьома викликами: семантичною складністю, геопросторовими запитами та зберіганням "важких" артефактів. На перший погляд, реляційна СУБД з розширенням PostGIS чудово зберігає структуровані дані (списки котелень) і є світовим стандартом для високопродуктивних геопросторових запитів. Однак для нашої задачі вона має фундаментальні недоліки: семантична бідність та відсутність логічного виведення. Також жорсткість схеми утруднює додавання нового типу джерела (напр., EcoNoise для шумового забруднення) або нової складної управлінської сутності та вимагатиме складної та ризикованої міграції всієї схеми бази даних, що робить систему крихкою та складною в розвитку [10299].

Онтологія вирішує всі семантичні проблеми реляційної БД, оскільки вона гнучка, підтримує логічне виведення і чудово моделює складні зв'язки (як у сценаріях) [34]. Однак вона має недоліки у двох інших критичних для нас аспектах:

- неефективність зберігання «важких артефактів». Онтологія призначена для зберігання знань, а не файлів. Спроба "зберегти" 1-річний метеорологічний файл .SFC (сотні мегабайт) або растровий файл рельєфу GeoTIFF (гігабайти) безпосередньо в онтологію у вигляді xsd:base64Binary літералу є технічно неможливою або призведе до деградації продуктивності всієї бази знань.

- низька продуктивність складних геопросторових запитів. Хоча стандарт GeoSPARQL існує для онтологій, він своєю потужністю та швидкістю поступається оптимізованим R-tree індексам PostGIS. Виконати запит «знайти всі котельні в межах полігону D зі складною геометрією» засобами GeoSPARQL буде на порядки повільніше, ніж аналогічний запит у PostGIS [28].

Тому було обрано гібридну модель зберігання, що складається з трьох компонентів:

- файлове сховище (Артефакти) – великі, часто бінарні файли, які неможливо або недоцільно зберігати в базі даних (напр., багаторічні метеофайли .sfc або .pfl, оброблені файли рельєфу);

- база даних (наприклад, PostgreSQL + PostGIS) – високопродуктивна база даних для зберігання специфічних даних, які потребують оптимізованої обробки. В першу чергу, це геопросторові дані (геометрія у форматі WKT/WKB), для яких PostGIS надає потужні інструменти індексації та просторових запитів.

- онтологія – семантичний хаб всієї інфраструктури. Це не просто сховище даних, а сховище знань. Онтологія UESO має зберігати усі класи та властивості, а також, усі індивіди та їхні атрибути.

Найголовніше – онтологія пов'язує всі компоненти в єдине ціле. Як показано на схемі, індивід в онтології містить посилання на геометрію (WKT) (яке вказує на відповідний запис у PostGIS) та посилання на артефакти (яке вказує на відповідний файл у файловому сховищі).

Схема ілюструє два типи потоків, ініційованих ETL-сервісами:

- потік даних – Cadastre ETL Service має створювати індивіди в онтології та одночасно зберігати їхню геометрію в PostGIS;

- потік артефактів та метаданих – AERMET Service розміщує файл .sfc у файлове сховище, а потім лише оновлює метадані в онтології (напр., для індивіда `MeteoData_2025` встановлює властивість `hasSfcFilePath = 'path/to/file.sfc'`).

Таким чином, будь-який інший сервіс системи повністю ізольований від хаосу зовнішніх даних. Достатньо зробити один семантичний запит (SPARQL) до онтології і вона поверне йому повний, валідований набір даних, включаючи всі необхідні посилання та файли-артефакти.

Отже, така гібридна архітектура дозволяє системі одночасно бути семантично багатою (завдяки онтології), здатною зберігати великі артефакти

(завдяки файловому сховищу) та високопродуктивною у геопросторових запитах (завдяки PostGIS). В свою чергу, препроцесорні сервіси (ETL) інкапсулюють складність розподілу цих даних, а обчислювальні сервіси отримують всі необхідні посилання через один простий SPARQL-запит до онтології, яка виступає центральним інтеграційним хабом.

2.2. Математичні та логічні основи онтологічної моделі Urban Emission Source Ontology

Для подолання фундаментальних обмежень файло-орієнтованих та реляційних підходів, що полягають у семантичній неоднозначності, відсутності механізмів логічного виведення та низькій інтероперабельності, в даній роботі пропонується використання онтології як формальної основи для моделювання предметної області. Запропонована онтологія UrbanEmissionSourceOntology (UESO) формалізовано на основі апарату теорії множин та алгебри кортежів.

Онтологія *UESO* формально визначається як кортеж з елементів:

$$UESO = \langle C, P, I, L, A, T, G, S, F \rangle \quad (2.1)$$

де,

C – скінченна множина класів (понять) предметної області;

P – скінченна множина властивостей (ролей, предикатів), що описують зв'язки між поняттями;

I – скінченна множина індивідів (екземплярів класів), що представляють конкретні об'єкти реального світ;

L – множина літералів (значень даних, таких як рядки, числа, дати);

A – скінченна множина аксіом, що визначають логічні обмеження та зв'язки між елементами онтології;

T – множина типів даних, що використовуються для літералів (напр., xsd:string, xsd:decimal);

G – множина геометричних примітивів, визначених у стандарті GeoSPARQL;

S – множина сценаріїв, що використовуються для прогностичного моделювання;

F – функція трансляції, що відображає семантичне представлення даних у синтаксичний формат моделі AERMOD.

Множина класів C є частково впорядкованою множиною відношенням включення підкласу $\sqsubseteq$ (subsumption). Ієрархія класів для гетерогенних джерел викидів визначається наступним чином:

$$C_{sources} = \{PointSource, AreaSource, VolumeSource, LineSource\}, \quad (2.2)$$

де

$$\forall c \in C_{sources}, c \sqsubseteq EmissionSource, \quad (2.3)$$

кожен тип джерела є підкласом *EmissionSource*;

$$\forall c_i, c_j \in C_{sources}, i \neq j, : c_i \sqcap c_j \equiv \perp, \quad (2.4)$$

жоден індивід не може одночасно належати до двох різних типів джерел.

На рис. 2.3 наведено множину основних класів розробленої онтології. Запропонована онтологія UESO спроектовано за модульними та ортогональними принципами, успадкованими від онтології верхнього рівня SWEET (рис. 2.4). Врахування онтології SWEET через успадкування класу *ueso:EmissionParameter* від *sweet:PhysicalProperty* забезпечує інтеграцію з іншими науковими інформаційними системами.

Інтеграція геопросторових даних реалізована на основі стандарту OGC GeoSPARQL. Кожен екземпляр класу *ueso:EmissionSource* (як підкласу *geo:Feature*) пов'язується з його геометрією через об'єктну властивість *geo:hasGeometry*. Сама геометрія зберігається як літерал з типом даних *geo:wktLiteral*, що містить представлення у форматі Well-Known Text.

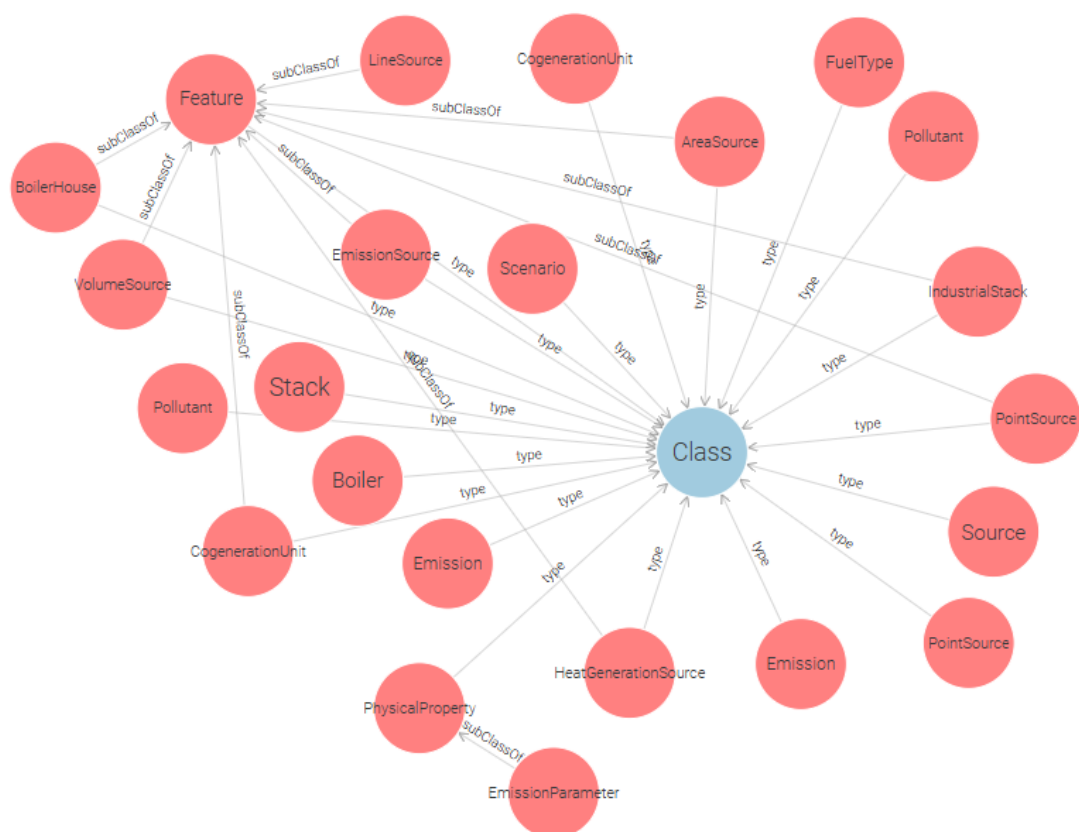


Рис. 2.3. Множина класів (понять) онтології *UESO*

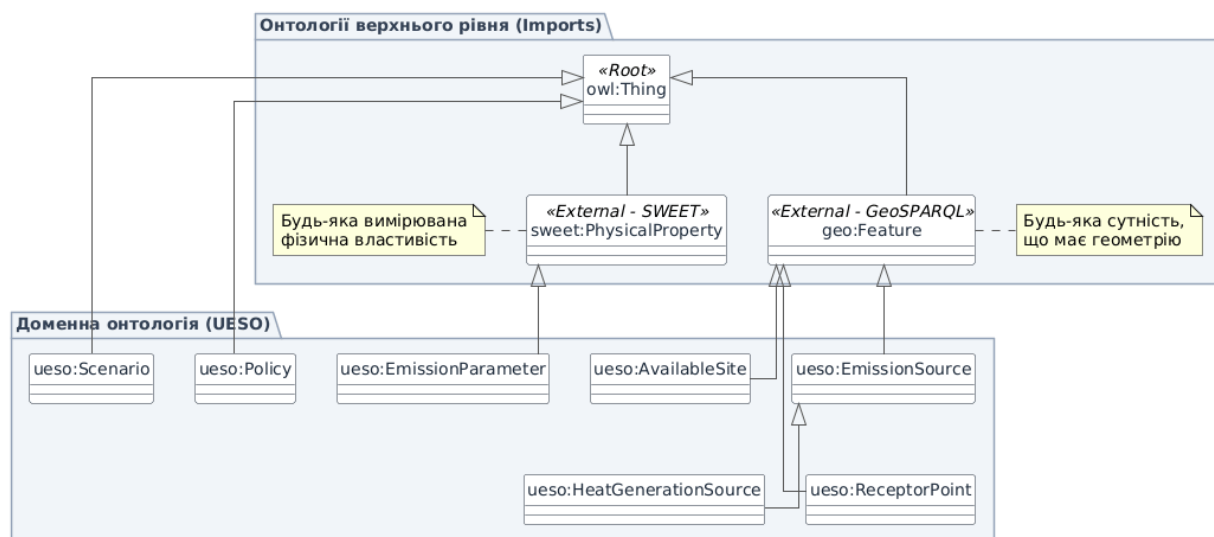


Рис. 2.4. Діаграма наслідування UESO від онтологій верхнього рівня

Для врахування специфіки об'єктів теплогенерації клас *PointSource* далі деталізується:

- $Boiler \sqsubseteq PointSource$ – котельня є підкласом точкового джерела (рис. 2.5);

- $CogenerationUnit \sqsubseteq PointSource$ – когенераційна установка є підкласом точкового джерела (рис. 2.6);

- $Boiler \sqcap CogenerationUnit \equiv \perp$ – об'єкт не може бути одночасно котельнею і когенераційною установкою.

Множина властивостей P складається з двох непересічних підмножин:

- P_O – множина об'єктних властивостей (Object Properties), що пов'язують індивіди з індивідами. Формально, кожна $p \in P_O$ є бінарним відношенням $p \subseteq I \times I$;

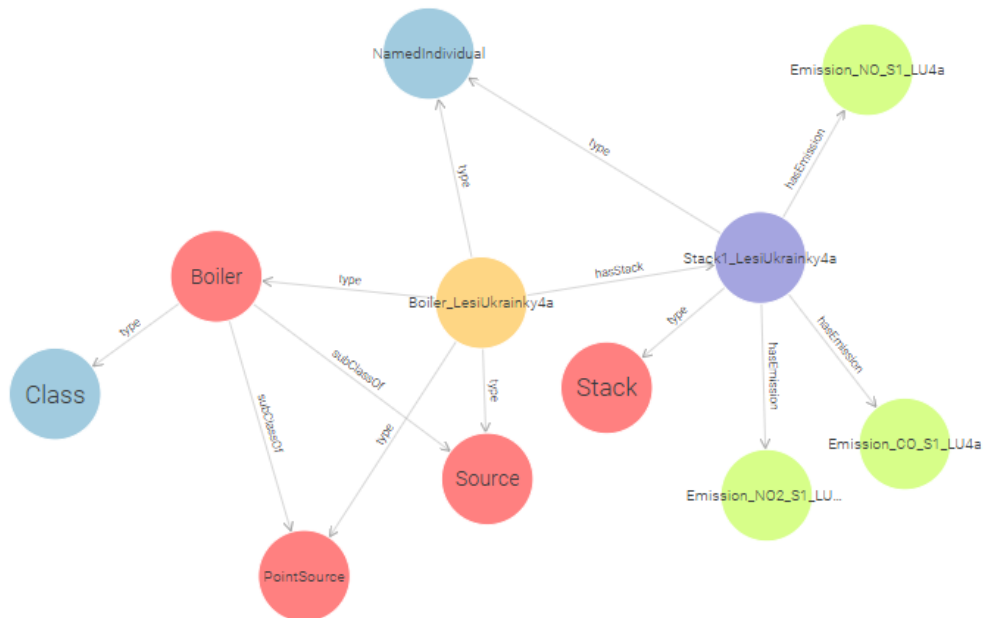


Рис. 2.5. Фрагмент графу онтології опису об'єкта теплогенерації – котельні на природному газі

- P_D – множина властивостей даних (Datatype Properties), що пов'язують індивідів з літералами. Формально, кожна $p \in P_D$ є бінарним відношенням $p \subseteq I \times L$.

Для кожної властивості $p \in P$ визначається її домен $dom(p) \subseteq C$ та діапазон $ran(p) \subseteq C \cup T$. Наприклад, для властивості *hasStackHeight* (має висоту труби):

$$\text{dom}(\text{hasStackHeight}) = \text{PointSource},$$

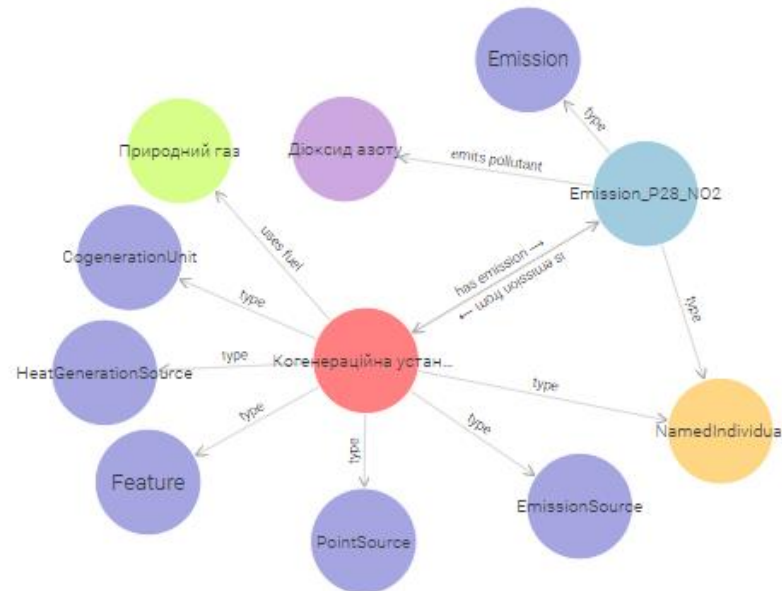
$$\text{ran}(\text{hasStackHeight}) = \text{xsd:decimal}.$$


Рис. 2.6. Фрагмент графу онтології опису когенераційної установки на природному газі

Множина аксіом A є основою для логічного виведення та валідації. Вона включає:

- термінологічні аксіоми (TBox), які визначають ієрархію та обмеження для класів. Наприклад, клас *PointSource* визначається не лише як нащадок *EmissionSource*, а й через обмеження на властивості (property restrictions), що кодують вимоги AERMOD:

$$\begin{aligned} \text{PointSource} \equiv \text{EmissionSource} \sqcap (\exists \text{hasStackHeight. xsd:decimal}) \sqcap \\ (\exists \text{hasStackDiameter. xsd:decimal}), \end{aligned} \quad (2.5)$$

тобто, точкове джерело є таким джерелом викидів, яке має висоту та діаметр труби.

Асертивні аксіоми (ABox), що визначають факти про індивідів. Кожен факт є кортежем, зокрема:

- твердження про належність до класу:

$$(i, c), i \in I, c \in C, \quad (2.6)$$

яке означає, що індивід i є екземпляром класу c ;

- твердження про властивість:

$$(i_1, p, i_2) \forall p \in P_O, \quad (2.7)$$

$$(i, p, l), \forall p \in P_D, \quad (2.8)$$

які, означають, що індивіди i_1 та i_2 пов'язані властивістю p , або індивід i має властивість p зі значенням l .

Кожен сценарій з множини сценаріїв $s \in S$ визначається як кортеж:

$$s = \langle s_{id}, s_{label}, I_s, I_R, I_L, I_M, f_M \rangle, \quad (2.9)$$

де,

s_{id} – унікальний ідентифікатор сценарію;

s_{label} – текстовий опис;

$I_s \subseteq I$ – множина індивідів-джерел, що включені в даний сценарій;

$I_R \subseteq I$ – множина індивідів-рецепторів, що включені в даний сценарій;

$I_P \subseteq I$ – множина індивідів-політик імплементації, що включені в даний сценарій;

$I_M \subseteq I$ – множина індивідів-специфікацій вхідних даних моделювання (метеодані, забудова тощо), що включені в даний сценарій;

$f_M \subseteq F$ – множина функцій трансляції.

Функції трансляції F є ключовим елементом, що пов'язують семантичну модель з обчислювальним модулем. Вони визначається як відображення, що для заданого індивіда-джерела $I_s \subseteq I$, індивіда-рецептора $I_R \subseteq I$ та індивіда-специфікацій вхідних даних моделювання $I_M \subseteq I$ зі сценарію $s \in S$ генерує послідовність рядків, що відповідають синтаксису опису вхідних даних для моделювання на основі обчислювального модуля AERMOD:

Отже, в даному пункті розроблено доменну онтологію UESO, яка формалізує знання про гетерогенні міські джерела викидів. На відміну від реляційних баз даних, онтологічна модель забезпечує семантичну інтероперабельність та логічну несуперечливість даних, створюючи обчислювальну основу для автоматизації та інтелектуалізації процесів моделювання.

2.3. Семантичне моделювання сценаріїв для проактивного управління

В основі запропонованої інтелектуалізованої системи лежить формальна математична модель «сценарію», яка перетворює абстрактні управлінські задачі на чіткі, відтворювані та автоматизовані обчислювальні експерименти.

Завдяки розробленій онтологічній моделі та гнучкій сервіс-орієнтованій архітектурі, система надає інструментарій для комплексного, багаторівневого сценарного аналізу (рис. 2.8).

Основні типи сценаріїв, які можна оцінювати за допомогою інтелектуалізованої системи, можна згрупувати за їхньою управлінською метою:

1. Сценарії стратегічного розвитку та планування. Ця група сценаріїв відповідає на питання "що-якщо" при плануванні нової міської інфраструктури.

Зокрема,

- «Оцінка впливу нового об'єкта» - проєкт будівництва нової котельні або встановлення когенераційної установки (КГУ) для забезпечення енергією нового житлового комплексу. Користувач створює в онтології новий «віртуальний» індивід (напр., `ueso:BoilerHouse_Project`) з проєктними параметрами (координати, висота труби, планова потужність викидів). Створюється новий сценарій, який включає всі існуючі джерела плюс це нове

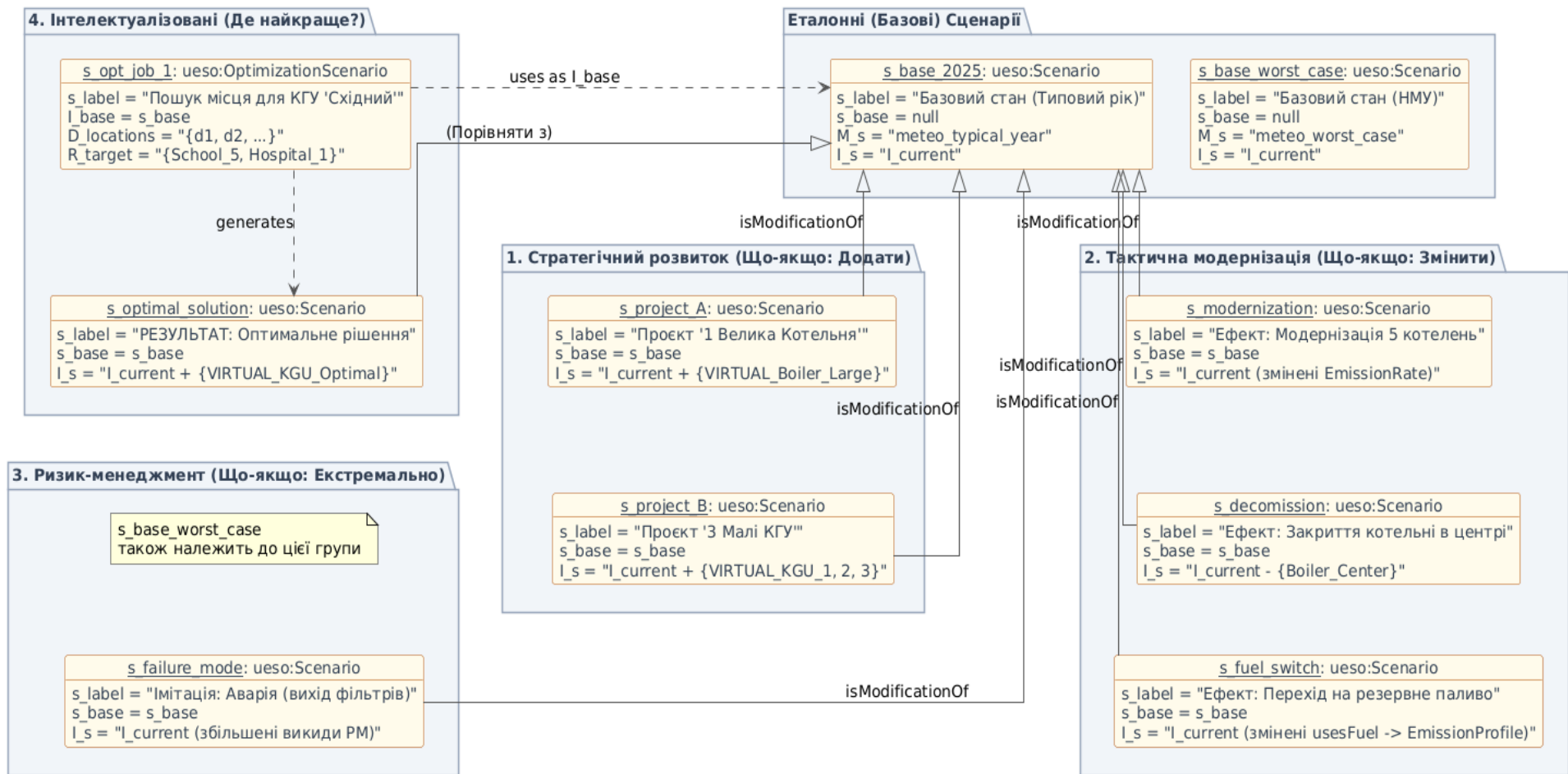


Рис. 2.8. Таксономія сценарного аналізу

джерело. Система розраховує кумулятивний вплив та показує, який саме внесок (зону впливу) створить новий об'єкт та чи не призведе це до перевищення ГДК у найближчих чутливих точках (школах, лікарнях).

- «Порівняльний аналіз технологій» Вибір між будівництвом однієї великої котельні або трьох менших децентралізованих когенераційних установок. Створюються два окремі, але паралельні сценарії. Одна велика котельня та три малі КГУ. Система розраховує обидва варіанти, і управлінець може наочно на карті порівняти поля концентрацій, сумарний внесок та пікові значення в рецепторах, щоб обрати екологічно безпечніший варіант.

2. Сценарії тактичної модернізації та політики

Ця група відповідає на питання, пов'язані з покращенням або зміною існуючої інфраструктури:

- «Оцінка ефекту від модернізації». Підприємство енергопостачання планує модернізувати 5 найстаріших котелень, встановивши нові пальники, що на 30% зменшать викиди NO₂. Потрібно оцінити, як це покращить загальну якість повітря в місті. Створюється базовий сценарій з поточними даними. Потім створюється дочірній сценарій, в якому для п'яти обраних індивідів `ueso:BoilerHouse` змінюється значення властивості `ueso:hasEmissionRate` для NO₂. Система може розрахувати обидва сценарії і показати карту дельти – тобто, на скільки саме (в мкг/м³) зменшиться забруднення в кожній точці міста.

- «Виведення з експлуатації / консервація». Оцінка впливу планового закриття старої котельні, наприклад, в центрі міста. Створюється сценарій, в якому множина джерел просто не включає індивід котельні, що закривається. Це дозволяє спрогнозувати покращення екологічної ситуації в прилеглих районах.

- «Зміна паливної політики». Оцінка наслідків тимчасового переходу котелень з природного газу на резервне паливо (напр., мазут) під час енергетичної кризи. Створюється сценарій, де у відповідних індивідів змінюється властивість `ueso:usesFuel`, що автоматично (через правила в

онтології або в логіці сервісу) призводить до зміни всього профілю викидів (напр., різко зростають викиди твердих часток та SO₂).

3. Сценарії ризик-менеджменту та «Найгіршого випадку»

Ця група відповідає на питання "що-якщо" при екстремальних або нештатних ситуаціях:

- «Аналіз найгіршого випадку». Оцінка впливу всіх джерел за найнесприятливіших метеорологічних умов – наприклад, тривалий штиль, температурна інверсія, що не дозволяє забрудненню розсіюватися. Користувач запускає розрахунок базового сценарію (з повним набором джерел), але в параметрах сценарію обирає не стандартний кліматичний файл, а спеціально підготовлений файл «найгірших» метеоумов. Це дозволяє побачити максимальні теоретично можливі концентрації та перевірити, чи не будуть вони перевищувати ГДК.

- «Імітація аварійного режиму». На одній з найбільших котелень вийшла з ладу система очистки димових газів. Створюється тимчасовий сценарій, де для одного конкретного індивіда `ueso:BoilerHouse` властивість `ueso:hasEmissionRate` для твердих часток тимчасово збільшується в 10 разів. Це дозволяє для служб ДСНС та екологічного контролю спрогнозувати зону та інтенсивність забруднення для вживання оперативних заходів.

4. Інтелектуалізовані (оптимізаційні) сценарії

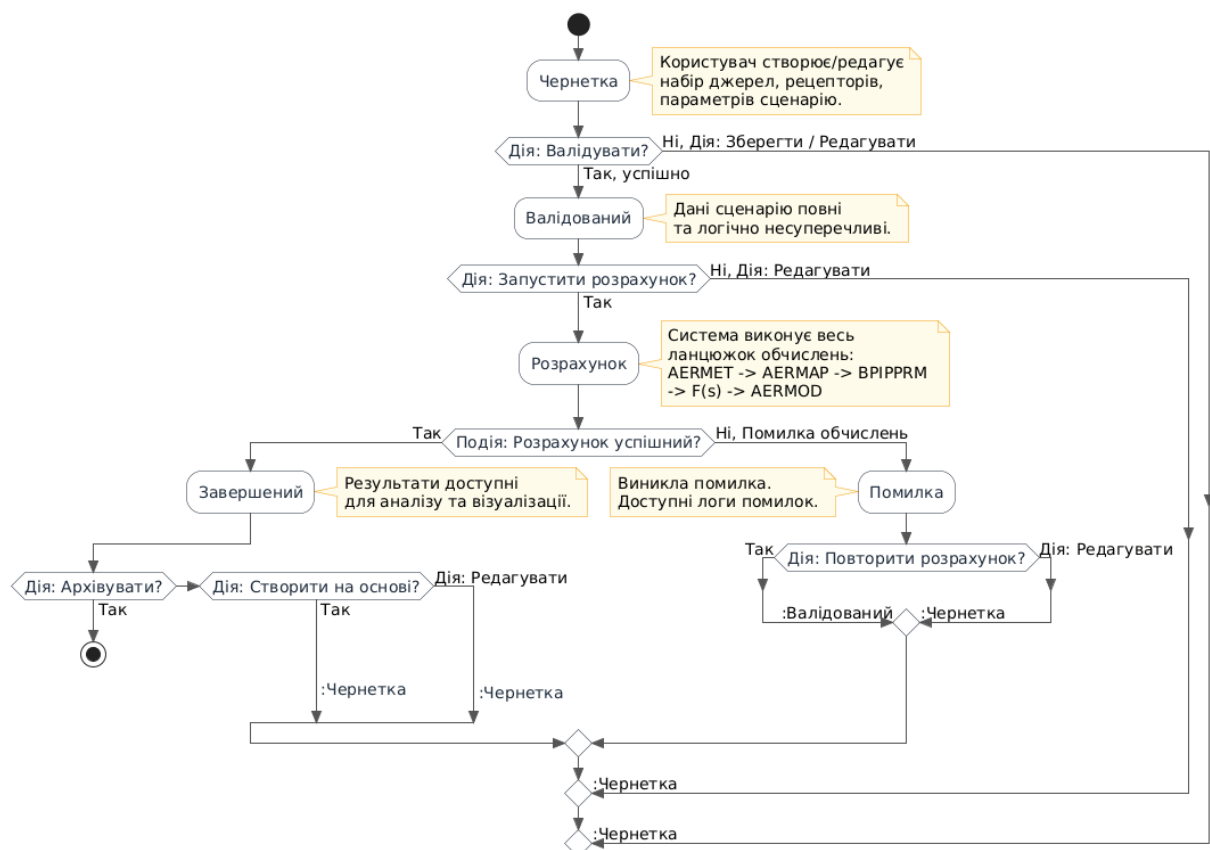
Це найвищий рівень сценарного аналізу, що реалізується. Він відповідає не на питання «де найкраще?»:

- «Оптимізація розміщення нового об'єкта». Міська влада виділила територію (напр., промислову зону) площею 2 км² для будівництва нової КГУ. Потрібно знайти конкретну точку всередині цієї зони, де розміщення установки створить мінімальний сукупний вплив на три найближчі житлові масиви. Користувач створює сценарій, вказуючи: параметри віртуального джерела (потужність, висота труби), множину рецепторів (три житлові масиви), полігон пошуку (можливі точки розміщення на заданій території). Сервіс самостійно запускає ітерації моделювання, оптимізуючи точки

розміщення за допомогою метаевристичних алгоритмів, і в кінці видає рекомендовані координати для будівництва.

Завдяки онтологічній базі та сервіс-орієнтованій архітектурі, перехід між усіма цими сценаріями стає гнучким, швидким та, головне, академічно відтворюваним, оскільки кожен сценарій є формально визначеним об'єктом у базі знань.

Фундаментом для будь-якого проактивного аналізу є базовий сценарій, який позначаємо як S_{base} та який описаний в попередньому пункті розділу. Цей сценарій являє собою найбільш точний та актуальний опис поточного стану екосистеми («as-is»). Формально, S_{base} є індивідом онтології (ueso:Scenario), що містить повний набір існуючих джерел I_s , ключових рецепторів R_s та типових метеорологічних умов M_s (напр., усереднені дані за рік). Він слугує еталонною «нульовою точкою», з якою порівнюються всі альтернативні варіанти розвитку. Життєвий цикл сценарію проілюстровано на рисунку 2.9.



2.9. Діаграма послідовності для життєвого циклу сценарію

Відштовхуючись від цього базового стану, система проактивного управління реалізує два фундаментально різні, але взаємопов'язані, типи прогностичного моделювання: оцінювальний та оптимізаційний.

Оцінювальний сценарій – це фундаментальний тип сценарію, призначений для відповіді на питання «що-якщо?» (наприклад, «Яким буде вплив, якщо ми додамо котельню в точці А?»). У цьому випадку користувач надає повний, фіксований набір вхідних даних, а система розраховує детермінований результат.

Оцінювальний сценарій s_{eval} формально визначається як кортеж:

$$s_{eval} = \langle s_{id}, s_{label}, s_{base}, I_s, R_s, M_s, C_s \rangle, \quad (2.11)$$

де,

s_{id} – унікальний ідентифікатор (індивід ueso – Scenario);

s_{label} – текстовий опис (напр., «Проект – Модернізація котельні на вул. Лесі Українки»);

s_{base} – посилання на індивід базового сценарію s_{base} ;

$I_s \subseteq I$ – фіксована множина індивідів-джерел, що включені в сценарій. Кожен $i \in I_s$ має повністю визначені параметри (координати, висота, потужність викидів тощо);

$R_s \subseteq R$ – фіксована множина індивідів-рецепторів, для яких виконується оцінка (напр., житловий масив, лікарня);

$M_s \in M$ – фіксований набір метеорологічних даних;

$C_s \in C$ – фіксований набір параметрів конфігурації моделі (напр., врахування урбанізованої забудови).

Цей кортеж є повним, самодостатнім вхідним аргументом для функції-генератора F вхідного файлу для обчислювального модуля AERMOD, яку ми визначили в онтології UESO:

$$F(s_{eval}) \rightarrow String^*. \quad (2.12)$$

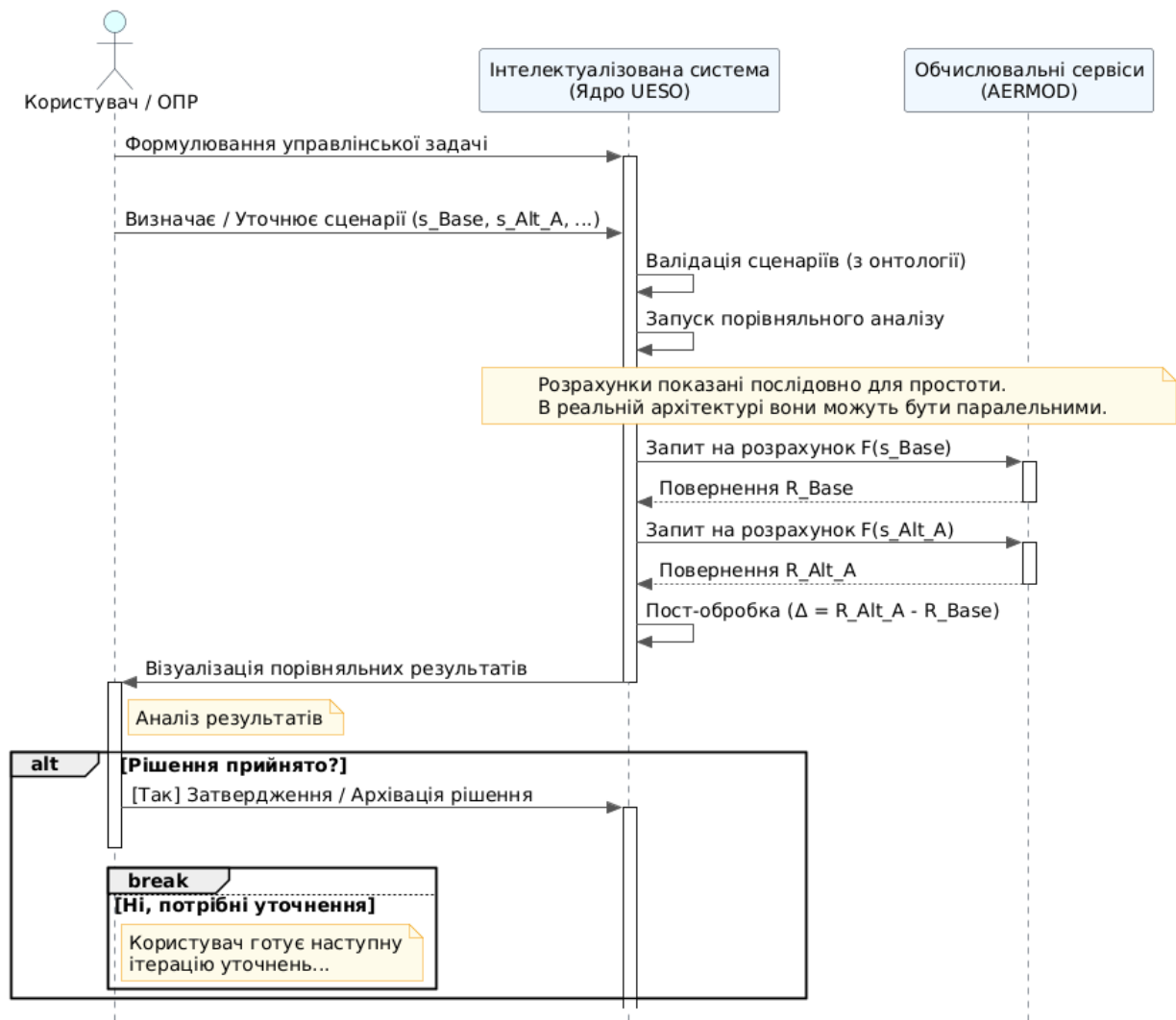


Рис. 2.10. Діаграма послідовності виконання оцінювального сценарію

Застосування $F(s_{eval})$ генерує керуючий файл, а подальший розрахунок повертає одне поле концентрацій R_{map} для одного фіксованого стану системи. Наприклад, сценарій – «Показати вплив від модернізації котельні на вул. Лесі Українки». Користувач створює індивід сценарію s_{modern} , де s_{base} посилається на $s_{base_{2025}}$, а I_s містить ту саму множину джерел, що і в $s_{base_{2025}}$, але у індивіда «Котельня вул. Лесі Українки» змінено (зменшено) значення властивості *ueso* – *hasEmissionRate*. Система запускає $F(s_{modern})$ і візуалізує результат, автоматично порівнюючи його з результатом $s_{base_{2025}}$.

Оптимізаційний сценарій – це інтелектуалізований тип сценарію, що відповідає на питання «Де найкраще?» (напр., «Де розмістити нову

когенераційну установку (КГУ), щоб мінімізувати вплив на лікарні, школи, житлові масиви в районі?»). Тут ключові параметри (координати) є невідомими змінними, які система повинна знайти шляхом застосування методів оптимізації.

Оптимізаційний сценарій s_{opt} визначається як кортеж, що формалізує задачу оптимізації:

$$s_{opt} = \langle o_{id}, o_{label}, I_{base}, I_{virt}, V, D, R_{target}, M_s, C_s, f_{obj}, O_{method} \rangle, (2.13)$$

де,

o_{id} – унікальний ідентифікатор (напр., індивід *ueso* – *Optimization*);

o_{label} – текстовий опис (напр., «Пошук оптимального місця для КГУ в районі м. Тернопіль»);

$I_{base} \subseteq I$ – фіксована множина фонових джерел. Зазвичай, це множина джерел з базового сценарію I_s з s_{base} ;

$I_{virt} \in I$ – віртуальне джерело, яке оптимізується (напр., «проектна КГУ 5 МВт»);

V – множина дискретних змінних, які потрібно оптимізувати. Це невідомі характеристики I_{virt} . Наприклад – $V = \{x, y\}$, де x, y – координати I_{virt} ;

D – область допустимих значень для змінних V , що визначає простір пошуку. Наприклад, $D = \{ (x, y) \mid (x, y) \in Polygon_{Industrial_Zone} \}$ (координати x, y повинні знаходитись у межах полігону промислової зони);

$R_{target} \subseteq R$ – цільова множина рецепторів, для яких розраховується вплив. Наприклад – $R_{target} = \{ Receptor_{School_5}, Receptor_{Hospital_1} \}$;

f_{obj} – цільова функція, яку необхідно мінімізувати. Вона приймає набір змінних V і повертає одне числове значення (оцінку «якості» рішення), Наприклад, $f_{obj(x,y)} \rightarrow C$, де C – це максимальна концентрація на будь-якому рецепторі з R_{target} ;

O_{method} – метод оптимізації, що використовується (напр., модуль EcoOptimizer на базі метаевристичного алгоритму).

Ключовий зв'язок між цими двома моделями полягає в тому, що цільова функція f_{obj} оптимізаційного сценарію реалізується через багаторазовий виклик оцінювальних сценаріїв s_{eval} .

Наприклад, сценарій – «Знайти в районі таке місце для нової КГУ 5 МВт, щоб максимальна концентрація NO_2 на території лікарні та школи була мінімальною». Користувач створює $s_{opt} - I_{base}$ (з $s_{base_{2025}}$), I_{virt} (шаблон КГУ 5 МВт), $V = \{x, y\}$, D (полігон району), R_{target} (координати рецепторів (лікарня, школа)), f_{obj} («max NO_2 on R_{target} »). Система запускає EcoOptimizer, який ітеративно викликає EcoHeat Service (реалізуючи f_{obj}) і повертає користувачу рекомендовані координати (x_{best}, y_{best}) .

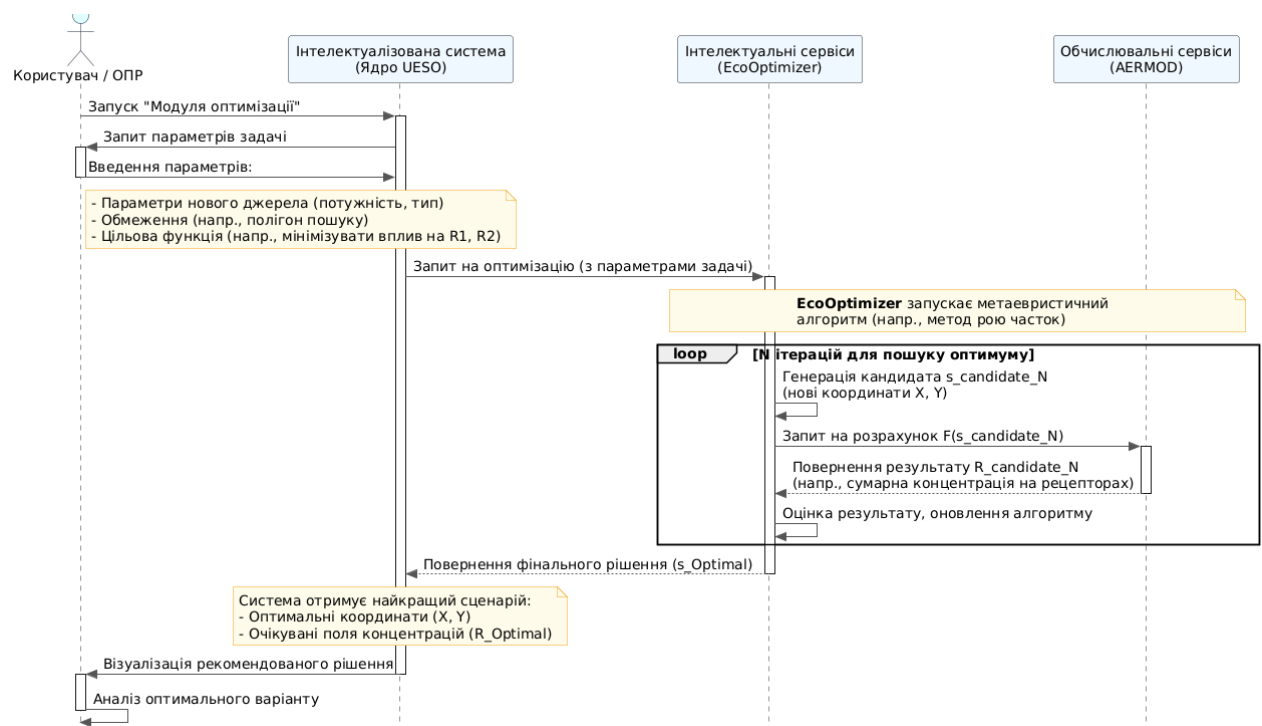


Рис. 2.11. Діаграма послідовності виконання оптимізаційного сценарію

Таким чином, ця двоетапна математична модель дозволяє системі підтримувати як керований користувачем аналіз «що-якщо» (через s_{eval}), так і повністю автоматизований пошук «де найкраще» (через s_{opt}),

використовуючи єдиний, логічно несуперечливий набір даних з онтології. Відповідно здійснено ключовий перехід від статичної моделі знань до динамічного інструментарію управління шляхом математичної формалізації сценарного аналізу. Ця формалізація слугує «семантичним містком» між онтологією UESO та обчислювальними алгоритмами.

2.4. Оцінка ефективності онтології UESO на основі її структурної складності

Для об'єктивного оцінювання якості розробленої онтології UESO недостатньо лише перевірки її логічної несуперечливості. Необхідно провести кількісний аналіз її структурної складності. Такий аналіз дозволяє оцінити придатність онтології для супроводу, розширення та ефективність її використання для автоматизованих запитів. Оцінка проводиться на основі набору стандартизованих онтологічних метрик, що характеризують розмір, глибину, ширину та зв'язність моделі знань [67].

Опис формальних метрик для оцінки ефективності онтології.

Метрики розмірності (Size Metrics) є базовими показниками, що характеризують загальний обсяг онтології:

C – кількість класів, що відображає загальну кількість унікальних понять (категорій) у словнику предметної області;

P_O – кількість об'єктних властивостей, що характеризує кількість унікальних типів семантичних зв'язків між класами;

P_D – кількість властивостей даних, що характеризує кількість унікальних атрибутів, що пов'язують класи з літеральними значеннями (рядками, числами).

Метрики ієрархії (Inheritance Metrics) описують вертикальну складність таксономії:

max_{DIT} – максимальна глибина ієрархії, яка характеризується довжиною найдовшого шляху `rdfs:subClassOf` від кореневого класу `owl:Thing` до найглибшого похідного класу. Рекомендовано такі значення max_{DIT} :

$$3 < max_{DIT} < 10,$$

де

$max_{DIT} < 3$ – це, як правило, погано структурована модель;

$max_{DIT} > 10$ – це, як правило, надмірно ускладнена (“over-engineered”) академічна модель, яку важко розуміти, підтримувати та ефективно використовувати для запитів;

avg_{DIT} – середня глибина ієрархії – середнє арифметичне глибини для всіх класів онтології.

Метрики зв'язності (Coupling Metrics), які описують, наскільки сильно класи пов'язані між собою (на відміну від простої таксономії):

RR – коефіцієнт зв'язності (Relationship Richness), який показує середню кількість зв'язків на один клас та розраховується як:

$$RR = \frac{P_o}{C},$$

$RR \ll 1$ – «бідні» онтології (Poor Ontologies), які є простими таксономіями (ієрархіями «є різновидом», як-от класифікація тварин), та не описують, як сутності взаємодіють;

$RR \approx 1 \mid RR > 1$ – «насичені» онтології (Rich Ontologies), що мають високу зв'язність між даними та знаннями;

$max_{Coupling}$ – найбільш зв'язний клас (Hub Class), тобто клас, що має найбільшу кількість вхідних та вихідних об'єктних властивостей.

Метрики семантичної насиченості (Richness Metrics):

$A_{restrict}$ – кількість аксіом обмежень, що характеризує кількість формальних логічних правил (напр., екзистенційних $\exists$ або універсальних $\forall$

обмежень), що визначають семантику класів. Це ключовий показник, що відрізняє онтологію від реляційної схеми.

На основі розробленої у попередніх онтологіях UESO було розраховано її метричний паспорт. Результати наведено у таблиці 2.1.

Таблиця 2.1.

Метричний паспорт онтології UESO

Категорія	Метрика	Позначення	Значення
Метрики розмірності	Кількість класів (вкл. імпортовані)	C	35
	Кількість об'єктних властивостей	P_O	18
	Кількість властивостей даних	P_D	16
Метрики ієрархії	Максимальна глибина ієрархії	max_{DIT}	5
	Середня глибина ієрархії	avg_{DIT}	2.28
Метрики зв'язності	Коефіцієнт зв'язності	RR	0.51
	Найбільш зв'язний клас (Hub)	$max_{coupling}$	ueso:Optimization Scenario
Метрики насиченості	Кількість аксіом обмежень	$A_{restrict}$	≈ 97

Аналіз метричного паспорта UESO дозволяє зробити висновок про високу ефективність та якість запропонованої онтологічної моделі для вирішення поставлених завдань. Відносно невелика кількість класів ($C = 35$) свідчить про те, що онтологія є сфокусованою на предметній області. Вона не намагається моделювати всю предметну область «Екологія», а фокусується виключно на поняттях, необхідних і достатніх для автоматизації сценарного аналізу на базі AERMOD. Це забезпечує легкість її супроводу та модифікації.

Високий коефіцієнт зв'язності, $RR \approx 0.51$, та значна кількість аксіом обмежень, $A_{restrict} > 25$, є ключовими показниками. Вони доводять, що UESO є не «пласкою» таксономією, а насиченим графом знань. На відміну від реляційної схеми, яка є «пасивною», онтологія UESO є «активною». Вона

містить логічні правила, що дозволяють системі автоматично валідувати дані (напр., перевірити, що *PointSource* має *hasStackHeight*) та виконувати логічне виведення.

Показники ієрархії $max_{DIT} = 5, avg_{DIT} = 2.28$ свідчать про збалансовану, логічно зрозумілу структуру без надмірного ускладнення, що позитивно впливає на швидкість виконання SPARQL-запитів. Добре структурована спеціалізація (напр., *Thing* $\rightarrow$ *Feature* $\rightarrow$ *PointSource* $\rightarrow$ *HeatGenerationSource* $\rightarrow$ *BoilerHouse*).

Клас інтелектуалізованого сценарію є центральним інтеграційним вузлом системи.

Таким чином, кількісні метрики підтверджують, що розроблена онтологія UESO є оптимальною компромісною моделлю. Вона достатньо компактна для ефективної програмної реалізації та супроводу, і водночас достатньо семантично насичена та зв'язана, щоб слугувати надійним інтелектуальним ядром для автоматизації складних робочих процесів, які розглядаються в наступних розділах.

Висновки до розділу 2

У даному розділі обґрунтовано перехід від неструктурованих, файло-орієнтованих робочих процесів до формалізованої керованої знаннями моделі, що є необхідною передумовою для створення інтелектуалізованої системи проактивного управління.

Проведений аналіз довів, що традиційні підходи до зберігання даних (файли, реляційні БД) є семантично бідними, негнучкими та нездатними ефективно моделювати складні, гетерогенні сутності міської екосистеми, що унеможливорює автоматизацію сценарного аналізу.

У другому розділі:

- розроблено та обґрунтовано гібридну архітектуру збору та інтеграції даних. Дана архітектура використовує онтологію UESO як центральний

семантичний хаб, що інкапсулює складність гетерогенних джерел (AERMET, BPIPRM, кадастр) та відокремлює логіку знань від зберігання артефактів (файлове сховище) та високопродуктивних гео-даних (PostGIS);

- спроектовано доменну онтологію UESO. Розроблена ієрархія класів формалізує не лише фізичні об'єкти (джерела та рецептори), але й ключові управлінські та обчислювальні сутності, забезпечуючи при цьому інтероперабельність через наслідування від онтологій верхнього рівня (SWEET, GeoSPARQL);

- формалізовано основи сценарного аналізу. На базі онтології UESO надано формалізоване визначення двох фундаментально різних типів проактивного аналізу: оцінювального та оптимізаційного сценаріїв. Оцінювальний сценарій формалізовано як повний, самодостатній кортеж, що забезпечує відтворюваність експериментів «що-якщо». Оптимізаційний сценарій формалізовано як задачу комбінаторної оптимізації для якої необхідно застосовувати метаевристичні алгоритми;

- проведено кількісну оцінку структурної складності розробленої онтології. Метричний паспорт показав, що модель UESO є збалансованою: компактною, що забезпечує її легкий супровід, та семантично насиченою (коефіцієнт зв'язності $RR = 0.51$, кількість аксіом $A_{restrict} \approx 97$). Це доводить, що UESO є не простою таксономією, а мережевим графом знань з вбудованою логікою валідації, що є її ключовою перевагою над традиційними реляційними схемами.

Таким чином, у розділі 2 сформовано повний, валідований та семантично багатий набір інструментів інформаційного забезпечення – «єдине джерело істини». Це створює необхідний і достатній фундамент для переходу до наступного етапу – розробки математичного забезпечення для вирішення формалізованих задач сценарного аналізу та оптимізації.

РОЗДІЛ 3.

МАТЕМАТИЧНЕ ТА АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛІЗОВАНОЇ СИСТЕМИ

Попередній розділ заклав повний інформаційний та логічний фундамент для інтелектуалізованої системи. Було розроблено доменну онтологію UESO як «єдине джерело істини» та, на її основі, надано строгу математичну формалізацію двох ключових типів проактивного аналізу: оцінювального та оптимізаційного сценаріїв.

Якщо розрахунок оцінювального сценарію є детермінованою, хоча й обчислювально-затратною, задачею, то практична реалізація оптимізаційного сценарію є нетривіальною задачею. Як було показано, вона зводиться до NP-складної задачі комбінаторної оптимізації з високою обчислювальною складністю цільової функції.

У цьому розділі розглядається розробка математичного та алгоритмічного забезпечення інтелектуалізованої програмної системи для вирішення задачі проактивного управління екосистеми урбанізованих територій. Ключовим завданням розділу є формалізація задачі оптимізаційного сценарію – розміщення нових джерел техногенного навантаження (котелень, когенераційних установок) як задачі комбінаторної оптимізації. Проактивний характер управління забезпечується шляхом інтеграції точних фізичних моделей розсіювання (AERMOD) безпосередньо в цільову функцію оптимізації, таким чином, щоб мінімізувати ризик порушення санітарних норм у контрольних точках-рецепторах. Це дозволяє системі не просто фіксувати факт забруднення, а гарантувати, що запропонована конфігурація енергетичної мережі не призведе до порушення санітарних норм у майбутньому.

Описується адаптація метаевристичних алгоритмів (GA, PSO, ABC) для роботи в дискретному просторі рішень з використанням уніфікованих операторів у середовищі DEAP.

Особлива увага приділяється інтелектуалізації процесу пошуку рішень. Враховуючи високу обчислювальну складність та варіативність сценаріїв міського планування, розроблено адаптивний метод вибору стратегії оптимізації. Запропонований підхід дозволяє системі автоматично аналізувати складність поставленої задачі та обирати найбільш релевантний алгоритм (від повного перебору до складних метаевристик PSO чи ABC). Такий підхід трансформує теоретично складну математичну задачу у практичний інженерний інструмент, забезпечуючи наукове підґрунтя для стратегічного планування сталого розвитку міста. Завершується розділ практичною апробацією розробленого підходу на прикладі сценарного моделювання для мікрорайону «Сонячний» у м. Тернопіль.

Всі результати отримані в цьому розділі опубліковано в працях [115, 124].

3.1. Математична модель дискретної оптимізації розміщення джерел забруднення в умовах екологічних обмежень

Задача проактивного управління «знайти найкраще місце для k нових джерел» є задачею оптимізації розміщення. Цей клас задач, як детально описано у працях [20, 89, 128], має фундаментальне значення і традиційно фокусується на вирішенні проблем геометричного проектування, таких як упакування або розкрій, де головною складністю є аналітичний опис обмежень взаємного неперетину об'єктів (напр., за допомогою апарату Φ -функцій).

Однак, в даному випадку задача має фундаментальні особливості, що докорінно змінюють її математичну постановку та вимоги до методів вирішення. Ми вирішуємо не геометричну задачу пакування, а задачу розміщення джерел фізичних (концентраційних) полів. Це накладає чотири характеристики на цільову функцію оптимізаційної задачі:

- на відміну від геометричних об'єктів, де вплив є локальним (об'єкт або перетинається, або ні), джерело забруднення створює поле, що поширюється

на всю область. Вплив у будь-якій точці-рецепторі є суперпозицією (сумою) впливів від всіх існуючих фонових джерел та всіх k нових джерел;

- зв'язок між шуканим розв'язком, оптимальне розміщення, та результатом, значеннями концентрації в точках де розміщені рецептори, не описується простою аналітичною формулою (як відстань чи перетин). Він є результатом складного фізичного моделювання (атмосферна дифузія, метеорологія, рельєф), яке в нашому випадку інкапсульоване у зовнішньому обчислювальному модулі (AERMOD). Таким чином, цільова функція є обчислювально дорогою «чорною скринькою»;

- метою проактивного управління екологічною безпекою є не мінімізація середнього забруднення, а мінімізація максимального ризику. Ми повинні гарантувати, що навіть у точці розміщення рецептора з найбільшим значенням концентрації, вона не перевищить норму для модельованого забруднювача;

- на відміну від багатьох задач моделювання (напр., побудови карт для візуалізації), в даному випадку цільова функція не вимагає розрахунку повного, високороздільного поля концентрацій по всій території міста, що є надзвичайно дорогою операцією. Оцінка «якості» розв'язку здійснюється виключно у скінченній, зазвичай невеликій, множині цільових точок-рецепторів. Це дозволяє на порядки скоротити обчислювальну вартість кожного виклику обчислювального модуля AERMOD. Оскільки, розраховується концентрація лише в дискретних точках, а не в точках деякої сітки.

Ці особливості унеможливають застосування класичних градієнтних методів або методів, що базуються на аналітичному описі обмежень (як Ф-функції). Натомість, вони вимагають формулювання задачі як дискретної (комбінаторної) оптимізації «чорної скриньки» та застосування для її вирішення відповідних методів, зокрема, повного перебору або метаевристик.

Формалізуємо її таким чином. Нехай визначено наступні скінченні множини та константи:

$D = \{d_1, d_2, \dots, d_m\}$ – множина m дискретних, допустимих локацій для розміщення нових об'єктів;

k – ціле число, що позначає кількість нових джерел, які необхідно розмістити ($k \leq m$);

$R = \{r_1, r_2, \dots, r_q\}$ – множина q цільових точок-рецепторів, в яких проводиться контроль впливу;

$P = \{p_1, p_2, \dots, p_s\}$ – множина s забруднюючих речовин, що контролюються;

ГДК_p – значення гранично допустимої концентрації ($\text{ГДК}_p > 0$) для забруднюючої речовини $p \in P$;

$C_{r,p}^{base}$ – значення фонові концентрації речовини p в точці розміщення рецептора r , створене всіма існуючими (не оптимізованими) джерелами.

Рішенням X є підмножина з k елементів, обраних з множини D .

$$X \subseteq D, |X| = k. \quad (3.1)$$

Простором пошуку S_D є множина всіх таких рішень:

$$S_D = \{X \mid X \subseteq D \wedge |X| = k\}. \quad (3.2)$$

Загальна кількість можливих рішень (розмірність простору пошуку) дорівнює

$$|S_D| = C_m^k = \frac{m!}{(k! \cdot (m - k)!)}, \quad (3.3)$$

що визначає обчислювальну складність задачі.

Цільова функція $f_{obj}(X)$ відображає будь-яке можливе рішення $X \in S_D$ у кількісну оцінку варіанту розміщення, яку необхідно мінімізувати. Значення цієї функції обчислюється через виклик обчислювального модуля (AERMOD):

$$f_{obj}: S_D \rightarrow \mathbb{R}. \quad (3.4)$$

Позначимо, $C_{r,p}^X$ – додаткова концентрація речовини p в точці розміщення рецептора r , обумовлена k новими джерелами, розміщеними в локаціях X .

Загальна концентрація $C_{r,p}(X)$ в точці розміщення рецептора r для речовини p при рішенні X є лінійною суперпозицією фонові концентрації та додаткового впливу на основі концентрації нових джерел:

$$C_{r,p}(X) = C_{r,p}^{base} + \sum_{i=1}^k {}^i C_{r,p}^X. \quad (3.5)$$

Формально модель дискретної оптимізації розміщення джерел забруднення запишемо кортежем:

$$M = \langle D, R, P, k, \Theta, X, S_D, f_{obj} \rangle, \quad (3.6)$$

де $\Theta = \{ГДК_p, C^{base}\}$ – множина параметрів екологічного стану.

Задача оптимізації розміщення полягає у знаходженні такого рішення X^* з простору S_D , яке мінімізує значення цільової функції $f_{obj}(X)$:

$$X^* = \arg \min_{X \in S_D} f_{obj}(X) \quad (3.7)$$

$$0 \leq C_{r,p}(X) \leq ГДК_p, \forall r \in R. \quad (3.8)$$

Використовуючи нормалізований індекс якості, запишемо цільову функцію f_{obj} за мінімаксним критерієм – як найгірше (максимальне) відхилення від $ГДК_p$ серед усіх цільових рецепторів для контрольованої речовини:

$$f_{obj}(X) = \max_{r \in R} \left\{ \frac{C_{r,p}(X)}{ГДК_p} - 1 \right\}. \quad (3.9)$$

Підставивши $f_{obj}(X)$ у задачу (), отримуємо математичну постановку задачі:

$$X^* = \arg \min_{X \in S_D} \max_{r \in R} \left\{ \frac{C_{r,p}(X)}{\Gamma DK_p} - 1 \right\} \quad (3.10)$$

$$0 \leq C_{r,p}(X) \leq \Gamma DK_p, \forall r \in R. \quad (3.11)$$

Запропонована математична постановка задачі оптимізації, яка базується на дискретному представленні простору пошуку S_D та обмеженій множині рецепторів R , має низку суттєвих переваг порівняно з класичними неперервними підходами до моделювання полів концентрацій. Ці переваги стосуються як обчислювальної ефективності, так і практичної придатності для систем підтримки прийняття рішень.

Такий підхід в першу чергу забезпечує зменшення обсягу обчислень. Найбільш ресурсоємною частиною будь-якого екологічного моделювання (наприклад, в AERMOD) є розрахунок концентрації в кожній точці розрахункової сітки. Для отримання детальної карти поля забруднення необхідно виконати розрахунки для тисяч або десятків тисяч точок. У нашому підході цільова функція f_{obj} обчислюється лише для обмеженої множини ключових рецепторів (наприклад, 10-50 точок – шкіл, лікарень, житлових масивів). Це дозволяє скоротити час виконання одного циклу моделювання на порядки (наприклад, з хвилин до секунд), що є критичним для ітеративних методів оптимізації, які вимагають тисяч запусків моделі.

В свою чергу, перехід від нескінченного неперервного простору координат x, y , до скінченної множини дискретних локацій D (наприклад, 50 доступних розташувань) дозволяє уникнути складнощів, пов'язаних з локальними мінімумами у складних топологічних ландшафтах. Це робить задачу більш структурованою і дозволяє застосовувати ефективні комбінаторні алгоритми (GA, Discrete ABC) або навіть повний перебір для малих розмірностей.

Це також забезпечує практичне спрямування результатів оптимізації. У реальному міському плануванні нові об'єкти (котельні, КГУ) не можна

розмістити у довільній точці простору (наприклад, посеред парку, дороги чи приватної власності). У запропонованому підході множина D формується заздалегідь з урахуванням усіх містобудівних, земельних та технічних обмежень (наявність комунікацій, вільна ділянка, допустимість забудови). Рішення, яке отримується на основі оптимізації гарантовано є легальним та технічно можливим для реалізації, на відміну від неперервної оптимізації, яка може запропонувати точку в недоступному місці.

Окремо також слід відмітити переваги оцінювання в точках-рецепторах множини R прямо відповідає меті екологічної безпеки – захисту здоров'я людей. Замість абстрактного «зменшення середнього забруднення по місту», система гарантує дотримання нормативів саме там, де це найбільш важливо – у зонах проживання та перебування вразливих груп населення.

Це також забезпечує підвищення точності та надійності прийняття рішень, зокрема, уникнення помилок інтерполяції. Оскільки, при побудові неперервних полів концентрацій часто використовуються методи інтерполяції між вузлами сітки, що може вносити похибки. Розрахунок безпосередньо в точках рецепторів дає точні модельні значення для конкретних об'єктів.

Використання дискретної множини варіантів дозволяє особі, що приймає рішення, чітко порівнювати альтернативи (наприклад, «Варіант А на території котельні кращий за Варіант Б на пустирі»), а не намагатися інтерпретувати незначні зміщення координат.

Таким чином, відмова від побудови повних полів концентрацій на етапі оптимізації та перехід до дискретного вибору локацій з оцінкою в рецепторах є стратегічно виправданим рішенням. Воно трансформує задачу з теоретично складної та обчислювально складної у практично розв'язувану інженерну задачу, дозволяючи реалізувати інтерактивний сценарний аналіз у режимі прийнятного часу.

Щодо методів розв'язування задачі оптимізації розміщення нових джерел забруднення, яка є NP-складною задачею комбінаторної оптимізації. Через неможливість застосування повного перебору при $k > 1$ (NP-

складність) та класичних градієнтних методів (через природу природу цільової функції, яка є «чорною скринькою»), метаевристичні алгоритми є єдиним практично придатним інструментарієм для вирішення поставленої задачі. Метаевристичні алгоритми, зокрема їхні версії, адаптовані для дискретної комбінаторної оптимізації, такі як генетичний алгоритм (GA) або дискретний алгоритм штучної бджолоїної колонії (Discrete ABC), є безальтернативною обчислювальною основою, яка здатна знаходити квазіоптимальні рішення у NP-складній, обчислювально дорогій задачі оптимізації, оскільки середній час оцінки цільової функції, який обумовлений часом роботи модуля AERMOD, складає 3 с [6, 16, 41, 48, 49, 50].

Для метаевристичних алгоритмів (GA, ABC, PSO), найкращим способом врахувати обмеження є модифікація цільової функції методом штрафних функцій. Це дозволить алгоритму уникати рішення, які перевищують ГДК і є неприпустимими, навіть якщо вони математично мінімальні. Ми змінюємо функцію f_{obj} так, щоб вона складалася з двох частин: власне мінімізації забруднення та штрафу за порушення санітарних норм:

$$f_{obj}(X) = f_{risk}(X) + P(X), \quad (3.12)$$

де

$f_{risk}(X)$ – складова мінімізації ризику на основі індексу якості:

$$f_{risk}(X) = \max_{r \in R} \frac{C_{r,p}(X)}{\text{ГДК}_p}, \quad (3.13)$$

$P(X)$ – штрафна функція за порушення обмежень:

$$P(X) = \mu \cdot \sum_{r=1}^R \max \left(0, \max_{r \in R} \frac{C_{r,p}(X)}{\text{ГДК}_p} - 1 \right)^2, \quad (3.14)$$

де μ – штрафний коефіцієнт (наприклад, 10^4).

Функція $\max \left(0, \max_{r \in R} \frac{C_{r,p}(X)}{\Gamma \text{ДК}_p} - 1 \right)^2$ гарантує, що штраф нараховується тільки якщо концентрація перевищує $\Gamma \text{ДК}_p$. Якщо концентрація менша за $\Gamma \text{ДК}_p$, то доданок дорівнює 0.

Запишемо задачу оптимізації із врахуванням вище наведених міркувань:

$$X^* = \arg \min_{X \in S_D} \left[\max_{r \in R} \frac{C_{r,p}(X)}{\Gamma \text{ДК}_p} + \mu \cdot \sum_{r=1}^R \max \left(0, \max_{r \in R} \frac{C_{r,p}(X)}{\Gamma \text{ДК}_p} - 1 \right)^2 \right]. \quad (3.15)$$

Цільова функція трансформується до вигляду мінімізації відносного показника забруднення. На відміну від попереднього запису, константний зсув «-1» вилючено, оскільки операція віднімання константи не впливає на положення точки мінімуму X^* , але використання невід'ємних значень індексу якості покращує збіжність метаевристичних алгоритмів (зокрема, операторів селекції в GA).

Таким чином, запропонований у роботі підхід є гібридним. Він використовує достовірність сертифікованого модуля AERMOD, але, на відміну від стандартної практики, вбудовує його як «чорну скриньку» в оптимізаційний метод. А на відміну від відомих неперервних оптимізаційних моделей, постановка задачі є дискретною, що робить задачу практично розв'язуваною завдяки обчислювальній сфокусованості (розрахунок значень цільової функції лише в точках розміщення рецепторів).

3.2. Уніфікація програмної реалізації операторів метаевристичних алгоритмів для задачі дискретної оптимізації

Вирішення сформульованої NP-складної задачі комбінаторної оптимізації (вибір k найкращих локацій з m доступних) в умовах високої обчислювальної вартості цільової функції вимагає застосування ефективних метаевристичних алгоритмів.

Для дослідження ефективності їх застосування було обрано фреймворк DEAP (Distributed Evolutionary Algorithms in Python) [11]. DEAP виступає не лише засобом реалізації, а й ефективним середовищем для порівняльного аналізу ефективності різних метаевристичних стратегій у контексті задач оптимізації. Вибір DEAP як технологічної основи зумовлений необхідністю вирішення двох специфічних проблем:

- по-перше, задача вимагає пошуку підмножини фіксованого розміру k . Більшість стандартних бібліотек оптимізації (напр., `scipy.optimize`, `pyswarms`) орієнтовані на неперервні або прості бінарні простори і не підтримують обмеження на кардинальність множини ($|X| = k$). DEAP є не просто бібліотекою алгоритмів, а гнучким фреймворком, що дозволяє довільно визначати структуру «особини» (наприклад, як перестановку індексів або множину) та розробляти користувацькі оператори мутації та схрещування, специфічні для даної предметної області.

- по-друге, оскільки обчислення цільової функції (3.4) передбачає запуск зовнішнього модуля AERMOD (час виконання, якого може коливатися в середньому в діапазоні 3-5 с), послідовне виконання алгоритму робить його непридатним для інтерактивного використання. DEAP архітектурно спроектований для розподілених обчислень. Він дозволяє тривіально замінити стандартний оператор *map* на його паралельну версію (на основі *multiprocessing*), що забезпечує лінійне прискорення розрахунків пропорційно кількості ядер процесора без зміни логіки самого алгоритму.

Адаптивні оператори для дискретних метаевристик

Класичні метаевристичні алгоритми, такі як метод рою часток (PSO) та алгоритм бджолоїної колонії (ABC), були розроблені для неперервних просторів, де поняття «швидкості» та «координат» мають чіткий геометричний зміст. Їхнє застосування до нашої дискретної комбінаторної задачі вимагає адаптації. Існуючі «бінарні» версії цих алгоритмів (BPSO, BABC) працюють з бінарними векторами довільної ваги, що не гарантує дотримання обмеження на кількість обраних джерел, k . Щоб уникнути

використання складних механізмів «штрафів» або «ремонту» невалідних рішень, у даній роботі запропоновано підхід на основі уніфікованих дискретних операторів.

Ці оператори перевизначають поняття «руху» у дискретному просторі рішень, дозволяючи реалізувати логіку різних метаевристик (GA, PSO, ABC) в єдиній термінології.

Формалізуємо позначення простору та операторів. Нехай, $D_{locations} = \{d_1, d_2, \dots, d_m\}$ – множина всіх доступних локацій. Рішення (позиція) v визначається як підмножина $v \subseteq D_{locations}$, така що $|v| = k$.

Для адаптації базових версій алгоритмів розроблено три базові оператори:

1. `Create_Random_Solution()` – оператор ініціалізації, який забезпечує глобальне дослідження (exploration) та буде використовуватися для генерування початкової популяції та в фазі «бджіл-розвідників» для алгоритму D-ABC. Даний оператор повертає випадкову вибірку k унікальних елементів з множини $D_{locations}$ (рис. 3.1).

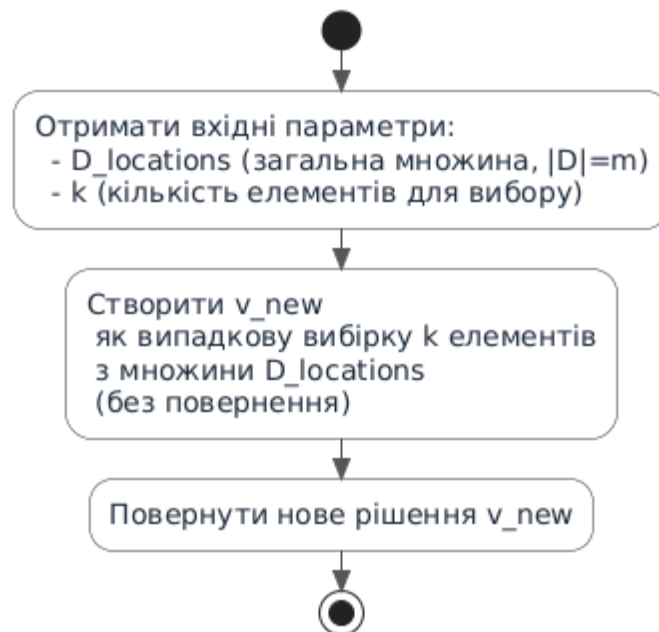


Рис. 3.1. Діаграма активності для оператора `Create_Random_Solution()`

2. Random_Swap() – оператор випадкової заміни, який реалізує ненапрямлений локальний пошук або мутацію та створює «сусіда» для поточного рішення (рис. 3.2). Даний оператор вибирає компонент, що видаляється, випадковий елемент $d_{out} \in v$ та компонент, що додається, як випадковий елемент:

$$d_{in} \in D_{locations} \setminus v. \quad (3.16)$$

Далі формується нове рішення:

$$v_{new} \leftarrow (v \setminus d_{out}) \cup d_{in}. \quad (3.17)$$

Використовується як оператор мутації в GA та як основний механізм пошуку для «робочих бджіл» в D-ABC.

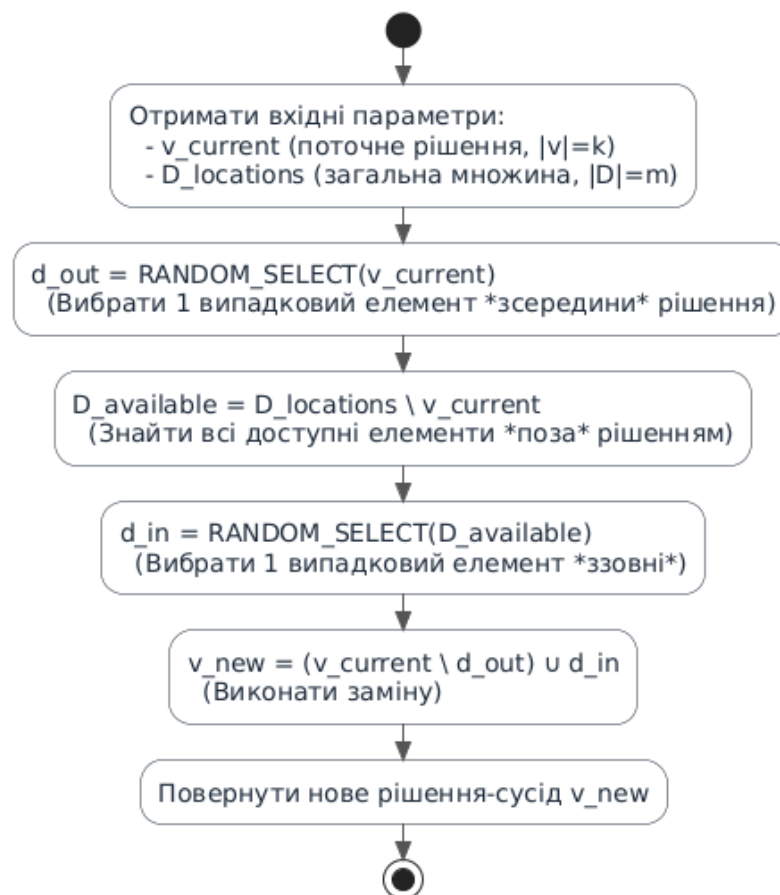


Рис. 3.2. Діаграма активності для оператора Random_Swap()

3. Targeted_Swap() – оператор напрямленої заміни, який реалізує напрямлений рух до кращого рішення, що є аналогом вектора швидкості в PSO. Оператор визначає різницю між поточним рішенням $v_{current}$ та цільовим v_{target} , формуючи множини зайвих елементів

$$Diff_{out} = v_{current} \setminus v_{target} \quad (3.18)$$

та множину відсутніх елементів

$$Diff_{in} = v_{target} \setminus v_{current}. \quad (3.19)$$

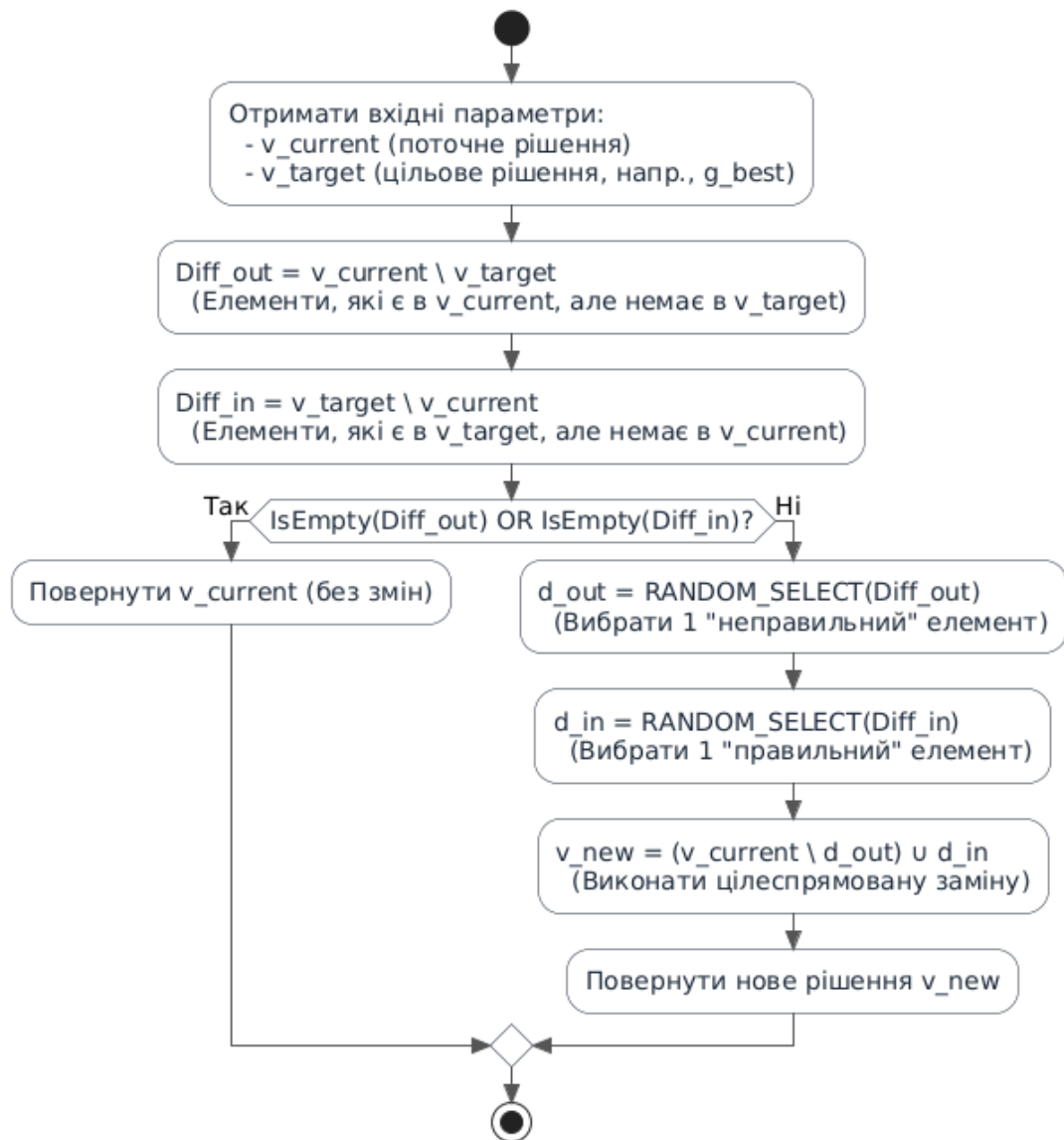


Рис. 3.3. Діаграма активності для оператора Targeted_Swap()

Якщо множини ідентичні, рух не виконується. У випадку неспівпадання вибираються $d_{out} \in Diff_{out}$ та $d_{in} \in Diff_{in}$ та формується нове рішення:

$$v_{new} \leftarrow (v_{current} \setminus d_{out}) \cup d_{in}. \quad (3.20)$$

Даний оператор дозволяє адаптувати алгоритм PSO, імітуючи «притягування» частки до персонального (p_{best}) та глобального (g_{best}) кращих рішень шляхом поступової заміни відмінних елементів.

Отже, розробка запропонованих операторів дозволяє уніфікувати програмну реалізацію різних метаевристик у середовищі DEAP, забезпечуючи при цьому суворе дотримання обмежень задачі (кардинальність множини k) на кожному кроці алгоритму без необхідності додаткових перевірок (рис. 3.4).

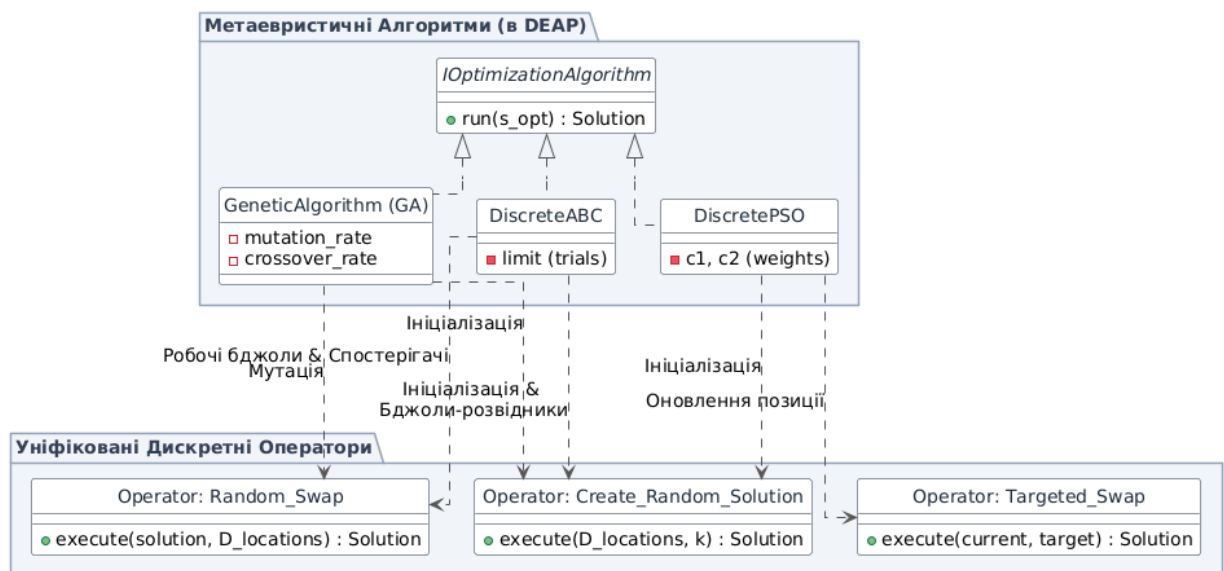


Рис. 3.4. Використання уніфікованих операторів в архітектурі DEAP

Таблиця 3.1 ілюструє фундаментальні відмінності в поведінці трьох адаптованих алгоритмів, що обумовлені їхніми внутрішніми механізмами пошуку.

Робота GA базується на еволюційних принципах (рис. 3.5). Оператор схрещування комбінує частини двох успішних батьківських рішень, сподіваючись отримати ще кращого нащадка. Оператор мутації

(Random_Swap()) вносить випадкові зміни для підтримки різноманітності. GA не має механізму прямого «руху до цілі». Він покладається на стохастичний відбір кращих особин з покоління в покоління, що є повільнішим процесом, ніж цілеспрямований рух часток у PSO.

Таблиця 3.1.

Якісне порівняння адаптованих метаевристик

Алгоритм	Основний механізм	Швидкість збіжності	Здатність до глобального пошуку	Ризик передчасної збіжності
GA	Схрещування / Мутація (Random_Swap())	Середня	Середня (через мутацію)	Середній
D-PSO	«Притягання» до g_best (Targeted_Swap())	Висока	Низька	Високий
D-ABC	Локальний пошук (Random_Swap()), «Розвідники» (Create_Random())	Середня	Висока (через розвідників)	Низький

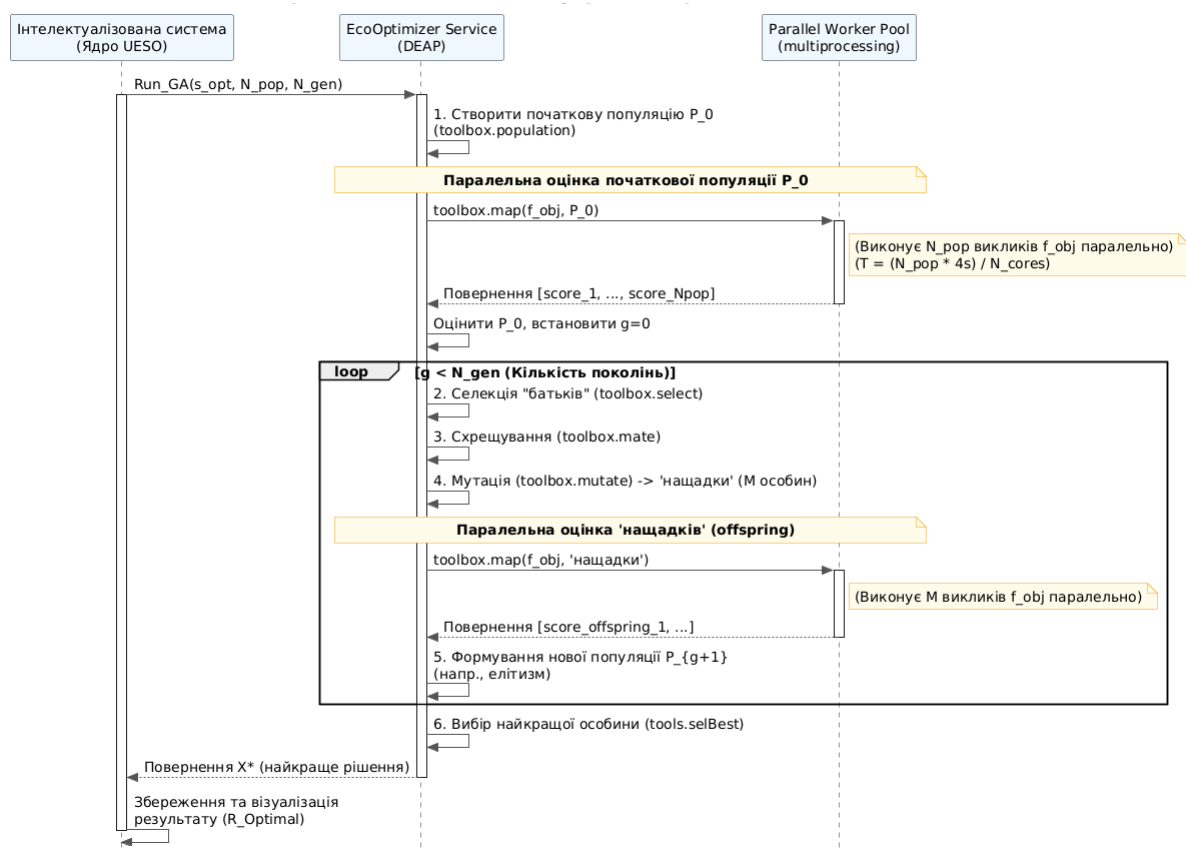


Рис. 3.5. Діаграма послідовності роботи адаптованого GA з механізмом паралелізації обчислень в DEAP

Глобальний пошук забезпечується виключно оператором мутації. Якщо мутація занадто слабка, популяція може виродитися; якщо занадто сильна – алгоритм перетворюється на випадковий пошук. GA менш схильний до застрягання, ніж PSO, але при складних ландшафтах вся популяція може прийти в один локальний оптимум.

D-PSO – це алгоритм «соціальної поведінки» (рис. 3.6). Кожна частка має «пам'ять» про свій найкращий результат (p_{best}) і знає про найкращий результат усього рою (g_{best}). Оператор Targeted_Swap() змушує частку активно змінювати свою позицію, щоб стати схожою на лідерів. Це імітує «вектор швидкості». Швидкість збіжності найсильніша сторона D-PSO. Завдяки механізму притягання, весь рій дуже швидко рухається в зону найкращого знайденого рішення. При цьому D-PSO майже не має механізмів для дослідження нових областей, коли рій вже згрупувався. Якщо g_{best} (глобальний лідер) потрапив у локальний оптимум, то алгоритм зупиниться в субоптимальному рішенні.

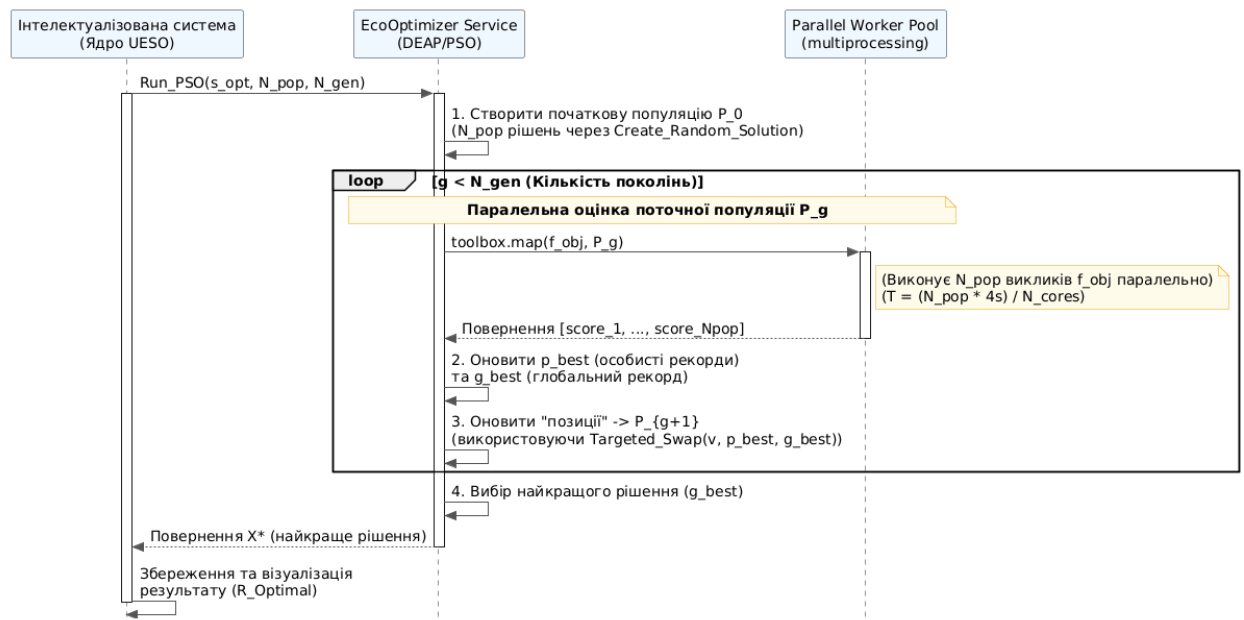


Рис. 3.6. Діаграма послідовності роботи дискретного PSO з механізмом паралелізації обчислень в DEAP

Унікальність D-ABC полягає в розділенні ролей (рис. 3.7). «Робочі бджоли» займаються локальним покращенням відомих рішень (Random_Swap()).

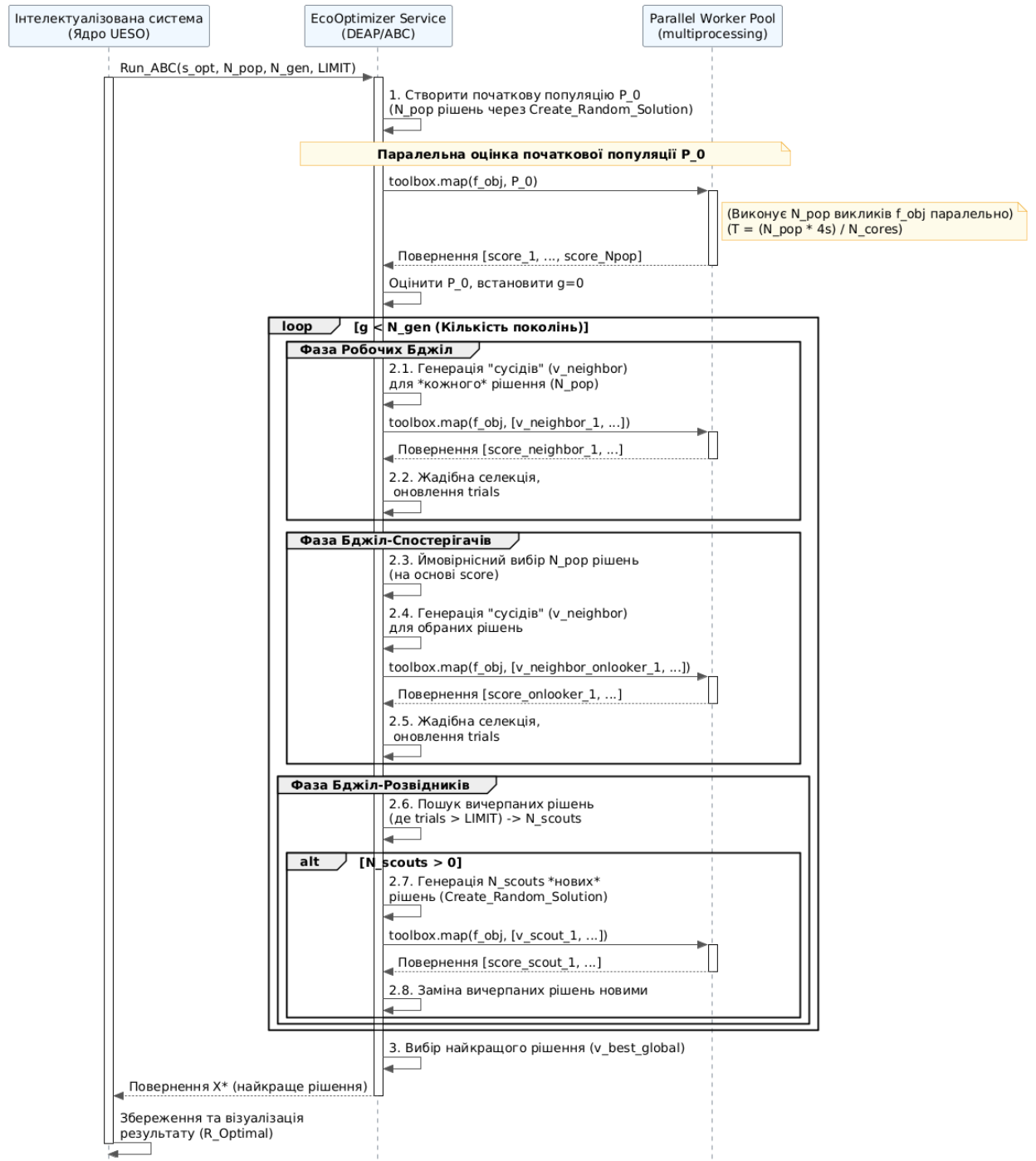


Рис. 3.7. Діаграма послідовності роботи дискретного ABC з механізмом паралелізації обчислень в DEAP

«Розвідники» відповідають за глобальний пошук. Якщо рішення не покращується певний час (LIMIT), воно викидається, і на його місце генерується абсолютно нове, випадкове рішення (Create_Random()). Через постійне введення нових випадкових рішень (розвідниками), середня якість популяції зростає повільніше, ніж у D-PSO.

Механізм «розвідників» гарантує, що алгоритм ніколи не припиняє шукати нові області, навіть якщо основна частина популяції вже знайшла рішення. Це робить його найбільш надійним (робастним) для складних, багатомодальних задач високої розмірності.

Ключовою перевагою використання DEAP є можливість розпаралелювання популяційних алгоритмів. При використанні 8-ядерного Parallel Worker Pool ($N_{cores} = 8$) час виконання одного покоління (напр., $N_{pop} = 50$) скорочується з $T = 50 * 4 = 200$ с до $T_P = \frac{50}{8} * 4 = 28$ с (прискорення ~ 7.1 x).

3.3. Метод інтелектуалізованого вибору стратегії оптимізації залежно від складності оптимізаційної задачі

Обґрунтування необхідності інтелектуалізованого вибору стратегії оптимізації базується на аналізі трьох ключових факторів: NP-складності задачі, високій обчислювальній складності цільової функції та різній ефективності алгоритмів у різних просторах пошуку.

По-перше, експоненційне зростання простору пошуку. Як вже зазначалося, задача розміщення k джерел серед m локацій належить до класу NP-складних задач комбінаторної оптимізації. Розмір простору пошуку S_D визначається біноміальним коефіцієнтом, який зростає факторіально. При невеликих розмірностях (наприклад, $m = 10, k = 5$) простір пошуку складає лише 252 варіанти. Тут застосування складних метаевристик є надлишковим, повний перебір гарантує ідеальний результат за прийнятний час, 2.1 хв. (табл.

3.1). При збільшенні параметрів до $m = 30, k = 15$, кількість варіантів сягає $1.55 \cdot 10^8$. Повний перебір стає фізично неможливим, що вимагає переходу до наближених методів (метаевристик). Отже, один і той самий метод (наприклад, повний перебір) не може бути застосований для всіх сценаріїв через «комбінаторний вибух».

По-друге, висока обчислювальна складність цільової функції. На відміну від класичних задач, функція f_{obj} є «чорною скринькою» (виклик модуля AERMOD). Середній час одного розрахунку може складати близько 4 секунд. Це робить «вартість» кожної ітерації надзвичайно високою. Інтелектуалізований вибір алгоритму дозволяє мінімізувати кількість звернень до AERMOD, обираючи стратегію, яка збігається найшвидше для конкретного розміру задачі.

Таблиця 3.1.

Результати дослідження ефективності алгоритмів

№	Доступні локації, m	Джерела, k	Розмірність задачі, C	Обраний алгоритм	Час, хв
1	10	5	252	Exhaustive	2.1
2	14	5	2,002	Exhaustive	6.5
3	16	6	8,008	D-PSO	12.4
4	18	6	18,564	D-PSO	23.4
5	22	8	319,770	D-ABC	48.2
6	30	15	1.55×10^8	D-ABC	98.0

Проведені дослідження (рис. 3.6) показали, що жодна метаевристика не є універсальною: D-PSO забезпечує високу швидкість збіжності завдяки направленому руху, проте має високий ризик зупинки в локальних оптимумах, особливо на складних ландшафтах («передчасна збіжність»). Він ефективний для задач середньої складності, де важлива швидкість. В свою чергу D-ABC є

більш повільним, але значно надійнішим (робастним) завдяки механізму «бджіл-розвідників», які постійно досліджують нові області та не дають алгоритму зациклитися. Це єдиний вибір для задач високої складності (стратегічного рівня), де точність важливіша за час.

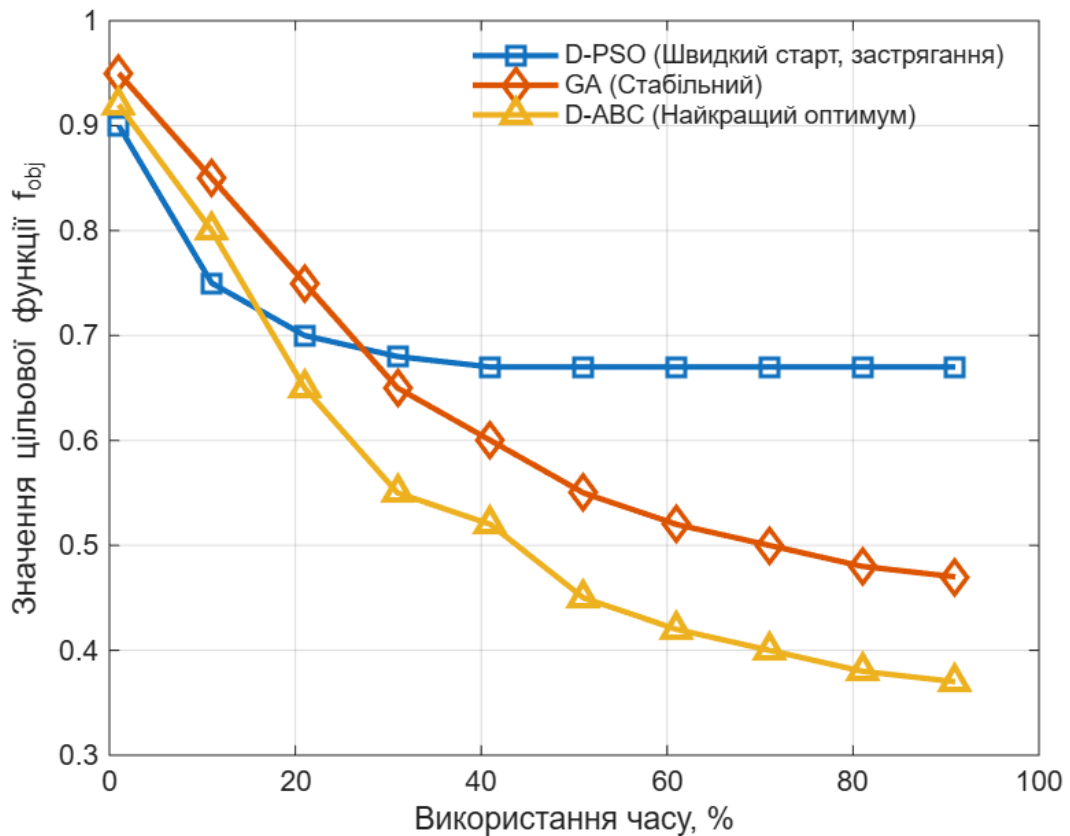


Рис. 3.8. Аналіз збіжності алгоритмів

Представимо логіку вибору алгоритму оптимізації у вигляді кортежної структури $\mathfrak{S}$:

$$\mathfrak{S} = \langle S, A, C, P, \Phi \rangle, \quad (3.21)$$

де S – простір пошуку рішень:

$$S = \{ \langle m, k \rangle \in \mathbb{N}^2 \mid 1 \leq k \leq m \}, \quad (3.22)$$

де

m – кількість доступних локацій (ueso:Locations),

k – кількість джерел, розміщення яких оптимізується (ueso:Sources);

A – множина алгоритмів:

$$A = \{a_{ES}, a_{DPSO}, a_{DABC}\}, \quad (3.23)$$

де

a_{ES} – алгоритм, що реалізує повний перебір (Exhaustive Search),

a_{DPSO} – алгоритм, що реалізує евристику на основі дискретної версії рою частинок для сценаріїв тактичного значення (середня складність),

a_{DABC} – алгоритм, що реалізує евристику на основі дискретної версії бджолиної колонії для сценарії стратегічного значення (висока складність);

C – множина констант:

$$C = \{\tau_{comp}, \tau_{scale}\}, \quad (3.24)$$

де

$\tau_{comp} = 4000$ – поріг складності,

$\tau_{scale} = 20$ – поріг масштабу;

P – множина предикатів:

$$p_{triv}(s): C(m, k) < \tau_{comp}, \quad (3.25)$$

$$p_{tact}(s): m < \tau_{scale}, \quad (3.26)$$

$$p_{strat}(s): m \geq \tau_{scale}; \quad (3.27)$$

Φ – функціональне відображення

$$\Phi: S \rightarrow A, \quad (3.28)$$

де

$$\Phi(m, k) = \begin{cases} a_{ES}, & \text{якщо } p_{triv}(m, k) = 1, \\ a_{PSO}, & \text{якщо } \neg p_{triv} \wedge p_{tact}(m) = 1, \\ a_{ABC}, & \text{якщо } \neg p_{triv} \wedge p_{strat}(m) = 1. \end{cases} \quad (3.29)$$

Необхідність введення порогів масштабу τ_{scale} та складності τ_{comp} обумовлена потребою формалізувати процес прийняття рішень системою провибір алгоритму. Оскільки обчислення цільової функції є дорогим, система не може використовувати один універсальний метод. Ці пороги діють як «перемикачі» в логіці вибору алгоритму, балансуючи між часом виконання та гарантією знаходження оптимуму.

Поріг складності τ_{comp} визначає межу між «тривіальними» задачами та задачами, що вимагають евристичного підходу. Задача класифікується як тривіальна p_{triv} . У цьому випадку простір пошуку настільки малий, що застосування метаевристик є недоцільним. Система обирає повний перебір a_{ES} .

Поріг масштабу τ_{scale} використовується для диференціації NP-складних задач за рівнем складності «ландшафту» пошуку (рис. 3.7). Він розділяє задачі на тактичні та стратегічні.



Рис. 3.9. Ілюстрація порогу масштабу у виборі стратегії оптимізації

Коли задача стає занадто великою для повного перебору, система аналізує кількість доступних локацій m . При $m < 20$ простір пошуку є

помірним (наприклад, ~ 18 тис. варіантів). PSO має високу швидкість збіжності і здатний знайти прийнятне рішення за 12-23 хвилини. Ризик зупинки в локальному мінімумі існує, але на таких розмірностях він виправданий швидкістю. Якщо $m \geq \tau_{scale}$, встановлюється стратегічний рівень $p_{strat}(m) = 1$ та використовується дискретний ABC, a_{DABC} . Введення цих порогів дозволяє реалізувати функцію відображення $\Phi(m, k)$, яка автоматично адаптує обчислювальну стратегію.

Тривимірне відображення логарифмічної функції складності $z = \log_{10}(C(m, k))$ наведено на рис. 3.10. Кольорові зони відповідають оптимальним алгоритмічним стратегіям згідно з онтологічними правилами.

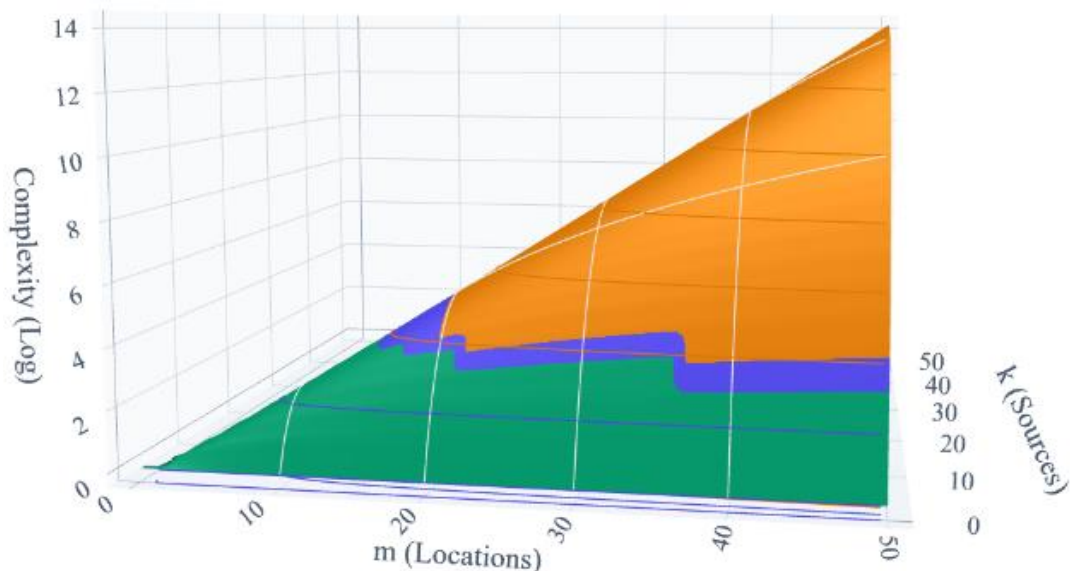


Рис. 3.10. Ілюстрація вибору алгоритмів у просторі пошуку: зона зеленого кольору – алгоритм повного перебору, зона синього кольору – дискретний PSO, зона жовтого кольору – дискретний ABC

3.4. Апробація методу адаптивного вибору стратегії оптимізації

Для апробації розробленого методу проведемо сценарного аналіз розміщення нових джерел забруднення – об’єктів теплогенерації, на прикладі м. Тернопіль. У 2023-2024 роках Тернопіль отримав партію нових

когенераційних установок від USAID (Проект енергетичної безпеки). Це мобільні установки, які встановлюються на локальних котельнях (школи, лікарні, мікрорайони). Для експерименту «розміщення нових джерел» було взято усереднені характеристики цих нових блоків:

Тип: Міні-ТЕС контейнерного типу (часто на базі двигунів MAN або MWM, аналог Jenbacher J208).

Електрична потужність: ~50-500 кВт (різні модифікації). Для моделювання прийнято 330 кВт (середнє значення для локальних котелень).

Таблиця 3.2.

Вхідні дані про джерела для модуля AERMOD

Параметр в AERMOD	Опис	Значення для JMS 312	Значення для USAID (330 кВт)*
emission_rate	Потужність викиду	0.4 г/с	0.25 г/с
stack_height	Висота труби	12.0 м (мін. для міста)	8.0 м (моб. блок)
stack_diameter	Діаметр гирла	0.35 м	0.25 м
exit_velocity	Швидкість газів	22.0 м/с	18.0 м/с
exit_temperature	Температура газів	723 К (450°C)	723 К (450°C)

У реальному міському плануванні когенераційні установки (КГУ) контейнерного типу можна встановлювати не лише на території діючих котелень, але й на інших комунальних майданчиках, де є доступ до інфраструктури (газ/електромережа): трансформаторні підстанції (ТП), території закладів освіти (госп. двір), комунальні/промислові майданчики: гаражні кооперативи, стоянки, пустирі поруч з теплотрасами.

Множина локацій можливого розміщення (табл. 3.3):

$$D = \{d_1, d_2, \dots, d_{16}\}.$$

До множини рецепторів включено об'єкти з високим пріоритетом захисту (школи, садочки) та точки щільної житлової забудови, які знаходяться в радіусі дії нових локацій (зокрема ТП та гаражів). Множина R містить 12 об'єктів (табл. 3.4):

$$R = \{r_1, r_2, \dots, r_{12}\}.$$

Таблиця 3.3

Характеристики локацій можливого розміщення джерел

ID	Адреса (Орієнтир)	Тип об'єкта	Координати (Lat, Lon)	Обґрунтування
d ₁	вул. Київська, 3с	Котельня (Велика)	49.5621, 25.6285	Базова точка генерації
d ₂	вул. Л. Українки, 4	Котельня (Квартальна)	49.5605, 25.6350	Існуюча інфраструктура
d ₃	вул. 15 Квітня, 20А	ЦТП / Котельня	49.5682, 25.6321	Густонаселений район
d ₄	вул. Купчинського, 14	Лікарня №2 (Котельня)	49.5658, 25.6420	Критична інфраструктура
d ₅	вул. Корольова, 7	Промзона	49.5580, 25.6405	Мінімум скарг на шум
d ₆	вул. Симоненка, 5	ЦТП	49.5645, 25.6201	Спальний район
d ₇	вул. 15 Квітня, 6	Трансф. підстанція	49.5665, 25.6380	Доступ до електромережі
d ₈	вул. Київська, 11	Трансф. підстанція	49.5642, 25.6245	Внутрішній двір, є місце
d ₉	бул. П. Куліша, 1	Трансф. підстанція	49.5685, 25.6270	Поруч з висотками
d ₁₀	вул. В. Великого, 4	Школа №19 (Госп. двір)	49.5615, 25.6305	Велика огорожена територія
d ₁₁	вул. Л. Курбаса, 9	Спортмайданчик/Пустир	49.5635, 25.6335	Вільна зона біля "Сільпо"
d ₁₂	вул. Овочева, 11	Склади/Гаражі	49.5590, 25.6450	Віддалено від житла
d ₁₃	вул. Тарнавського, 22	Технічний коледж (Тиль)	49.5638, 25.6180	Освітній заклад, є місце
d ₁₄	вул. Морозенка, 5	Парковка ТРЦ	49.5690, 25.6240	Асфальтований майданчик
d ₁₅	вул. Стуса, 10	ЦТП (Старий)	49.5655, 25.6355	Потребує реконструкції
d ₁₆	вул. Байди Вишневецького	Гаражний кооператив	49.5575, 25.6290	Промзона в житловому масиві

Віртуально встановлюємо установку типу Jenbacher JMS 312 (або аналог USAID 330 кВт). Забруднювачем є оксиди азоту (NO_x), оскільки для газових двигунів це єдиний викид, що може наблизитися до $\text{ГДК} = 0.2 \frac{\text{мг}}{\text{м}^3}$.

Відповідно необхідно знайти $k = 3$ оптимальні точки для встановлення нових КГУ так, щоб не перекрити викидами забруднення житлові будинки (рецептори), де вже є високий фон від транспорту.

Таблиця 3.4

Характеристики об'єктів-рецепторів

ID	Адреса (Об'єкт)	Тип	Координати (Lat, Lon)	Обґрунтування
r ₁	бул. П. Куліша, 9 (ЗОШ № 27)	Школа	49.5670, 25.6295	Центр масиву, поруч d ₉ , d ₁
r ₂	вул. Київська, 6 (Дитсадок № 21)	Садок	49.5630, 25.6260	Дуже близько до d ₁ , d ₈
r ₃	вул. Купчинського, 14 (Лікарня №2)	Лікарня	49.5660, 25.6435	В зоні впливу d ₄ , d ₃
r ₄	вул. Братів Бойчуків, 2 (ЗОШ № 19)	Школа	49.5610, 25.6310	В зоні d ₂ , d ₁₀
r ₅	вул. 15 Квітня, 25 (Житло)	Висотна забудова	49.5695, 25.6340	Ризик від d ₃ , d ₆
r ₆	вул. Тарнавського, 34 (Техколедж)	Навчальний	49.5640, 25.6150	В зоні d ₁₃
r ₇	вул. Корольова, 10 (Дитсадок № 35)	Садок	49.5595, 25.6390	Критично: поруч промзона d ₅
r ₈	бул. Вишневецького, 8 (ЗОШ № 28)	Школа	49.5630, 25.6210	В зоні нових джерел d ₆ , d ₁₆
r ₉	вул. Лесі Українки, 10 (Дитсадок № 14)	Садок	49.5612, 25.6365	Щільна забудова біля d ₂
r ₁₀	просп. Злуки, 55 (Церква Петра)	Громад. місце	49.5668, 25.6330	Перехрестя потоків від d ₁ , d ₃
r ₁₁	вул. Купчинського, 7 (ЗОШ № 26)	Школа	49.5675, 25.6460	Крайня східна точка d ₄
r ₁₂	вул. Симоненка, 2 (Житло)	Житло	49.5650, 25.6190	Західна межа району d ₆ , d ₁₃

Віртуально встановлюємо установку типу Jenbacher JMS 312 (або аналог USAID 330 кВт). Забруднювачем є оксиди азоту (NO_x), оскільки для газових двигунів це єдиний викид, що може наблизитися до ГДК = $0.2 \frac{\text{мг}}{\text{м}^3}$.

Відповідно необхідно знайти $k = 3$ оптимальні точки для встановлення нових КГУ так, щоб не перекрити викидами забруднення житлові будинки (рецептори), де вже є високий фон від транспорту.

Значення фонової концентрації NO_x в точці розміщення рецепторів, створене всіма існуючими (не оптимізованими) джерелами, наведено в табл. 3.5.

Таблиця 3.5

Значення фонових концентрацій для рецепторів

ID	$C_{r,p}^{base}$ (мг/м ³)	% від ГДК	Обґрунтування (Джерело фону)
r1	0.06	30%	Спальний район, далеко від дороги
r2	0.08	40%	Близькість до вул. Київської
r3	0.05	25%	Паркова зона лікарні
r4	0.07	35%	Всередині кварталу
r5	0.14	70%	Критично! 6-смугова магістраль під вікнами
r6	0.09	45%	Інтенсивний трафік на підйомі
r7	0.11	55%	Вплив промислової зони та гаражів
r8	0.04	20%	Глибоко у дворах
r9	0.05	25%	Затишна зона
r10	0.13	65%	Перехрестя великих потоків транспорту
r11	0.03	15%	Окраїна міста, чисте повітря
r12	0.06	30%	Середня активність

Згідно з розробленою методикою інтелектуалізованого вибору стратегії (параграф 3.3), для даного експерименту система автоматично обрала дискретний алгоритм рою частинок (a_{DPSO}).

На рис 3.9 зображено фрагмент карти мікрорайону «Сонячний»/«Східний» у м. Тернопіль (перетин просп. Злуки, вул. 15 Квітня та вул. Київської). Чорні точки – це всі 16 кандидатів доступних локацій (котельні, ЦТП, пустирі), які ми розглядали. Червоні зірки – це рецептори: чутливі об'єкти (школи №19, №27, №28, дитсадки, лікарня №2), де контролюється рівень забруднення. Зелені зірки – це три оптимальні локації, обрані алгоритмом для розміщення нових когенераційних установок (КГУ). нових джерел

Алгоритм згрупував джерела генерації на західному фланзі району. Локація №1 (Верхня ліва) – район перетину вул. Текстильна та просп. Злуки.

Це зона, наближена до промислових/комерційних об'єктів (наприклад, ТРЦ або склади). Локація №2 (Центральна ліва) – трохи східніше від першої, район біля парку Національного Відродження або прилеглих територій (пустир/спортмайданчик). Локація №3 (Нижня) – район вул. Симоненка/просп. Злуки.

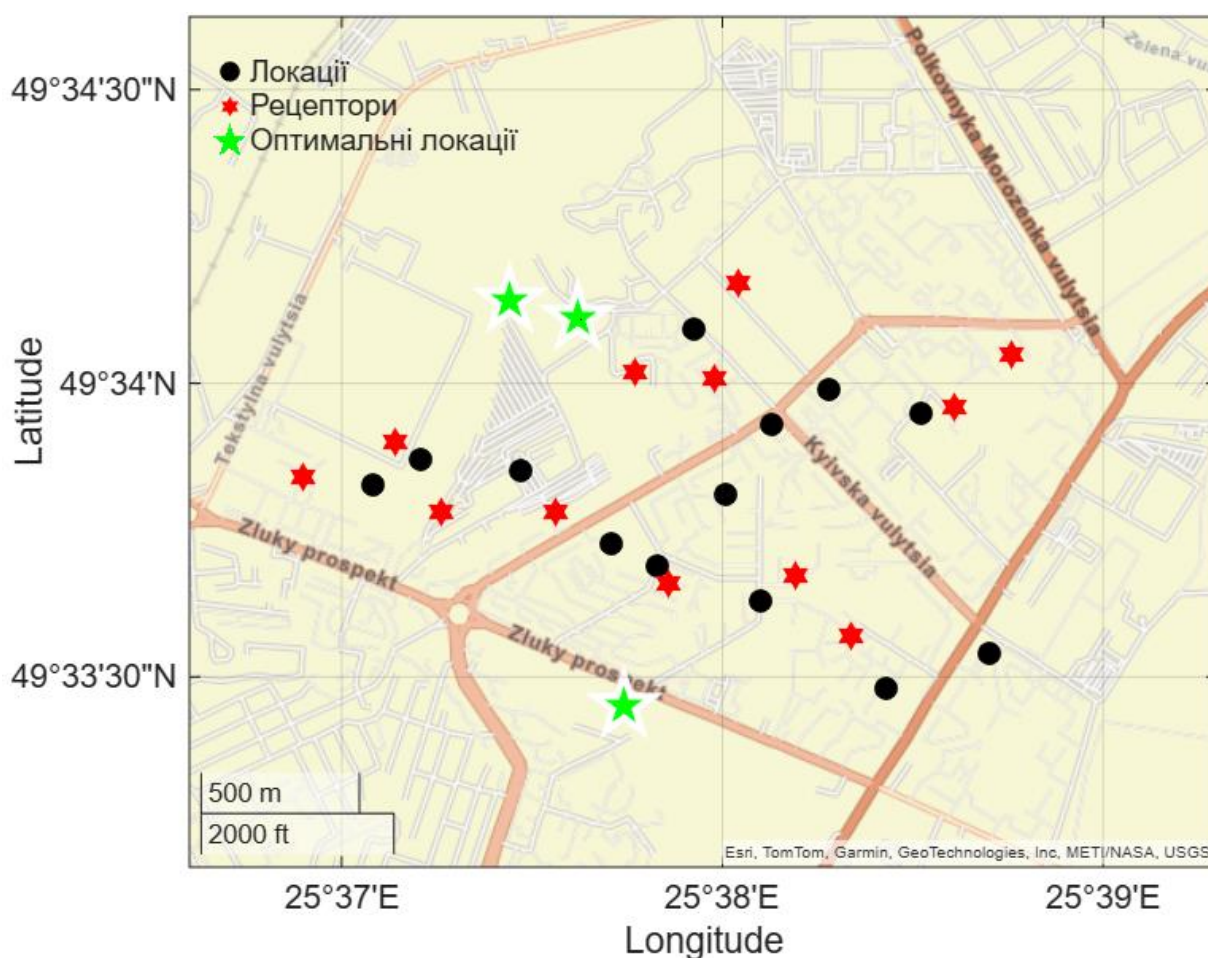


Рис. 3.11. Результати сценарного аналізу оптимального розміщення

Рецептори сконцентровані в Центрі та на Сході (вздовж вул. Київської та 15 Квітня). Це зона щільної житлової забудови. Джерела винесені максимально на Захід. Алгоритм знайшов рішення, яке мінімізує вплив на найбільше скупчення людей, фізично віддаливши труби від шкіл.

Результат свідчить про те, що фонове забруднення в центрі (на вул. Київській) настільки високе, що не можливо розмістити там жодного джерела. Джерела необхідно розміщувати на відносно чистому заході і дозволити

викидам розсіятися на відстані, ніж ставити їх у центрі, де навіть малий викид призведе до миттєвого перевищення ГДК. Обрані точки співпадають з зонами менш щільної забудови (паркові зони, широкі проспекти, початок промзони). Це дозволяє димовим газам краще розсіюватися завдяки відсутності ефекту «аеродинамічної тіні» від висотних будинків, що є характерним для вул. 15 Квітня.

Отриманий результат демонструє, що центральна частина району є екологічно перевантаженою. Розміщення нових потужностей на існуючих котельнях (вздовж вул. Київської) є недоцільним через ризик перевищення санітарних норм. Найбезпечнішим є створення «енергетичного вузла» на західному кордоні житлового масиву (вул. Текстильна/Симоненка), звідки теплоносій подаватиметься вглиб району. Метод успішно виконав розміщення на користь глобальної екологічної безпеки населення.

Висновки до розділу 3

1. Розроблено математичну постановку задачі комбінаторної оптимізації розміщення джерел техногенного забруднення. На відміну від класичних геометричних задач розміщення, запропонований підхід враховує суперпозицію полів концентрацій та використовує модель AERMOD як «чорну скриньку» для обчислення цільової функції. Для гарантування дотримання екологічних стандартів цільова функція модифікована методом штрафних функцій, що дозволяє мінімізувати максимальний ризик перевищення ГДК у точках-рецепторах. Перехід до дискретного простору пошуку дозволив уникнути проблем локальних мінімумів, характерних для неперервних ландшафтів, та забезпечити легальність пропонуваного рішення.

2. Адаптовано метаевристичні алгоритми (генетичний алгоритм, метод рою часток, алгоритм бджолої колонії) для вирішення задач дискретної оптимізації фіксованої розмірності. Розроблено та реалізовано у фреймворку DEAP три уніфіковані оператори: `Create_Random_Solution()`, `Random_Swap()`

та Targeted_Swap(), що дозволило уніфікувати програмну реалізацію різних евристик та гарантувати валідність рішень (дотримання кардинальності множини) без необхідності використання механізмів «адаптації» особин.

3. Запропоновано метод інтелектуалізованого вибору стратегії оптимізації, який базується на аналізі складності задачі. Введення порогів складності та масштабу дозволяє системі автоматично перемикатися між повним перебором (для тривіальних задач), D-PSO (для задач тактичного рівня) та D-ABC (для стратегічних задач високої розмірності). Це забезпечує баланс між часом отримання рішення та гарантією його оптимальності в умовах високої обчислювальної вартості звернень до моделі AERMOD.

4. Проведено апробацію методу на прикладі розміщення трьох когенераційних установок у м. Тернопіль. Експеримент показав, що центральна частина мікрорайону є екологічно перевантаженою через високий фоновий рівень забруднення. Алгоритм (у даному випадку – повний перебір, обраний системою автоматично) визначив оптимальну стратегію децентралізації генерації, запропонувавши розміщення джерел на західному фланзі району (вул. Текстильна, Симоненка), що дозволяє мінімізувати вплив на житлову забудову та навчальні заклади. Отримані результати підтверджують практичну цінність розробленого забезпечення для підтримки прийняття рішень у сфері енергетичного планування та екологічної безпеки.

РОЗДІЛ 4.

ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛІЗОВАНОЇ СИСТЕМИ ПРОАКТИВНОГО УПРАВЛІННЯ ЕКОСИСТЕМОЮ МІСТА

У розділі викладено результати проєктування, програмної реалізації та експериментальної апробації інтелектуалізованої системи проактивного управління екосистемою міста «Есо-City». Основною метою цього етапу роботи є створення цілісного програмного інструментарію, що реалізує розроблені у попередніх розділах онтологічні моделі, методи адаптивної оптимізації та алгоритми сценарного аналізу.

Особливу увагу в розділі приділено обґрунтуванню та побудові сервіс-орієнтованої архітектури, яка забезпечує гнучку інтеграцію гетерогенних компонентів системи: підсистеми математичного моделювання, що базується на регуляторному ядрі AERMOD, підсистеми управління знаннями, побудованої на основі розробленої онтології UESO, та модулів геопросторової аналітики.

Структура розділу відображає послідовність етапів розроблення системи та охоплює ключові аспекти проєктування, конструювання та тестування.:

Розглянуто запропоновані основні архітектурні рішення. Деталізовано принципи декомпозиції системи на незалежні мікросервіси та застосування структурного патерну «Wrapper» (обгортка/адаптер) для автоматизації взаємодії з компонентами розрахункового ядра. Описано організацію інформаційних потоків через стандартизовані REST-інтерфейси, що забезпечує масштабованість системи.

Наведено спроектоване інформаційне забезпечення. Розкрито особливості програмної реалізації концепції «єдиного джерела істини» з використанням стека технологій Semantic Web (RDF, OWL 2, SPARQL). Описано побудову гібридної інфраструктури зберігання даних, яка поєднує

семантичні графові сховища, реляційні бази даних з просторовим розширенням (PostGIS) та файлові сховища результатів моделювання.

Представлено реалізацію інтерактивного веб-інтерфейсу користувача, орієнтованого на зниження когнітивного навантаження диспетчера. Описано інструменти візуалізації онтологічних графів, генерації теплових карт забруднення та побудови ізоліній концентрацій.

Наведено результати функціонального та навантажувального тестування системи. Ефективність розробленого рішення підтверджено на прикладі вирішення практичної задачі оптимізації розміщення енергогенеруючих потужностей у м. Тернопіль, що демонструє переваги запропонованого підходу порівняно з традиційними засобами проектування.

Основні результати цього розділу опубліковані в працях [101, 102, 103, 115, 125, 126].

4.1 Архітектура програмної системи для проактивного управління екосистемою міста

Перехід від реактивного до проактивного управління міською екосистемою висуває принципово нові вимоги до програмних систем підтримки прийняття рішень. Проактивне управління, що має на меті прогнозування, попередження та оптимізацію майбутніх станів екосистеми, вимагає архітектури, що за своєю природою є гнучкою, масштабованою та здатною до інтеграції складних обчислювальних та інтелектуалізованих компонентів. Традиційні монолітні архітектури, де вся бізнес-логіка, доступ до даних та інтерфейс користувача тісно пов'язані в єдиному застосунку, є фундаментально непридатними для вирішення цієї задачі. Вони є крихкими, складними в модифікації та не дозволяють ефективно інтегрувати гетерогенні обчислювальні модулі. Запропонована в даній роботі сервіс-орієнтована архітектура (рис. 4.1) є не просто технічним вибором, а стратегічною необхідністю, що обумовлена ключовими вимогами проактивного управління.

Вона спроектована за сучасними принципами сервіс-орієнтованої архітектури (SOA) та мікросервісів, що забезпечує максимальну гнучкість, масштабованість та можливість незалежної розробки кожної підсистеми.

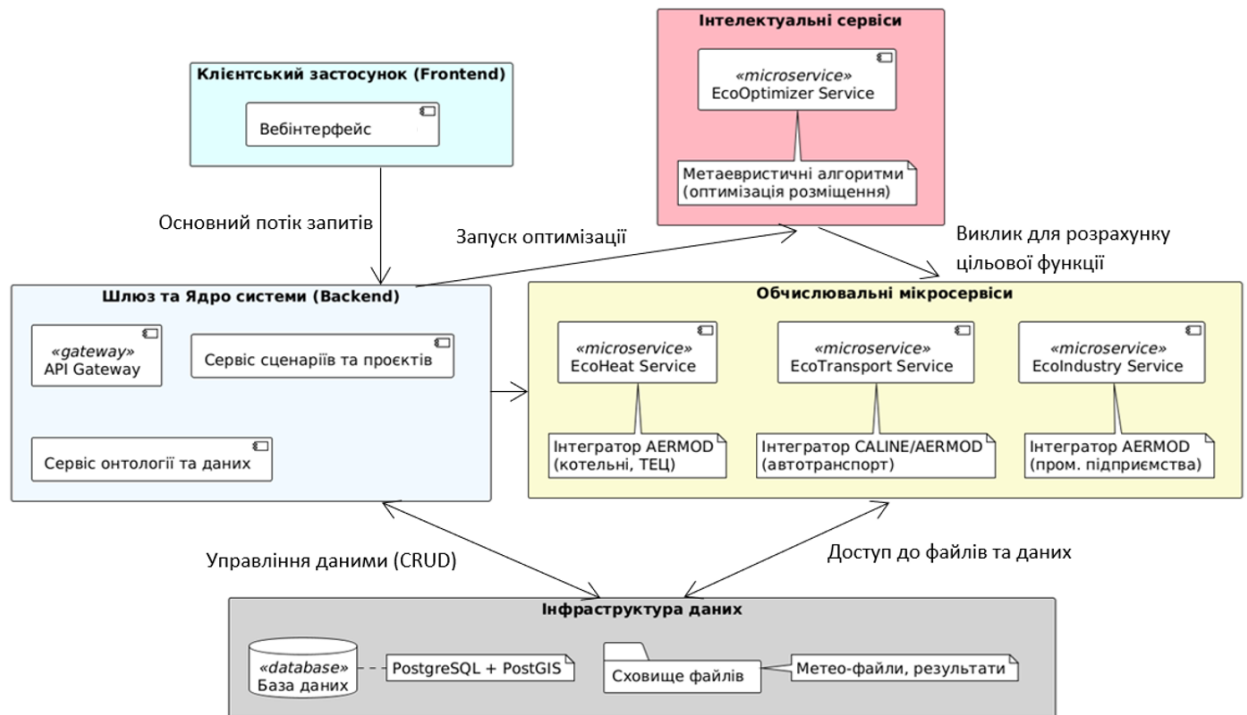


Рис. 4.1. Узагальнена архітектура системи

Міська екосистема є складною, гетерогенною системою, що складається з різних за своєю природою підсистем: теплоенергетики, транспорту, промисловості. Кожна з цих підсистем вимагає власного, специфічного математичного апарату для моделювання (AERMOD для стаціонарних джерел, CALINE для автотранспорту тощо). Вибір мікросервісної архітектури є прямим вирішенням цієї проблеми. Декомпозиція системи на незалежні обчислювальні мікросервіси (EcoHeat) дозволяє інкапсулювати складність, оскільки кожен сервіс є «чорною скринькою», що реалізує один конкретний тип моделювання, приховуючи деталі інтеграції з відповідним обчислювальним ядром. Систему можна легко розширювати, додаючи нові сервіси (наприклад, EcoTransport, EcoIndustry, EcoNoise) без будь-якого втручання в існуючі компоненти. Це є критично важливим для

довгострокового розвитку системи.

Проактивне управління базується на можливості оперативно відповідати на питання «що-якщо?». Це вимагає від системи здатності виконувати численні розрахунки за короткий час. Виділення обчислювальних сервісів в окремий пакет дозволяє забезпечити їхню масштабованість. У випадку високого навантаження, кількість екземплярів обчислювальних мікросервісів може бути динамічно збільшена, що гарантує збереження швидкодії системи та можливість проведення ітеративного моделювання.

Проактивне управління – це не лише прогноз, а й пошук найкращого рішення. Задача «знайти оптимальне місце для розміщення нового об'єкта» є нетривіальною оптимізаційною задачею. Виділення інтелектуалізованої логіки в окремий пакет сервісів (EcoOptimizer) є ключовим архітектурним рішенням. Це дозволяє розділити відповідальність, оскільки, обчислювальні сервіси відповідають на питання «яким буде вплив?», тоді як інтелектуалізований сервіс відповідає на питання «де вплив буде мінімальним?». Також, оптимізаційний сервіс може ітераційно викликати один або декілька обчислювальних сервісів для розрахунку цільової функції. Це дозволяє реалізовувати складні мультикритеріальні оптимізаційні задачі (наприклад, мінімізувати вплив і на якість повітря, і на рівень шуму одночасно).

Виділення інфраструктури даних в окремий архітектурний пакет, що включає як структуровану базу даних, так і файлове сховище, забезпечує «єдине джерело істини». Усі мікросервіси звертаються до цього централізованого сховища, що гарантує консистентність даних у всій системі. Використання онтологічної моделі даних замість простої реляційної схеми додатково забезпечує семантичну інтероперабельність – єдине розуміння термінів, що є фундаментом для майбутньої інтеграції з іншими міськими системами (транспортними, містобудівними) в рамках єдиної системи.

Таким чином, запропонована сервіс-орієнтована архітектура створює надійний та гнучкий фундамент для побудови повноцінної, інтегрованої

інтелектуалізованої системи для проактивного управління екосистемою міста.

Проектування інтелектуалізованої системи підтримки проактивного управління екосистемою міста потребує чіткого формального визначення як функціональних, так і нефункціональних вимог, що визначають поведінку, можливості, обмеження та якісні характеристики програмного комплексу. У цій роботі специфікація вимог ґрунтується на онтологічній моделі UESO, що забезпечує семантичну однозначність, а також на сервіс-орієнтованій та мікросервісній архітектурі системи, яка формує основу її масштабованості, розширюваності та відмовостійкості.

Функціональні вимоги до системи структуровано за модульним принципом, що забезпечує підтримку повного циклу сценарного моделювання, оцінювання та оптимізації. Архітектура функціональних вимог базується на декомпозиції системи на чотири ключові підсистеми: управління знаннями, сценарного аналізу, оптимізаційного синтезу та інтерактивної візуалізації.

Базовим компонентом є модуль управління знаннями, що реалізує концепцію «єдиного джерела істини» на основі онтології UESO. Він забезпечує виконання CRUD-операцій над екземплярами класів (PointSource, Receptor, MeteorologicalSet) із суворим дотриманням семантичних обмежень. Механізм логічної валідації гарантує цілісність даних: система унеможливорює збереження об'єктів із порушенням аксіоматики (наприклад, джерел без фізичних параметрів емісії) або геометрії, що не відповідає стандарту GeoSPARQL.

Цей блок відповідає за оркестрацію обчислювального процесу. Функціонал включає формування структури сценарію, агрегацію множин джерел та рецепторів, а також управління життєвим циклом обчислень через механізм скінченних автоматів. Ключовою функцією є автоматизована трансляція семантичних даних у синтаксис керуючого файлу AERMOD.INP, запуск зовнішнього ядра AERMOD та зворотний парсинг результатів у геопросторів шари, що забезпечує повну автоматизацію розрахункового

ланцюга.

Модуль реалізує розв'язання задач пошуку оптимальних конфігурацій джерел викидів. Особливістю є адаптивна стратегія вибору алгоритму: для задач низької розмірності застосовується детермінований повний перебір, тоді як для багатовимірних задач автоматично задіюються метаєвристичні підходи. Оптимізатор функціонує в режимі ітеративного зворотного зв'язку з ядром AERMOD, використовуючи його для обчислення значень цільової функції.

Підсистема візуалізації та підтримки рішень забезпечує інтерпретацію чисельних результатів засобами інтерактивної картографії. Функціонал охоплює генерацію теплових карт, побудову ізоліній концентрацій та візуалізацію Парето-оптимальних рішень. Реалізовано механізм диференціального аналізу (відображення різницевої карт), що дозволяє кількісно оцінювати екологічний ефект від впровадження різних управлінських сценаріїв.

Реалізація визначених вимог дозволяє сформувати замкнений технологічний цикл підтримки прийняття рішень, що інтегрує семантичне моделювання, високопродуктивні обчислення та геопросторову аналітику в єдине інформаційне середовище.

Нефункціональні вимоги визначають експлуатаційні характеристики системи, що є критичними у контексті застосування Eco-City як інструменту стратегічного планування екологічної політики міста. До таких вимог відносять продуктивність, масштабованість, надійність, безпеку, юзабіліті, супроводжуваність і відповідність стандартам.

Система повинна забезпечувати високу швидкість відгуку на операції користувача, що є суттєвим для роботи у режимі інтерактивної аналітики. Оцінювальні сценарії, що включають річні метеорологічні дані та сотні джерел, мають виконуватися впродовж хвилини, що досягається завдяки оптимізованій контейнеризації AERMOD та розподіленому виконанню на кластері.

Архітектура повинна підтримувати горизонтальне масштабування

мікросервісів без зміни бізнес-логіки. Обчислювальні компоненти, такі як EcoHeat та EcoOptimizer, запускаються у кількох екземплярах, що дозволяє пропорційно збільшувати пропускну здатність системи. Окремою вимогою є підтримка конкурентного доступу: система повинна залишатися стабільною за наявності десятків одночасних користувачів, що активно взаємодіють з кадастром і виконують сценарні розрахунки.

Оскільки система використовується для моделювання екологічних наслідків управлінських рішень, вона повинна відповідати високим стандартам доступності. Всі операції, пов'язані з онтологією, здійснюються транзакційно, що гарантує цілісність даних навіть у разі технічних збоїв.

Вимоги безпеки охоплюють автентифікацію на основі токенів (JWT), рольову авторизацію (RBAC) та захищену комунікацію між усіма сервісами (HTTPS/TLS). Доступ до бізнес-функцій чітко розподілений між ролями «Менеджер кадастру», «Аналітик» та «Адміністратор», що виключає неконтрольовані зміни в онтології та запуск обчислень неавторизованими користувачами. Діаграму варіантів використання представлено на рисунку 4.2.

Інтерфейс повинен забезпечувати інструментальну простоту роботи навіть для користувачів, які не мають програмістської підготовки. Архітектура UI базується на принципі когнітивного мінімалізму: складні операції з AERMOD, BPIPPRM чи SPARQL приховані за зрозумілими бізнес-поняттями. Мікросервіси системи повинні бути слабо зв'язаними, що дає змогу оновлювати або замінювати окремі компоненти (наприклад, перехід на нову версію AERMOD) без впливу на інші підсистеми. Усі сервіси здійснюють структуроване логування у централізоване сховище, яке полегшує моніторинг стану та налагодження роботи системи.

Надзвичайно важливим є використання відкритих стандартів обміну геопросторовими та семантичними даними. Онтологія UESO відповідає вимогам SWEET та GeoSPARQL, а геопросторові дані – стандартам OGC. Це забезпечує готовність системи до інтеграції з іншими міськими цифровими платформами – транспортними, енергетичними і містобудівними.

Сформульовані функціональні та нефункціональні вимоги визначають логічну, технічну і семантичну основу для побудови інтелектуалізованої системи Eсо-City. Вони забезпечують повноту покриття всіх аспектів циклу проактивного управління міською екосистемою – від збору й структурування даних до моделювання, оптимізації та аналітичної візуалізації.

Основними користувачами системи є (діаграма варіантів використання, яка представлена на рис. 4.2): аналітик, менеджер кадастру та адміністратор.

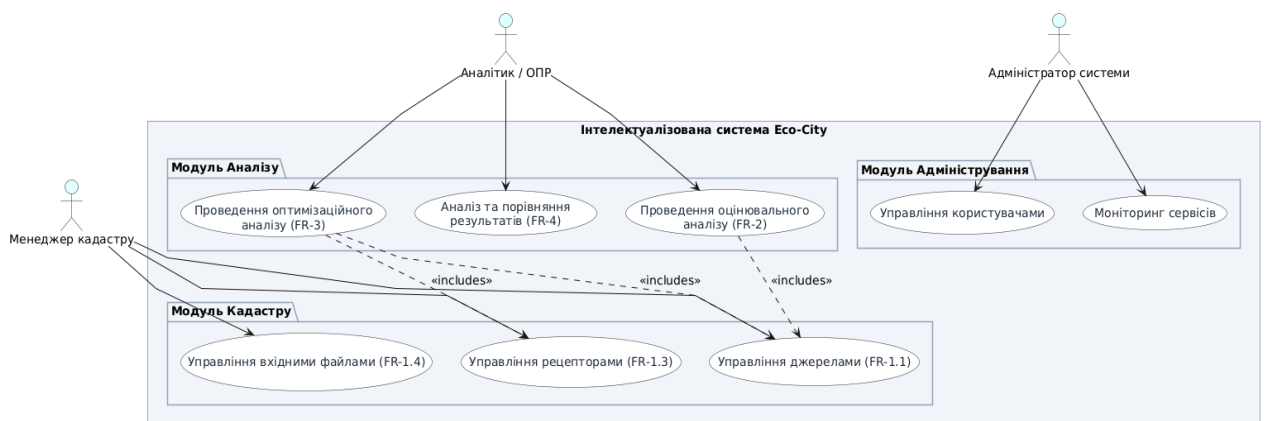


Рис. 4.2. Діаграма варіантів використання системи

Для наочної демонстрації комплексного процесу моделювання, що поєднує етапи інтерактивної взаємодії з користувачем, попередньої обробки даних та автоматизованих обчислень, розроблено діаграму діяльності (рис. 4.3), що візуалізує загальний потік даних запропонованої системи. Діаграма структурно розділена на чотири вертикальні доріжки, кожна з яких представляє окремого актора або логічний компонент системи: «Користувач», Система «Есо-City» (клієнтський та серверний застосунки), «Препроцесори» (зовнішні утиліти AERMET та BPIPPRM) та «Обчислювальне ядро» (AERMOD). Такий поділ чітко розмежовує зони відповідальності та ілюструє взаємодію між компонентами. Процес моделювання складається з наступних послідовних етапів:

1. Етап визначення сценарію. Процес ініціюється користувачем через веб-інтерфейс системи. Користувач формує екземпляр сценарію *s*, визначаючи

ключові параметри експерименту: набір джерел для моделювання I_s (вибираючи з кадастру та/або додаючи нові), набір точок-рецепторів I_R та посилання на попередньо підготовлений набір метеорологічних даних I_M .

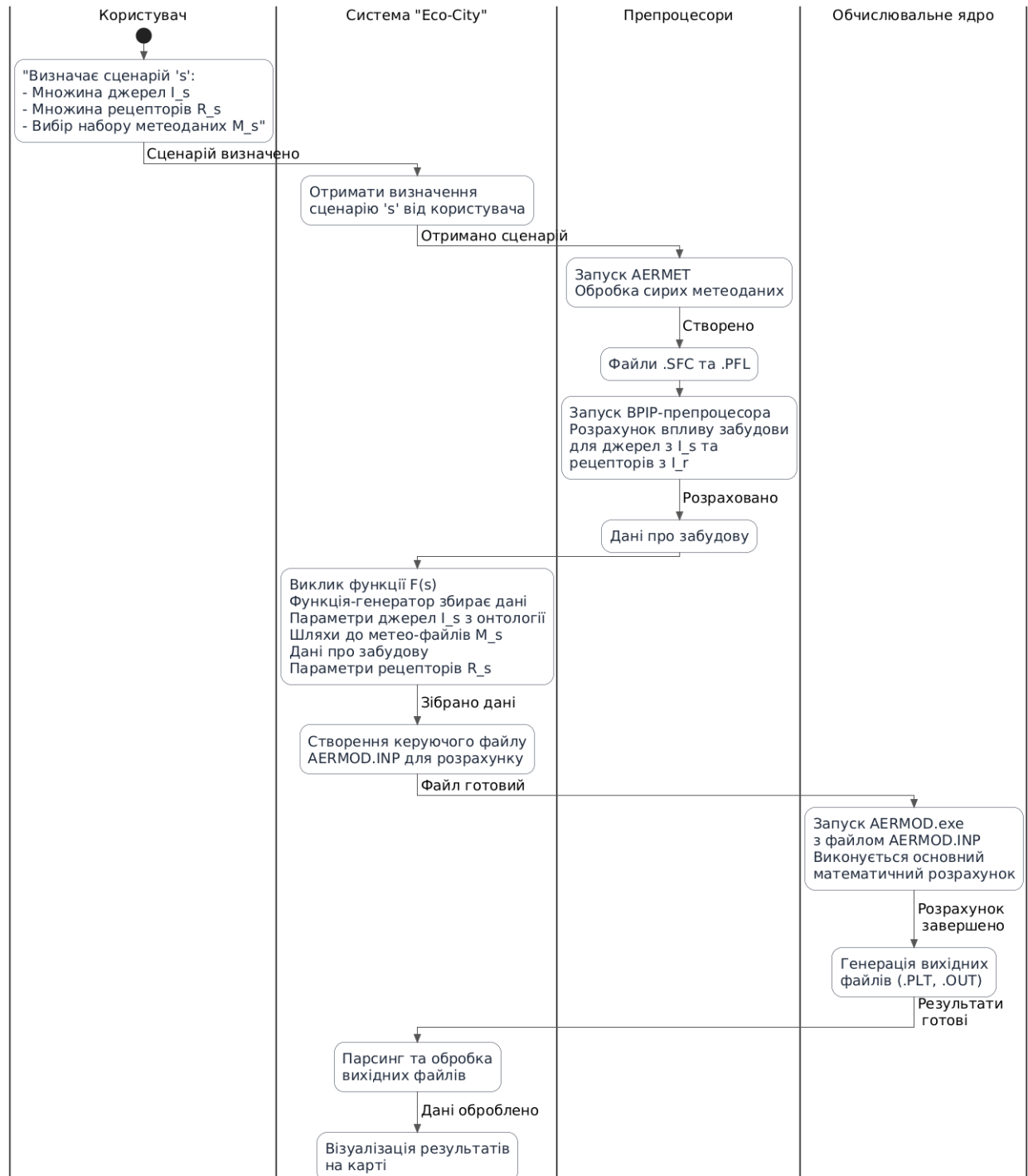


Рис. 4.3. Модель процесів і потоків даних системи

2. Етап попередньої обробки вхідних даних. Цей етап є разовим або

періодичним і виконується незалежно від основного циклу моделювання. Препроцесор AERMET приймає на вхід сирі метеорологічні дані (погодинні спостереження та дані аерологічного зондування) та генерує на їх основі два бінарні файли (.SFC та .PFL), що містять повний набір параметрів, необхідних для розрахунків розсіювання. Ці файли зберігаються у сховищі даних системи.

Після отримання визначення сценарію система «Eco-City» автоматично ініціює виклик препроцесора BPIPPRM. Цей модуль отримує з онтології координати та габарити будівель, а також координати джерел $i \in I_s$, та розраховує для кожного з них залежні параметри, що характеризують вплив аеродинамічної тіні. Результати зберігаються як тимчасовий артефакт, асоційований зі сценарієм s .

3. Етап генерації керуючого файлу. На цьому ключовому етапі система викликає функцію-генератор $F(s)$. Ця функція виступає центральним інтеграційним вузлом: вона збирає всі підготовлені артефакти – параметри джерел та рецепторів з онтології (ABox), шляхи до метеорологічних файлів, результати розрахунку BPIPPRM та параметри конфігурації – і на їх основі синтезує єдиний, синтаксично та логічно коректний керуючий файл AERMOD.INP.

4. Етап обчислення та постобробки. Згенерований файл AERMOD.INP передається на виконання обчислювальному ядру AERMOD, яке проводить математичний розрахунок розсіювання. Після завершення розрахунку система виконує парсинг вихідних файлів (зокрема, .PLT), трансформує отримані дані у формат GeoJSON та візуалізує результуючі поля концентрацій на інтерактивній карті для кінцевого аналізу користувачем.

Таким чином, представлена модель робочого процесу демонструє перехід від фрагментованих, ручних операцій до цілісного, автоматизованого та відтворюваного конвеєра обробки даних. Запропонована архітектура дозволяє інкапсулювати складність препроцесингу та обчислень, надаючи користувачеві простий та потужний інструмент для інтерактивного прогностичного моделювання.

На рис. 4.4 представлена діаграма послідовності побудови й реалізації сценарію. Ця діаграма відображає етапи створення сценаріїв, інтеграцію даних та застосування моделей для прогнозування та аналізу.

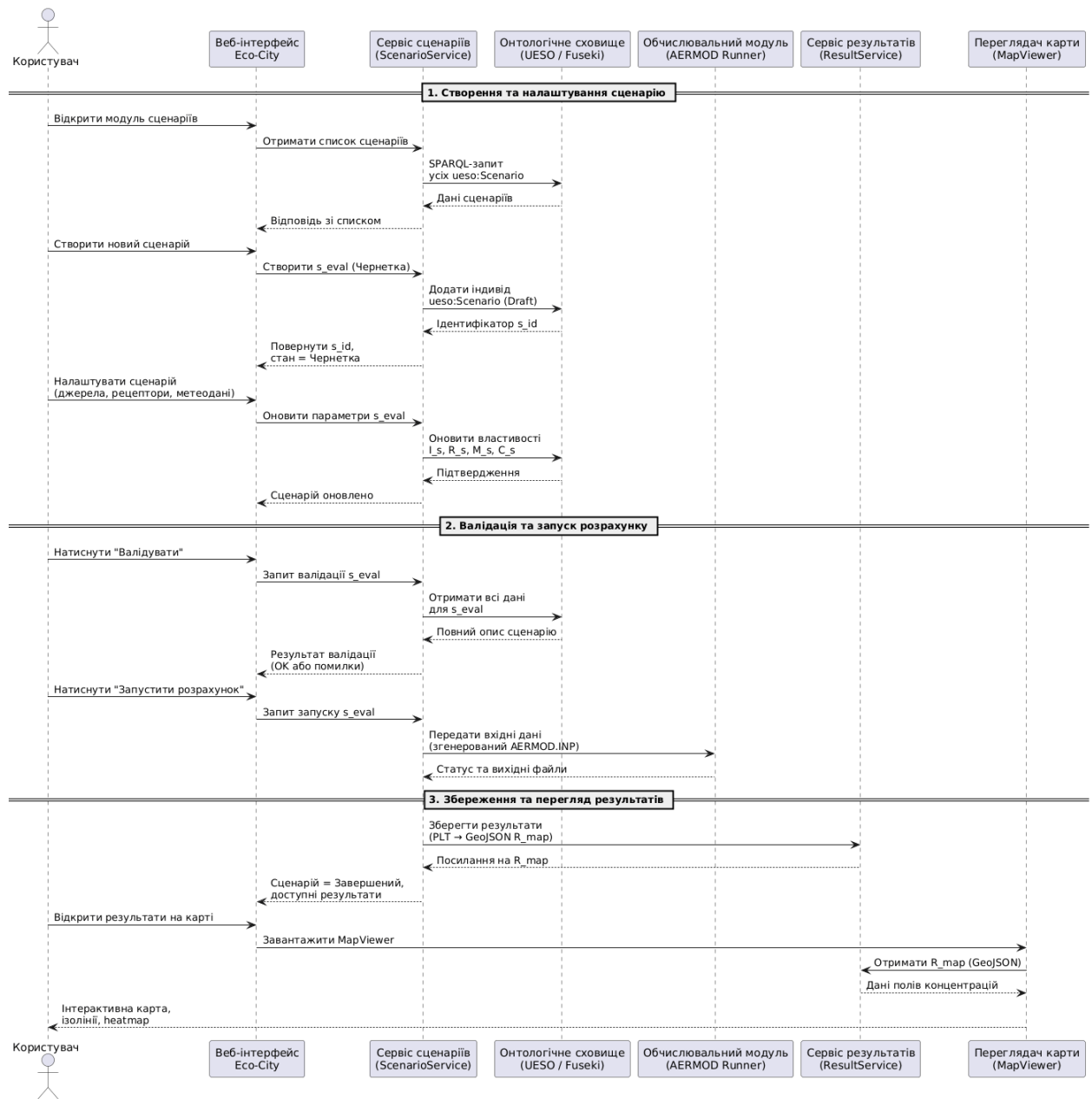


Рис. 4.4. Діаграма послідовності побудови й реалізації сценарію

Послідовність дій відображає взаємодію користувача з веб-інтерфейсом, сервісом сценаріїв (ScenarioService), онтологічною підсистемою (OntologyService), модулем логічної валідації (ValidationEngine), препроцесорами AERMET/BPIPPRM, функцією трансляції $F(s)$ для побудови

керуючого файлу aermod.inp, обчислювальним ядром AERMOD та сервісом збереження результатів (ResultService).

Важливим є те, що кожен етап – налаштування сценарію, валідація, запуск розрахунку, генерація вихідних карт концентрацій – формалізовано як окремий крок послідовності, при цьому система завжди працює з онтологічним поданням даних (ABox), а не з неструктурованими файлами. Це забезпечує відтворюваність, автоматизовану перевірку повноти даних та можливість повторного запуску сценаріїв із мінімальними витратами на підготовку вхідних даних.

На діаграмі класів предметної області системи Eco-City показано основні сутності, що відображають структуру онтології UESO та сценарно-орієнтовану модель моделювання (рис. 4.5).

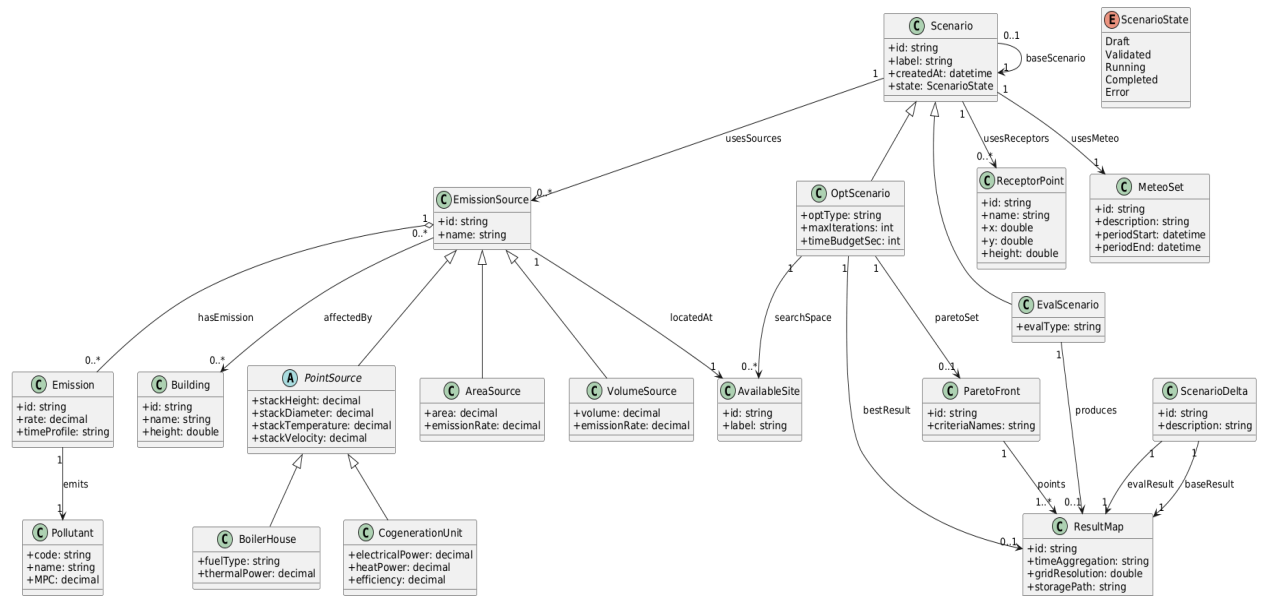


Рис. 4.5. Діаграма основних класів системи Eco-City

Базовим поняттям є клас EmissionSource, який має спеціалізації BoilerHouse, CogenerationUnit, AreaSource та VolumeSource, що відповідають гетерогенним типам джерел викидів у міському середовищі. Кожне джерело асоційоване з набором об'єктів класу Emission, які описують інтенсивність викидів конкретних поллютантів (Pollutant), їх часові профілі та інші

параметри. Просторовий аспект моделі представлено класами *ReceptorPoint*, *Building* та *AvailableSite*, які задають точки спостереження, об'єкти забудови та потенційні локації для розміщення нових джерел.

Класи *Scenario*, *EvalScenario* та *OptScenario* формалізують сценарний підхід до моделювання, де кожен сценарій пов'язаний із множиною джерел, рецепторів та метеорологічним набором (*MeteoSet*), а також має стан життєвого циклу. Результати моделювання представлені графом класів *ResultMap*, *ScenarioDelta* та *ParetoFront*, які підтримують як оцінювальний, так і оптимізаційний аналіз (у тому числі побудову Парето-фронтів). Така структуризація забезпечує безпосередню відповідність між онтологічною моделлю UESO, сценарними структурами та програмною реалізацією системи Eco-City.

Реалізація системи Eco-City ґрунтується на концептуальному поєднанні мікросервісної архітектури та семантичної моделі предметної області, що задається онтологією UESO (*Urban Emission Source Ontology*). Такий підхід дозволяє відокремити логіку моделювання процесів розсіювання забруднювачів від структур даних і способів їх зберігання, забезпечуючи незалежну еволюцію окремих модулів і, водночас, цілісність їхнього змістового представлення. Система будується навколо принципу слабкого зв'язування компонентів, коли кожен функціональний блок (робота зі сценаріями, управління джерелами та рецепторами, моделювання AERMOD, оброблення метеорологічних даних, оптимізація, візуалізація) є самостійною сервісною одиницею. Комунікація між сервісами виконується через стандартизовані протоколи REST/HTTP та структури даних JSON, RDF і GeoJSON, що спрощує масштабування та розподілене виконання обчислень.

Усі сервіси розгортаються в інфраструктурі Kubernetes, де вони отримують окремі контейнерні середовища виконання, незалежні політики масштабування, горизонтальні балансувальники навантаження та централізований контроль їхнього стану.

Важливою передумовою такої архітектури є визначення чіткої ролі

онтологічного шару, який виступає єдиним джерелом істини щодо структури даних, логічних зв'язків та критеріїв валідності предметної області. Завдяки цьому програмні модулі системи зосереджуються не на дублюванні логіки валідації чи трансформації даних, а на виконанні спеціалізованих функцій, використовуючи уніфіковану семантичну основу.

Семантичний рівень базується на онтології UESO, реалізованій у формальному форматі OWL2DL і розгорнутій у triple-store-сховищі Apache Jena Fuseki. Онтологія містить формальні визначення типів джерел викидів, рецепторів, просторових характеристик, параметрів моделей AERMOD, метаданих метеорологічних спостережень та сценаріїв моделювання. Таким чином, UESO виступає не просто сховищем структурованих фактів, а повноцінним механізмом контролю семантичної коректності даних у системі. Будь-який сервіс Eсо-City, який взаємодіє з даними, отримує їх уже у валідованому вигляді, що радикально знижує ризик логічних помилок у моделюванні.

Геопросторові дані відіграють ключову роль у моделюванні процесу розсіювання забруднювачів. Використання стандарту GeoSPARQL дозволяє зберігати геометрію джерел і рецепторів у форматі WKT, виконувати топологічні запити (перетини, включення, відстані) та автоматично інтегрувати їх із картографічними інструментами. У цьому контексті Eсо-City реалізує одночасно два підходи: геометрію для AERMOD зберігається у строгому семантичному форматі RDF/GeoSPARQL та візуалізація здійснюється через перетворення цих геометрій у GeoJSON.

Це забезпечує єдину логічну основу для всіх модулів системи. На рисунку 4.6. представлено діаграму розгортання системи.

На діаграмі розгортання відображено ключові компоненти програмно-апаратної інфраструктури системи Eсо-City, які у своїй сукупності формують архітектуру високої складності, орієнтовану на виконання ресурсоемних обчислювальних завдань, забезпечення логічної цілісності даних та підтримку масштабованого сценарного аналізу. У верхньому рівні інфраструктури

знаходиться клієнтський пристрій користувача, який взаємодіє з системою через веб-браузер. Саме браузер забезпечує доступ до багатосторінкового веб-застосунку, що включає інструменти створення сценаріїв, управління джерелами та рецепторами, перегляду результатів моделювання та роботи з онтологічною підсистемою.

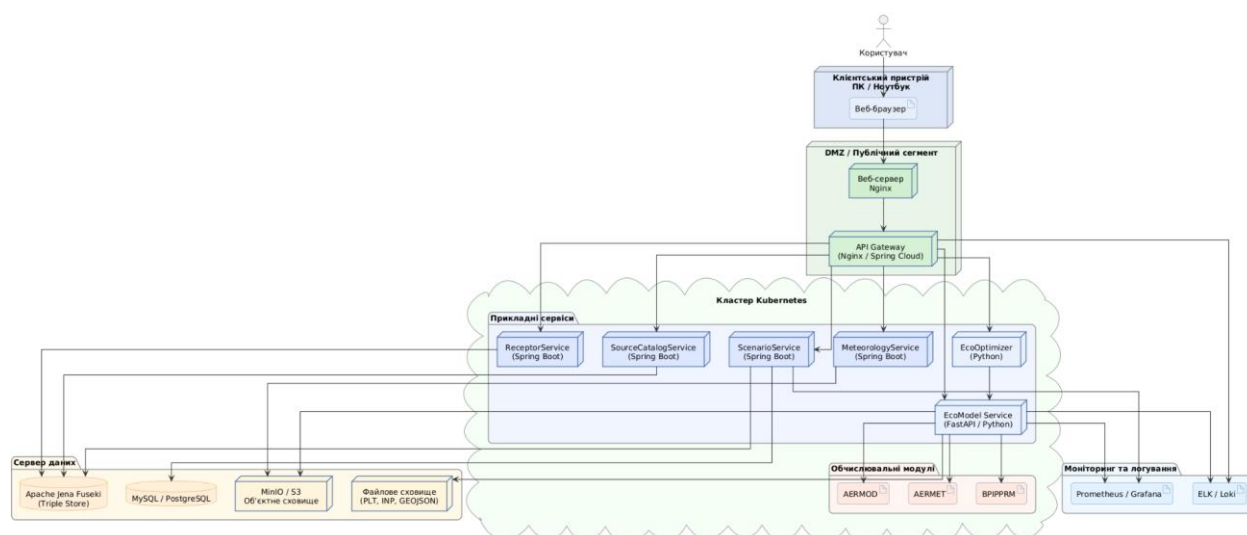


Рис. 4.6. Діаграма розгортання системи

Комунікація між клієнтом та внутрішніми компонентами системи здійснюється через публічний сегмент мережі (DMZ), де розгорнуто веб-сервер Nginx та API-шлюз, який виконує маршрутизацію запитів до відповідних мікросервісів. Наявність DMZ-домену є важливим елементом безпеки: він дозволяє ізолювати зовнішні запити від внутрішньої логіки, а також забезпечує кешування статичних ресурсів, обмеження швидкості запитів, захист від DDoS-атак та централізовану обробку HTTPS-трафіку.

Основна обчислювальна логіка розгорнута в кластері Kubernetes, що є ключовим елементом складності системи, оскільки забезпечує автоматичне створення, масштабування, відновлення та оновлення мікросервісів. Кожен сервіс (ScenarioService, SourceCatalogService, ReceptorService, EcoModel, EcoOptimizer та ін.) працює в окремому контейнері, ізольованому на рівні операційної системи. Така архітектура дозволяє системі ефективно

розподіляти навантаження, виконувати паралельні моделювальні експерименти та підтримувати одночасну роботу великої кількості сценаріїв – важливу властивість при реалізації проактивного управління міською екосистемою.

Особливої уваги потребують обчислювальні утиліти AERMOD, AERMET та BIPPRM. Вони не створювались як частина хмарних або контейнеризованих екосистем, тому їхня інтеграція в Kubernetes вимагає спеціального обгортання (wrapper-сервісів) і механізмів контролю ресурсів. Зокрема, AERMOD є високонавантаженим обчислювальним модулем, чутливим до переривань та зміни середовища виконання. Для забезпечення стабільності функціонування його запуск реалізується у відокремлених pod-екземплярах, які отримують гарантовані ресурси (CPU/Memory Requests & Limits), а результати його роботи зберігаються у стійких томах (Persistent Volume Claims). Таке рішення дозволяє уникати повторних обчислень та втрати даних при перезапуску сервісів, що є типовою проблемою хмарних розгортань.

Сервер даних представлений трирівневою підсистемою зберігання, яка реалізує структуру різних типів навантажень:

- онтологічне triple-store (Apache Jena Fuseki) відповідає за семантичне представлення даних та виконання SPARQL-запитів;
- реляційна база даних (MySQL) забезпечує транзакційну цілісність службових і системних даних;
- об'єктне файлове сховище MinIO використовується для зберігання великих результатів моделювання, файлів AERMOD, метеоданих та GeoJSON-артефактів.

Така гетерогенність зберігання даних визначає складність програмної реалізації: кожен тип даних має власні вимоги щодо форматів обміну, протоколів взаємодії, транзакційних властивостей, а також окремі механізми масштабування. Наприклад, SPARQL-запити до Fuseki мають інший профіль навантаження, ніж SQL-запити до реляційної бази, що вимагає різних

стратегій кешування та балансування.

Для забезпечення відмовостійкості та нагляду за працездатністю системи у структурі передбачено окремий сервер моніторингу та логування, який збирає метрики з усіх мікросервісів, кластерних вузлів та інфраструктурних компонентів. Використання Prometheus та Grafana дозволяє відслідковувати продуктивність моделі, завантаженість CPU і пам'яті, затримки запитів, час виконання AERMOD та інші параметри в реальному часі. Журнали подій зберігаються в централізованій системі логування (ELK або Loki), що забезпечує їхню доступність для аудиту, аналізу помилок і діагностики взаємодії між сервісами. Наявність такого рівня моніторингу є необхідною умовою функціонування складної розподіленої системи, яка має підтримувати високу надійність і бути здатною до автономного відновлення.

Узагальнюючи, представлена конфігурація системи Eco-City не лише відображає окремі функціональні компоненти, але й демонструє високий рівень технічної організованості, властивий сучасним інтелектуальним платформам. Комбінація мікросервісної архітектури, контейнеризації, семантичних технологій та спеціалізованих обчислювальних модулів створює комплексну інфраструктуру, здатну обробляти великий обсяг гетерогенних даних, виконувати ресурсомісткі оптимізаційні та модельні задачі й залишатися відкритою для подальшої інтеграції з транспортними, містобудівними, енергетичними та іншими міськими цифровими платформами. Таке рішення забезпечує не лише технічну ефективність, але й концептуальну гнучкість, необхідну для реалізації повномасштабної системи проактивного управління міською екосистемою.

4.2 Програмна реалізація підсистем управління знаннями та даними.

4.2.1. Проєктування та реалізації бази даних

Для проєктування та реалізації бази даних було обрано СУБД

PostgreSQL у поєднанні з просторовим розширенням PostGIS. Таке поєднання забезпечує надійну, відкриту та високопродуктивну геопросторову систему баз даних, здатну інтегрувати екологічні, просторові та часові дані в єдиному аналітичному середовищі. Однією з головних переваг СУБД PostgreSQL є її комплексна можливість просторового аналізу, яка дозволяє зберігати та обробляти як векторні, так і растрові дані безпосередньо. Джерела викидів, такі як котельні та когенераційні установки, можуть бути представлені як просторові об'єкти (точки або полігони), тоді як результати моделі розсіювання можуть зберігатися як растрові набори даних, що представляють поля концентрації забруднюючих речовин. Ця подвійна можливість дозволяє проводити прямий геопросторовий аналіз в рамках SQL-запитів, таких як розрахунок концентрації забруднюючих речовин в адміністративних межах, визначення постраждалих районів населення або обчислення індексів сукупного впливу, без потреби у зовнішніх ГІС-інструментах. Можливість поєднувати векторний та растровий аналіз в одному запиті до бази даних значно спрощує робочий процес та покращує відтворюваність.

Крім того, використання PostgreSQL + PostGIS забезпечує високу гнучкість моделювання та масштабованість системи, що є критично важливим для задач екологічного моніторингу та прогнозування. Крім того СУБД дозволяє ефективно керувати великими обсягами даних, від детальних погодинних метеорологічних вимірювань до багат шарових наборів результатів моделювання. У поєднанні з можливістю паралельної обробки та використанням розширень для оптимізації продуктивності PostgreSQL стає ефективною платформою не лише для зберігання даних, але й для виконання складних екологічних розрахунків, інтеграції з веб-сервісами та побудови повноцінних систем підтримки прийняття рішень.

Дана схема бази даних розроблена для підтримки всіх функціональних і даних вимог щодо моделювання та аналізу сукупного впливу котелень і когенераційних установок на якість повітря в місті. На рис. 4.7. представлено ER-діаграму бази даних.

Таблиці Districts, представляє адміністративні райони міста у вигляді багатокутників (geometry(MultiPolygon, 4326)). Кожен район має унікальний ідентифікатор і назву, що дозволяє групувати або фільтрувати об'єкти та результати моделювання за територією.

Таблиця Facilities є однією з ключових складових структури бази даних, оскільки містить повний кадастр стаціонарних джерел забруднення атмосферного повітря на території міста. Таблиця підтримує дві групи атрибутів: описові (name, address, facility_type) та моделювальні, пов'язані з формуванням вхідних параметрів для моделей дисперсії, таких як AERMOD. Додаткові службові поля (created_at, updated_at) забезпечують відстежуваність змін та контроль версій даних.

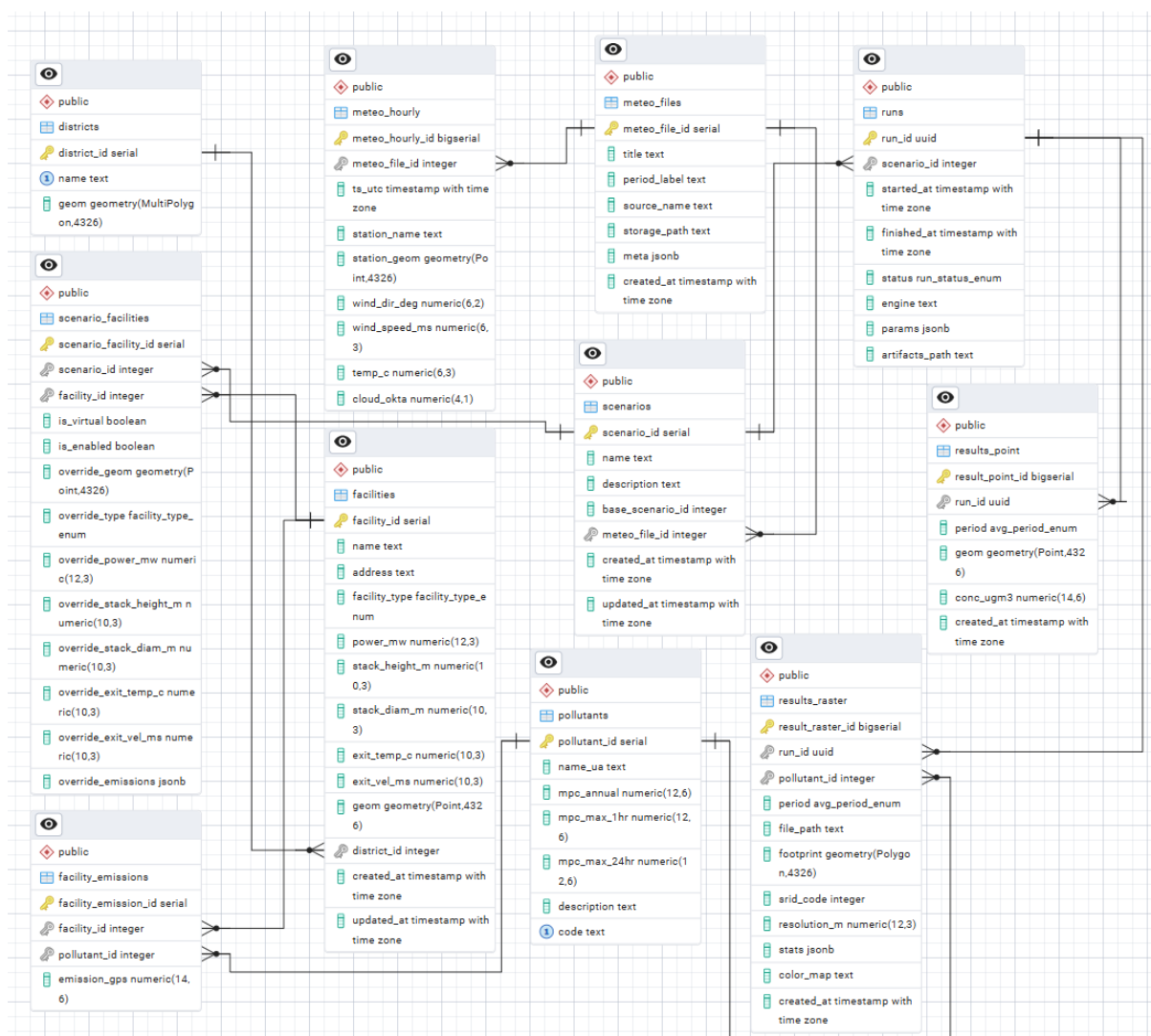


Рис. 4.6. ER-діаграма бази даних

Таблиця Pollutants виконує функцію словника доменних сутностей, що описують забруднювальні речовини, які враховуються під час моделювання якості атмосферного повітря. Крім інформаційних атрибутів, таблиця містить значення гранично допустимих концентрацій (ГДК) у середньорічному та максимальному одноразовому режимах, що забезпечує автоматичну оцінку відповідності результатів моделювання нормативним вимогам.

Таблиця Scenarios реалізує механізм сценарного аналізу. Кожен сценарій репрезентує окрему конфігурацію умов моделювання, включаючи вибір метеорологічного періоду, набір активних джерел викидів, наявність або відсутність віртуальних об'єктів, а також можливі зміни технічних параметрів існуючих установок.

Таблиця Runs фіксує кожен запуск моделювання дисперсії. Вона містить унікальний ідентифікатор (run_id), посилання на сценарій (scenario_id), часові мітки початку і завершення, статус (status), назву обчислювального модуля (engine), параметри запуску (params) і шлях до артефактів (artifacts_path). Це відповідає вимозі щодо можливості виконання розрахунків на основі вбудованої математичної моделі (наприклад, AERMOD) і зберігання інформації про кожен розрахунок.

Метеорологічні дані представлені двома пов'язаними таблицями: Meteo_files і Meteo_hourly. Таблиця meteo_files зберігає метадані про завантажені файли з метеоданими такі як назву, період спостережень, джерело даних, шлях до файлу (storage_path) і додаткову інформацію (meta). Таблиця meteo_hourly зберігає нормалізовані погодинні спостереження: час (ts_utc), назву станції, геометрію розташування (station_geom), напрямок і швидкість вітру (wind_dir_deg, wind_speed_ms), температуру (temp_c) і хмарність (cloud_okta). Ці таблиці реалізують вимогу завантаження та використання метеорологічних даних за певний період у процесах дисперсійного моделювання.

Таблиця Scenario_facilities дозволяє задавати параметри об'єктів у межах конкретного сценарію. Такий підхід реалізує вимоги щодо створення

«віртуальних» джерел та моделювання їхнього впливу. Результати моделювання зберігаються у двох таблицях: Results_point і Results_raster. Таблиця Results_point містить точкові результати розрахунків концентрацій для певних речовин і періодів осереднення (period), прив'язані до моделювання (run_id) та забруднювача (pollutant_id). Вона використовується для функції «Map Inspector», що дозволяє отримати інформацію про концентрацію в будь-якій точці карти, а також для аналізу перевищень МРС. Таблиця Results_raster зберігає метадані та шляхи до растрових файлів із полями концентрацій (file_path, footprint), що дозволяє візуалізувати результати у вигляді кольорових полів або ізоліній на карті.

Загалом ця схема формує цілісну структуру даних для системи підтримки прийняття рішень у сфері екологічної безпеки. Вона інтегрує просторові дані, інвентаризацію викидів, метеорологію, сценарії, розрахунки та результати в єдину реляційну модель. Зв'язки між таблицями забезпечують цілісність даних, а геометричні поля ефективний просторовий аналіз і візуалізацію. Схема дозволяє легко імпортувати початкові дані з файлів CSV або Excel, працювати з зовнішніми растровими результатами й експортувати карти або звіти у форматах PNG, PDF чи DOCX, повністю задовольняючи вимоги до системи моделювання впливу джерел на якість повітря в міських умовах.

Також у ході роботи над дослідженням було здійснено низку удосконалень схеми бази даних, спрямованих на підвищення продуктивності, масштабованості та надійності програмного комплексу для моделювання впливу джерел викидів на якість атмосферного повітря. Насамперед було впроваджено повну просторову індексацію геометричних полів у таблицях Districts, Facilities, Meteo_hourly, Scenario_facilities, Results_point та Results_raster. Додані GiST-індекси забезпечили істотне прискорення картографічних операцій, таких як просторові запити, визначення потрапляння точок у полігони та робота з результатами моделювання у вигляді точкових і растрових шарів. Це дозволило підвищити швидкодію

інтерактивної карти та оптимізувати обробку великих геопросторових масивів даних.

Паралельно з цим було оптимізовано часті запити, що виконуються під час аналізу сценаріїв та візуалізації результатів. Для цього були створені складені індекси на поля (run_id, pollutant_id, period) у таблицях Results_point та Results_raster, а також індекси на Facilities(district_id), Facilities(facility_type) та Meteo_hourly(meteo_file_id, ts_utc). Додавання цих індексів суттєво скоротило час вибірки даних під час завантаження результатів моделювання, формування тематичних карт, відображення впливу окремих забруднювачів та порівняння варіантів сценаріїв. Також було посилено цілісність довідника забруднювачів шляхом запровадження обов'язкового унікального коду (pollutants.code) та встановлення повноцінних зовнішніх ключів між таблицями Results_raster та Pollutants. Завдяки цьому вдалося усунути потенційні неузгодженості між результатами моделювання та характеристиками речовин, що використовуються при розрахунках перевищень гранично допустимих концентрацій.

Для обробки великих обсягів результатів було проаналізовано можливість секціонування таблиць Results_point та Meteo_hourly, що підвищує ефективність виконання запитів і спрощує зберігання результатів. Сукупність застосованих оптимізацій дозволила сформувати гнучку та стійку до масштабування інфраструктуру даних для екологічного моделювання та підтримки прийняття рішень.

4.2.2. Підсистема для семантичного моделювання гетерогенних джерел шкідливих викидів

В умовах, коли результати розрахунків AERMOD та інших моделей критично залежать від коректності опису джерел викидів, рецепторів і сценаріїв, стає важливо не лише «мати дані», а й мати формалізовані знання про них. Саме тому у запропонованій системі впроваджено онтологію UrbanEmissionSourceOntology (UESO), яка задає строгий формальний опис

предметної області, підтримує механізми логічного виведення й слугує семантичним «каркасом» цифрового кадастру джерел викидів.

Сторінка «Перегляд класів та екземплярів онтології UESO» (рис. 4.7.) виконує роль інтерфейсного «вікна» у онтологічну модель, роблячи її доступною для користувача-експерта. Інтерфейс поєднує TBox-рівень (ієрархію класів) та ABox-рівень (екземпляри й факти), дозволяючи у єдиному робочому просторі оглядати структуру онтології та конкретні індивіди, що представляють реальні об'єкти міської екосистеми. Ліва частина сторінки реалізує навігацію по класах UESO: від узагальненого EmissionSource до спеціалізованих PointSource, Boiler, CogenerationUnit, AreaSource, LineSource, а також класів Receptor, Scenario, EmissionParameter.

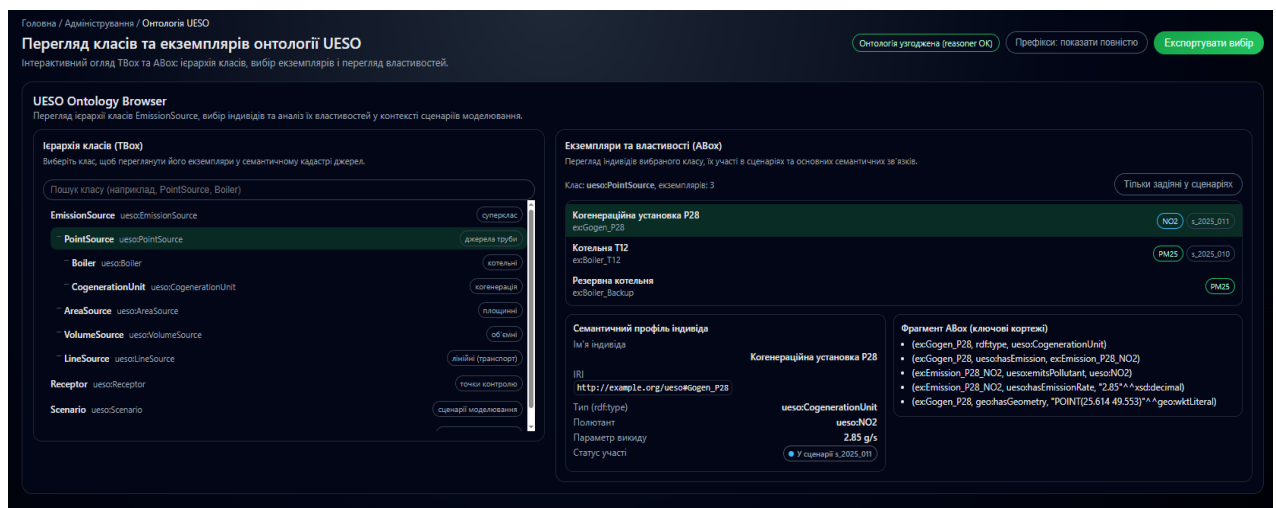


Рис. 4.7. Перегляд класів та екземплярів онтології

Таким чином, користувач отримує наочне уявлення про те, як формальні конструкції, описані в розділі онтологічної формалізації, реалізуються у конкретній ієрархії понять та онтологічному графі (рис. 4.8).

Права частина сторінки фокусується на екземплярах обраного класу та їх властивостях. Після вибору класу інтерфейс відображає список відповідних індивідів з короткими анотаціями: людино-орієнтованою назвою об'єкта, компактним ідентифікатором, позначками щодо домінуючого полютанта (наприклад, NO₂) і участі у конкретних сценаріях моделювання. Користувач

може моментально переключатися між котельнями, когенераційними установками, лінійними чи площинними джерелами, фільтруючи, за потреби, лише ті, що задіяні у сценаріях. Така можливість дозволяє швидко відповісти на змістовні запитання на кшталт: «які саме джерела класу PointSource включені до сценарію?» або «які рецептори використовуються для оцінки впливу нової когенераційної установки?».

Окремий блок сторінки присвячений деталізованому «семантичному профілю» вибраного індивіда. У компактній формі відображаються його IRI, тип (rdf:type), основні параметри викиду та статус участі у сценаріях, а поруч – фрагмент ABox у вигляді списку ключових кортежів.

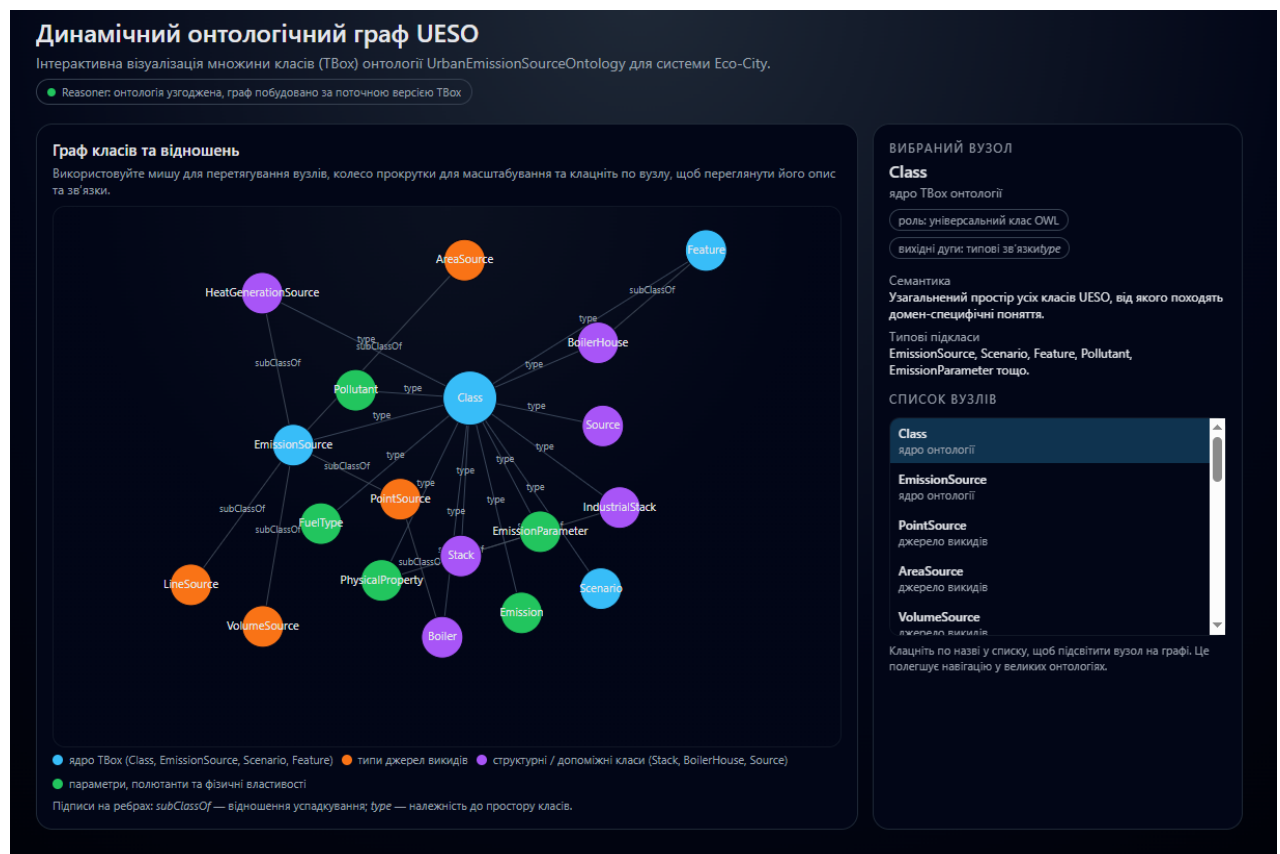


Рис. 4.8. Перегляд онтологічного графа

Це фактично дає користувачу безпосередній погляд на ті самі логічні твердження, які використовуються функціями трансляції для формування вхідних файлів AERMOD. У результаті забезпечується прозорий зв'язок між

високорівневою онтологічною моделлю, конкретним об'єктом міської інфраструктури та синтаксичною репрезентацією його параметрів у керуючому файлі моделі розсіювання (рис. 4.9).

З позицій проактивного управління міською екосистемою та відтворюваності обчислювальних експериментів, така сторінка виконує одразу кілька важливих функцій. Вона знижує когнітивне навантаження на експерта, оскільки замість роботи з розрізненими таблицями та текстовими файлами надає цілісний, семантично узгоджений огляд даних. Вона слугує інструментом валідації: невідповідності у класах, відсутні властивості, некоректно задані поллютанти чи геометрії стають помітними на рівні онтологічної структури ще до запуску моделювання. Нарешті, вона реалізує вимогу трасованості, дозволяючи прослідкувати шлях від сценарію моделювання до конкретних джерел і рецепторів, а далі – до відповідних тверджень, які лягають в основу автоматизованої генерації вхідних даних. Таким чином, інтерфейс перегляду класів та екземплярів онтології UESO органічно інтегрує формальний апарат онтологічного моделювання в щоденну практику роботи користувача системи Есо-City, роблячи онтологію не абстрактним теоретичним артефактом, а реально діючим інструментом проактивного екологічного управління.

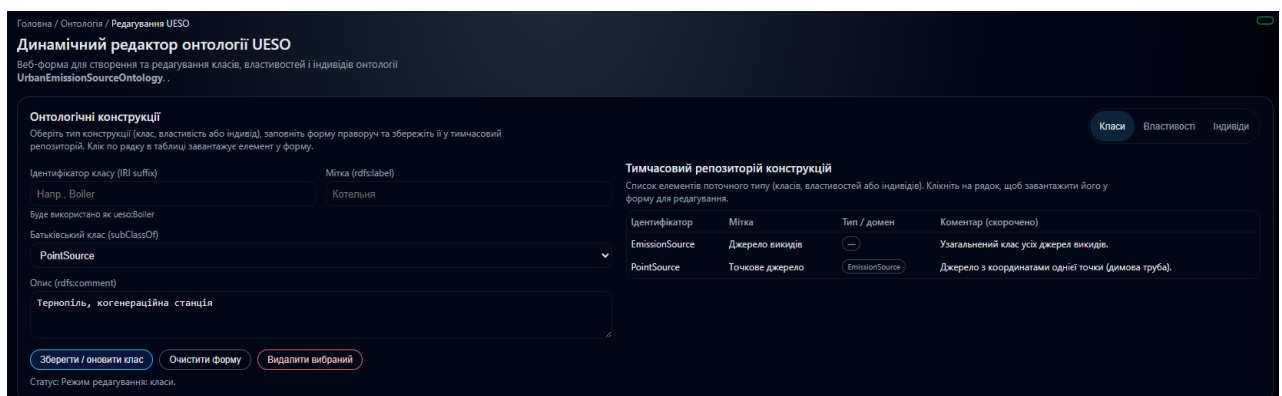


Рис. 4.9. Онтологічний редактор

Наступним елементом логічного ланцюга є сторінка геопросторової інтерпретації результатів онтологічних запитів (рис. 4.10). Замість традиційного табличного подання, інформація відображається на інтерактивній карті, що формує просторовий контекст для аналізу. Такий підхід особливо важливий для екологічних задач, оскільки позиційні характеристики джерел, рецепторів і будівель є визначальними для моделювання процесів розсіювання забруднювачів.

Результати SPARQL-запитів автоматично трансформуються у формат GeoJSON і візуалізуються у вигляді маркерів, ізоліній або зон впливу. Інтерфейс підтримує фільтрування, панорамування, масштабування та відображення додаткових атрибутів при виборі об'єкта (рис. 4.11).

Карта результатів SPARQL-запитів

Маркери на карті показують результат виконання відбраного запиту: точкові джерела, рецептори, джерела поблизу рецептора тощо.

Легенда: ● точкові джерела NO₂ (Запит 1) ● джерела сценарію (Запит 2) ● буфер рецептора / близькі джерела (Запит 4)

Керування візуалізацією

Виберіть набір результатів, що відобразяться на карті. Дані відповідають демонстраційним вибіркам з правої частини сторінки.

- Показати результати запиту 1 (PointSource NO₂)
- Показати результати запиту 2 (джерела сценарію)
- Показати результати запиту 3 (агрегація)
- Показати результати запиту 4 (радіус 500 м)

1. Точкові джерела NO₂ з геометрією

Вибір усіх індивідів *ueso:PointSource* з полем *NO₂* інтенсивністю викиду та геометрією WKT.

Copy

```

PREFIX ueso: <http://example.org/ueso#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>

SELECT ?source ?label ?rate ?geom
WHERE {
  ?source rdf:type ueso:PointSource ;
    ueso:hasLabel ?label ;
    ueso:hasEmission ?emission ;
    geo:hasGeometry ?g .

  ?emission ueso:emitsPollutant ueso:NO2 ;
    ueso:hasEmissionRate ?rate .

```

2. Джерела, включені до сценарію

Сценарій ex:Scenario_2025 та пов'язані з ним джерела, з обмеженням на підкласи *EmissionSource*.

Copy

```

PREFIX ueso: <http://example.org/ueso#>
PREFIX ex: <http://example.org/eco-city/>

SELECT ?source ?label ?type
WHERE {
  ex:Scenario_2025 ueso:includesSource ?source .

  ?source ueso:hasLabel ?label ;
    rdf:type ?type .

  ?type rdfs:subClassOf* ueso:EmissionSource .
}

ORDER BY ?type ?label

```

Рис. 4.10. Відображення результатів онтологічних запитів на карті

З погляду когнітивного дизайну, ця сторінка є ключовим інструментом семантичної інтеграції: вона поєднує логічну природу онтології з просторовою природою міських процесів, забезпечуючи користувачу миттєву інтуїтивну перевірку результатів запиту. Просторове подання дозволяє виявляти закономірності, які неможливо помітити в текстових даних, та забезпечує більш обґрунтовану підготовку сценаріїв моделювання.

Завершальним компонентом є інтерактивний редактор онтології, який надає користувачеві можливість керувати структурою та наповненням UESO безпосередньо через графічний інтерфейс. На відміну від традиційних середовищ редагування онтологій (Protégé), дана сторінка інтегрована у веб-архітектуру Eco-City та враховує доменні специфіки екологічного моделювання.

1. Вибір усіх точкових джерел NO₂ з базовими параметрами

Запит повертає всі індивіди класу `ueso:PointSource`, для яких задано потік викидів NO₂ та геометрію в форматі WKT.

```

PREFIX ueso: <http://example.org/ueso#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>

SELECT ?source ?label ?rate ?geom
WHERE {
  ?source rdf:type ueso:PointSource ;
    ueso:hasLabel ?label ;
    ueso:hasEmission ?emission ;
    geo:hasGeometry ?g .

  ?emission ueso:emitsPollutant ueso:NO2 ;
    ueso:hasEmissionRate ?rate .

  ?g geo:asWKT ?geom .
}
ORDER BY ?label

```

?source	?label	?rate	?geom
exBoiler_P28	Котельня П-28	2.85	POINT(25.60 49.55)
exCHP_East	ТЕЦ «Східна»	5.10	POINT(25.65 49.53)

2. Джерела, задіяні у визначеному сценарії моделювання

Запит демонструє, як сценарій `exScenario_2025` пов'язується з множиною джерел через онтологічні властивості, описані в ABox.

```

PREFIX ueso: <http://example.org/ueso#>
PREFIX ex: <http://example.org/eco-city/>

SELECT ?source ?label ?type
WHERE {
  ex:Scenario_2025 ueso:includesSource ?source .

  ?source ueso:hasLabel ?label ;
    rdf:type ?type .

  # обмежуємося лише підкласами EmissionSource
  ?type rdfs:subClassOf* ueso:EmissionSource .
}
ORDER BY ?type ?label

```

?source	?label	?type
exBoiler_P28	Котельня П-28	ueso:Boiler
exTrafficRing	Транспортне кільце	ueso:LineSource

Рис. 4.11. Результати SPARQL-запитів

Завершальним компонентом є інтерактивний редактор онтології, який надає користувачеві можливість керувати структурою та наповненням UESO безпосередньо через графічний інтерфейс. На відміну від традиційних середовищ редагування онтологій (Protégé), дана сторінка інтегрована у веб-

архітектуру Eco-City та враховує доменні специфіки екологічного моделювання. Особливе значення має той факт, що сторінка реалізує концептуальну модель, яка безпосередньо пов'язана з механізмом генерації вхідних файлів AERMOD. Відповідно, кожна модифікація онтології перетворюється у зміни в параметрах моделювання, що забезпечує повну семантичну трасованість та відтворюваність обчислювальних експериментів.

Завдяки використанню єдиного онтологічного словника та механізмів логічного виведення UESO здатна слугувати центральною семантичною складовою міжплатформної взаємодії, забезпечуючи не лише обмін даними, а й узгодження смислів, структур та правил їх інтерпретації. Таким чином, онтологічна підсистема Eco-City формує фундамент для побудови інтегрованої урбаністичної екосистеми даних, яка підтримує проактивне, науково обґрунтоване та багатокритеріальне управління розвитком міста.

4.2.4. Особливості програмної реалізації модельних сценаріїв

Процес побудови, валідації та реалізації модельних сценаріїв у системі Eco-City є центральним компонентом функціональності платформи проактивного управління екосистемою міста. Архітектура сценарного моделювання базується на поєднанні семантично орієнтованої моделі даних (UESO), мікросервісної структури застосунку, а також спеціалізованих модулів трансляції та обчислення, що відповідають за інтеграцію з AERMOD, AERMET та BPIPPRM. У цьому підрозділі детально описано особливості програмної реалізації компонентів сценарного аналізу та наведено відповідні фрагменти коду.

Кожен сценарій у системі реалізовано як об'єкт, що відображає ключові параметри моделювання: множину джерел, множину рецепторів, набір метеорологічних даних, а також базовий сценарій для порівняння. Архітектурно сценарії поділяються на два типи: оцінювальні та оптимізаційні. Для обох типів використано спільну абстракцію Scenario. Після конфігурації сценарію система повинна гарантувати, що всі необхідні параметри визначені

коректно. Валідація виконується не на рівні реляційного сховища, а через механізми логічного виведення та правила SHACL, що забезпечує перевірку відповідності аксіомам UESO (рис. 4.12).

Ключовим етапом є трансляція семантичного опису в синтаксичний формат AERMOD.INP. Трансляція реалізована як окремий мікросервіс TranslatorService, який викликається після валідації сценарію (рис. 4.13).

```
ex:PointSourceShape
  a sh:NodeShape ;
  sh:targetClass ueso:PointSource ;
  sh:property [
    sh:path ueso:hasStackHeight ;
    sh:datatype xsd:decimal ;
    sh:minCount 1 ;
  ] ;
  sh:property [
    sh:path ueso:hasStackDiameter ;
    sh:datatype xsd:decimal ;
    sh:minCount 1 ;
  ] .
```

Рис. 4.12. Лістинг коду реалізації правила валідації PointSource

```
private String generateStack(EmissionSource src) {
  return String.format(
    "SO %s\n LOCATION %f %f %f\n STACK %f %f %f %f\n",
    src.getName(),
    src.getX(), src.getY(), src.getHeight(),
    src.getStackHeight(),
    src.getStackDiameter(),
    src.getStackVelocity(),
    src.getStackTemperature()
  );
}
```

Рис. 4.13. Лістинг коду валідації даних

Уся логіка трансляції побудована на правилах, що витягуються з онтології (через SPARQL-запити), забезпечуючи гнучкість та незалежність від структури реляційної БД.

Запуск AERMOD у системі Eco-City реалізовано через контейнеризований модуль з інтерфейсом REST (рис. 4.14). Це забезпечує відмовостійкість, повторюваність та можливість масштабування. Парсинг результатів представлено в лістингу на рисунку 4.15.

```
docker run --rm \
  -v /data/scenarios/123:/work \
  eco-aermod:latest \
  ./aermod AERMOD.INP > output.log
```

Рис. 4.14. Лістинг коду запуску AERMOD у Docker-контейнері

```
def parse_plt(file_path):
    data = []
    with open(file_path) as f:
        for line in f:
            parts = line.split()
            if len(parts) == 6:
                x, y, z, conc = float(parts[1]), float(parts[2]), float(parts[3]), float(parts[5])
                data.append({"x": x, "y": y, "z": z, "conc": conc})
    return data
```

Рис. 4.15. Лістинг парсингу файлу результатів (.PLT)

Візуалізація виконується окремим модулем MapViewer, який побудовано на бібліотеці Leaflet з підтримкою heatmap-шарів та ізоліній, рисунок 4.16.

```
const heat = L.heatLayer(
  concentrations.map(p => [p.y, p.x, p.conc]),
  { radius: 18, blur: 12, maxZoom: 17 }
).addTo(map);
```

Рис. 4.16. Лістинг візуалізації heatmap

Реалізована модель сценарного аналізу має низку принципових переваг: сценарії повністю відокремлені від механіки обчислень; логічна валідація базується на онтології, а не на класичних SQL-обмеженнях; трансляція $F(s)$ допускає розширення без зміни логіки AERMOD; масштабування реалізується на рівні мікросервісів; обробка результатів узгоджується з OGC-стандартами для геопросторових даних; сценарії є повністю відтворюваними, а їх компоненти – прозорими для аудиту.

Запропонована програмна реалізація модельних сценаріїв забезпечує системний, модульний та формально обґрунтований підхід до

автоматизованого екологічного моделювання. Вона поєднує семантичну строгість UESO, алгоритмічну керованість функції трансляції $F(s)$, масштабованість контейнеризованих обчислень та інтерактивність геопросторової візуалізації – створюючи основу для проактивного управління міським середовищем.

4.3 Організація графічного інтерфейсу інтелектуалізованої системи для проактивного управління екосистемою міста

Графічний інтерфейс у сучасних інтелектуалізованих системах перестає бути лише засобом візуального представлення даних. У контексті проактивного управління екосистемою міста він виконує роль інтеграційного механізму, що поєднує гетерогенні обчислювальні модулі, онтологічні моделі знань та складні алгоритми прийняття рішень у єдине кероване середовище взаємодії користувача з системою. Для цього необхідно не лише створити привабливий та зручний інтерфейс, а й забезпечити його логічну відповідність архітектурі мікросервісів, яка лежить в основі всієї системи.

Вибір багатосторінкової архітектури веб-застосунку є принциповим рішенням, що виходить із вимог надійності, відмовостійкості та стабільності. Кожна сторінка системи вирішує завершене завдання: формування сценарію, управління джерелами викидів, попередня обробка метеорологічних даних, запуск моделювання або візуалізація результатів. Така сегментація дозволяє повністю ізолювати внутрішні стани між сторінками, забезпечуючи чистоту логіки та передбачуваність поведінки системи. Це особливо важливо в області моделювання розсіювання забруднювачів, де будь-які некоректні залишкові дані, помилки кешування або випадкове змішування параметрів могли б призвести до неправильних висновків чи хибних управлінських рекомендацій.

Кожна сторінка не лише має власну структурну логіку, а й відображає певний фрагмент предметної області. Загальний опис наявних сторінок представлено у таблиці 4.1.

Сторінка запуску моделювання, у свою чергу, звертається до обчислювальних сервісів через API-шлюз і відображає процес виконання AERMOD у формі підтримуваних журналів, що підвищує прозорість операцій. Візуалізаційний модуль представляє результати у вигляді інтерактивної карти з накладенням ізоліній концентрацій, теплових полів та маркерів джерел, що дозволяє інтерпретувати результати в просторі та часі.

Таблиця 4.1.

Основні сторінки системи

№	Сторінка / Модуль	Основні функції
1	HomePage	Загальна панель, навігація, короткі індикатори стану системи
2	ScenarioListPage	Перегляд списку сценаріїв, пошук, фільтрація
3	ScenarioEditorPage	Створення нового сценарію, редагування параметрів
4	ScenarioComparePage	Порівняння базового і альтернативних сценаріїв
5	SourceCatalogPage	Список джерел, категоризація за типами (точкові, лінійні тощо)
6	SourceEditorPage	Редагування характеристик джерела, його геометрії
7	ReceptorListPage	Перегляд та управління рецепторами
8	MeteorologyUploadPage	Завантаження сирих метеоданих
9	AERMETRunPage	Запуск AERMET, перегляд журналів
10	ModelingRunPage	Запуск AERMOD, передача AERMOD.INP
11	ModelingResultsPage	Відображення результатів, таблиці, графіки
12	MapViewPage	Інтерактивна карта, ізолінії, heatmap
13	OptimizationRunPage	Запуск оптимізації, статус виконання
14	TradeoffExplorerPage	Візуалізація Парето-фронтів
15	OntologyBrowserPage	Перегляд класів та екземплярів онтології UESO
16	DataRepositoryMonitorPage	Моніторинг стану сховищ, журнал транзакцій

Завдяки мікросервісній архітектурі інтерфейс не повинен знати нічого про внутрішній устрій окремих сервісів. Взаємодія з ними здійснюється через єдиний API-шлюз, який виступає посередником між користувачем і складною системою моделювання (рис. 4 17). Це усуває залежність інтерфейсу від способу, у який реалізовані моделі, та дозволяє змінювати, розширювати або оновлювати обчислювальні компоненти без необхідності редизайну клієнтської частини. У результаті інтерфейс стає універсальним вікном доступу до екосистеми сервісів, що забезпечує проактивне управління.

Логічна структура інтерфейсу відображає природний перебіг роботи аналітика екологічної служби або міського планувальника. Система відкривається головною сторінкою, де користувач отримує стислу інформацію про доступні сценарії, поточний стан підсистем, журнали останніх обчислень та можливості створення нового дослідницького експерименту. Перехід до модуля сценаріїв дозволяє сформувати структуру майбутнього моделювання. Користувач не лише обирає джерела та рецептори—інтерфейс забезпечує глибшу інтеграцію з онтологічною моделлю UESO, автоматично підказуючи відповідні типи об'єктів, контролюючи їхню семантичну сумісність та запобігаючи помилкам у формуванні вхідних параметрів. Такий підхід переводить інтерфейс із рівня «інструменту для введення даних» у рівень «інтелектуального асистента», який забезпечує відповідність дій користувача структурі знань предметної області.

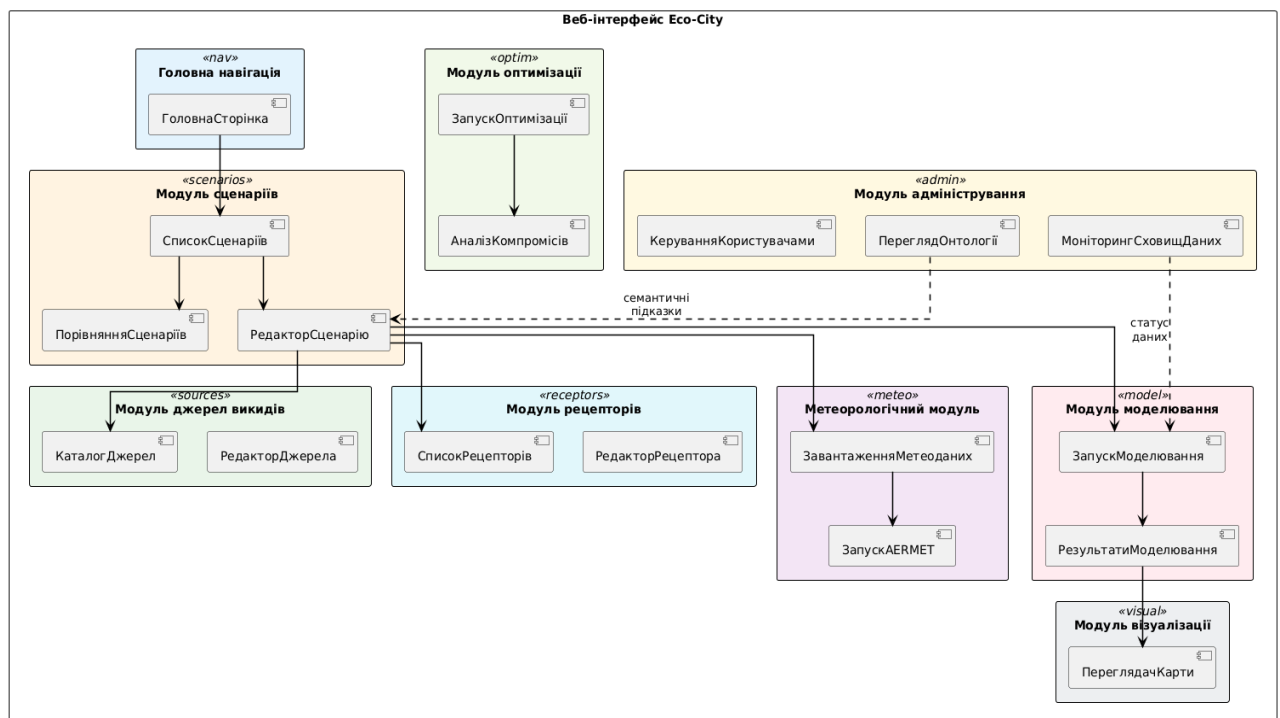


Рис. 4.17. Загальна схему взаємодії програмних модулів в рамках організації графічного інтерфейсу користувача системи

Підготовка метеорологічних даних та їх перетворення у формат,

придатний для AERMOD, традиційно є складною процедурою, що вимагає окремих утиліт. Система інкапсулює цю складність, розміщуючи весь цикл обробки на присвяченій сторінці. Користувач бачить процеси AERMET не як набір команд чи консольних операцій, а як структурований робочий етап із документуванням створених файлів, відображенням статусів та журналів. Завдяки цьому формування метеорологічних артефактів стає не лише прозорим, а й контрольованим у межах одного інтерфейсу.

Однією з найважливіших сторінок системи є модуль візуалізації. У проактивному управлінні найвагомішим є не сам факт отримання числових значень концентрацій забруднювачів, а можливість побачити й інтерпретувати їх у просторі. Модуль поєднує картографічне відображення, графіки та табличні зрізи таким чином, щоб користувач міг аналізувати результати моделювання комплексно. Карта дозволяє ідентифікувати просторові аномалії, графіки – дослідити часову динаміку або сезонні коливання, а таблиці – отримати точні значення, необхідні для звітів чи нормативних оцінок. Усі ці форми представлення даних пов'язані між собою: зміна параметрів карти одразу оновлює таблиці, а вибір окремої точки рецептора в таблиці синхронно позначає її на карті.

Організація графічного інтерфейсу системи Eco-City не обмежується формальним розподілом функцій між сторінками. Її основу становить когнітивний дизайн, орієнтований на те, яким чином фахівці екологічних служб, містобудівники та аналітики сприймають інформацію, формують ментальні моделі міської екосистеми і приймають рішення. Когнітивний дизайн тут проявляється насамперед у роботі з інформаційним навантаженням. Моделі на кшталт AERMOD, препроцесори AERMET та BPIPPRM, набір параметрів для кожного джерела й рецептора здатні легко перевантажити користувача деталями. Тому інтерфейс будується за принципом поступового розкриття складності: на перших кроках користувач бачить лише ключові параметри сценарію, найважливіші характеристики джерел і просту, зрозумілу карту.

Модель взаємодії «користувач – система» в Eco-City можна описати як цикл сприйняття, інтерпретації та дії. Користувач бачить певний стан міської екосистеми у вигляді карти, графіків чи таблиць; інтерпретує його з позиції своїх професійних знань; формулює гіпотезу «що станеться, якщо...» і трансформує її у сценарій моделювання. Система, у свою чергу, відповідає не просто «числом», а новим візуальним станом – новою конфігурацією забруднень, індикаторів ризику, діаграм. Результати моделювання повинні бути достатньо наочними, щоб їх можна було обговорювати на нарадах, демонструвати на спільних переглядах, включати до презентацій та звітів.

У сукупності ці рішення формують інтерфейс, який не просто «обслуговує» мікросервісну архітектуру, а органічно продовжує її філософію: модульність, прозорість, семантичну узгодженість і орієнтацію на проактивне управління міською екосистемою.

4.4 Тестування системи проактивного управління екосистемою міста

4.4.1. Функціональне тестування системи

У рамках даної роботи цей процес реалізовано за методологією «чорної скриньки», що передбачає перевірку коректності роботи вхідних та вихідних інтерфейсів без аналізу внутрішньої структури програмного коду.

Головна сторінка системи реалізована у вигляді центральної інформаційної панелі, яка консолідує ключові показники стану підсистем (активні сценарії, статус сервісів, актуальність метеоданих) та виступає навігаційним ядром для доступу до функціональних модулів (рис. 4.18).

Інтегрований журнал подій забезпечує моніторинг стабільності обчислювальних процесів, гарантуючи прозорість та контрольованість роботи системи підтримки прийняття рішень.

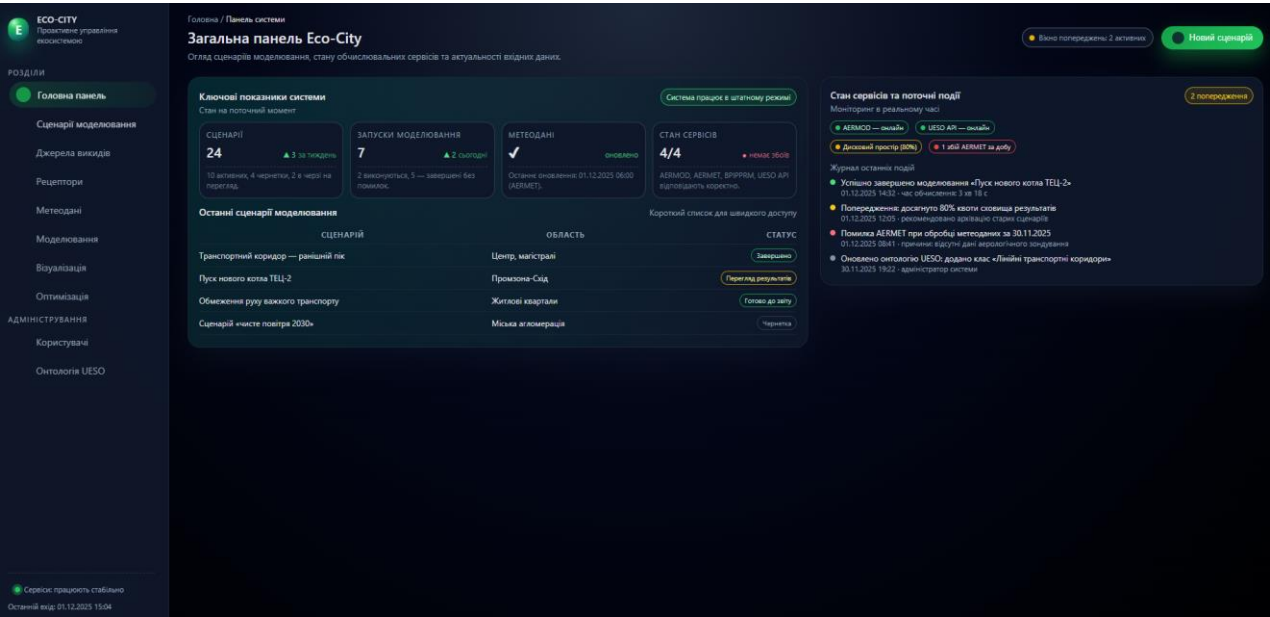


Рис. 4.18. Головна сторінка системи

Сторінка ScenarioListPage у веб-орієнтованій системі ESO-City виконує роль центрального інтерфейсного вузла, що забезпечує управління сценаріями моделювання як базовими аналітичними одиницями у процесі проактивного управління міською екосистемою (рис. 4.19).

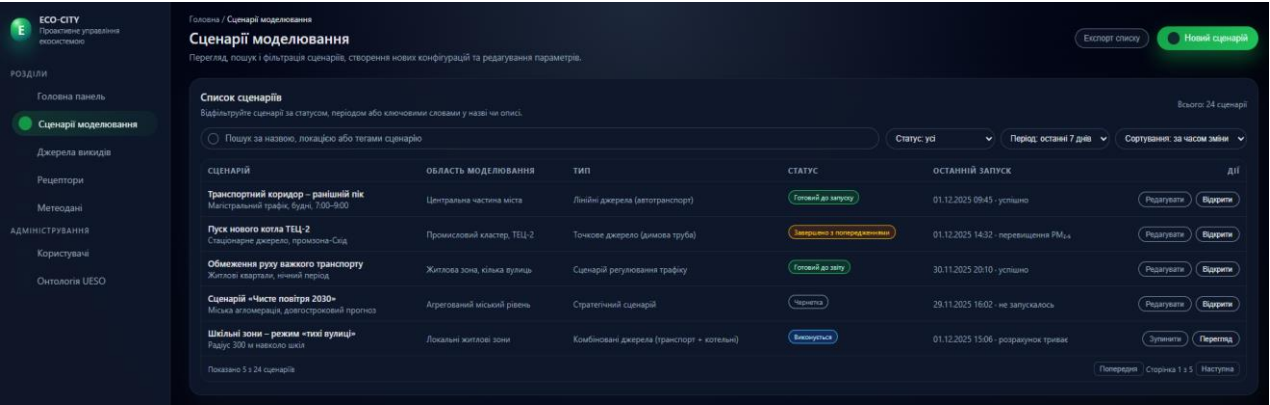


Рис. 4.19. Сторінка із сценаріями моделювання

Інтерфейс управління сценаріями реалізовано як єдиний робочий простір, що інтегрує функції повного життєвого циклу моделювання: від створення та модифікації конфігурацій до їх запуску (рис. 4.20).

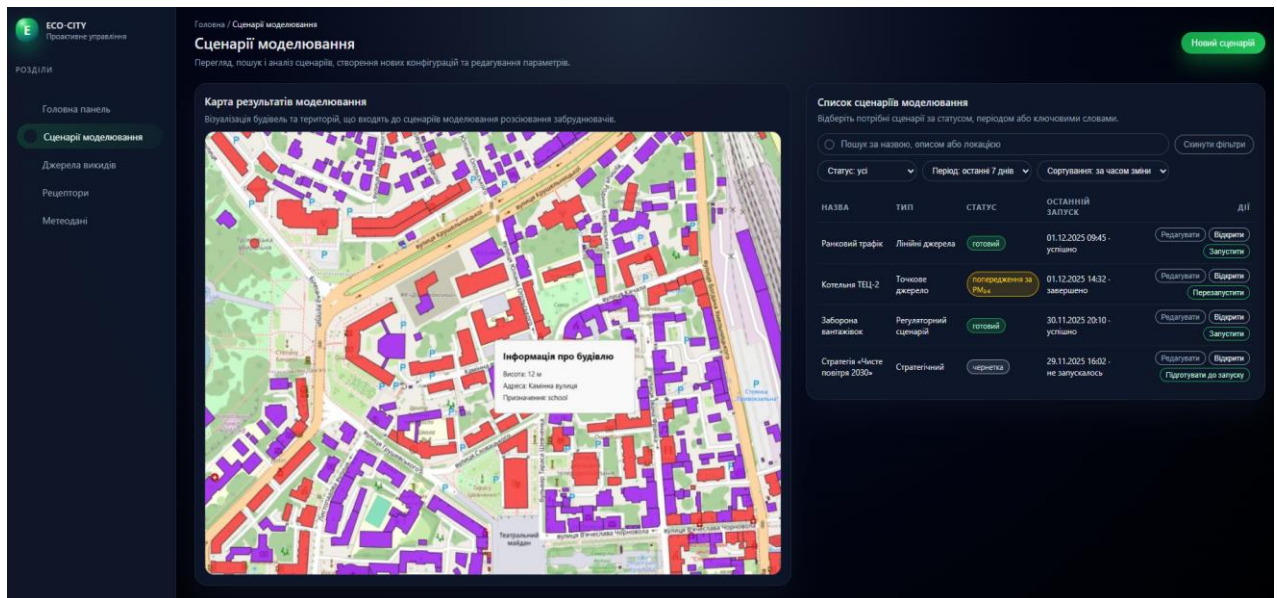


Рис. 4.20. Сторінка із результатами моделювання

Ключовою особливістю є синхронізація табличних даних з інтерактивною картою, що забезпечує безперервний просторовий контекст і дозволяє користувачеві миттєво співвідносити параметри сценаріїв з топологією міського середовища, підвищуючи ефективність аналітичної роботи.

Модуль ScenarioListPage реалізує функцію координаційного центру, що інтегрує сценарні, просторові та операційні аспекти управління. Забезпечення прямого доступу до редагування та запуску обчислень мінімізує операційну затримку (латентність), що є критичним для проактивних систем. Ергономічна організація інтерфейсу гарантує високу інформативну щільність та оперативний контроль над повним життєвим циклом моделювання – від параметризації до первинного аналізу результатів.

Модуль ReceptorListPage (рис. 4.21) забезпечує управління мережею контрольних точок шляхом інтеграції атрибутивних даних із геопросторовою візуалізацією. Використання семантичного кодування маркерів дозволяє миттєво ідентифікувати типологію рецепторів (наземні, висотні, критичні), а система контекстних анотацій надає оперативний доступ до характеристик локації. Такий підхід реалізує принципи когнітивної ергономіки, полегшуючи

оцінку впливу джерел викидів на чутливі елементи міської інфраструктури.

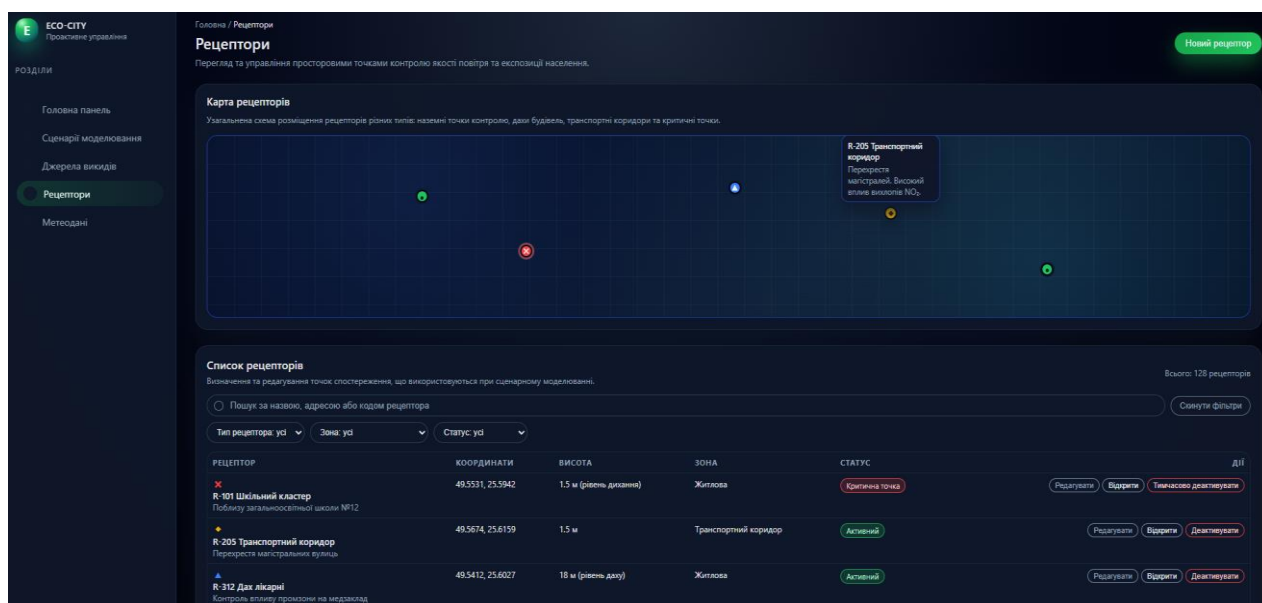


Рис. 4.21. Сторінка перегляду та управління рецепторами

Атрибутивна складова підсистеми реалізована через табличний інтерфейс, що динамічно синхронізується з картографічними маркерами за принципом єдиного візуального кодування. Кожен запис містить розширений набір метаданих (геопросторові координати, висоту вимірювання, тип функціональної зони, статус активності), що дозволяє користувачеві миттєво співвідносити аналітичні параметри з їх топологічною локалізацією.

Функціональна архітектура модуля забезпечує повний життєвий цикл управління мережею моніторингу: від багатофакторного пошуку та фільтрації до редагування параметрів і зміни статусів рецепторів. Це перетворює сторінку на гнучкий інструмент препроцесингу даних, дозволяючи оперативно адаптувати конфігурацію мережі спостереження (наприклад, фокусуючись виключно на селітебних зонах або транспортних коридорах) відповідно до специфічних вимог конкретного сценарію моделювання, що є необхідною умовою для забезпечення точності прогнозних розрахунків.

Модуль MeteorologyPreviewPage реалізує етап верифікації та валідації нормалізованих метеорологічних рядів перед їх передачею у препроцесор AERMET (рис. 4.22).

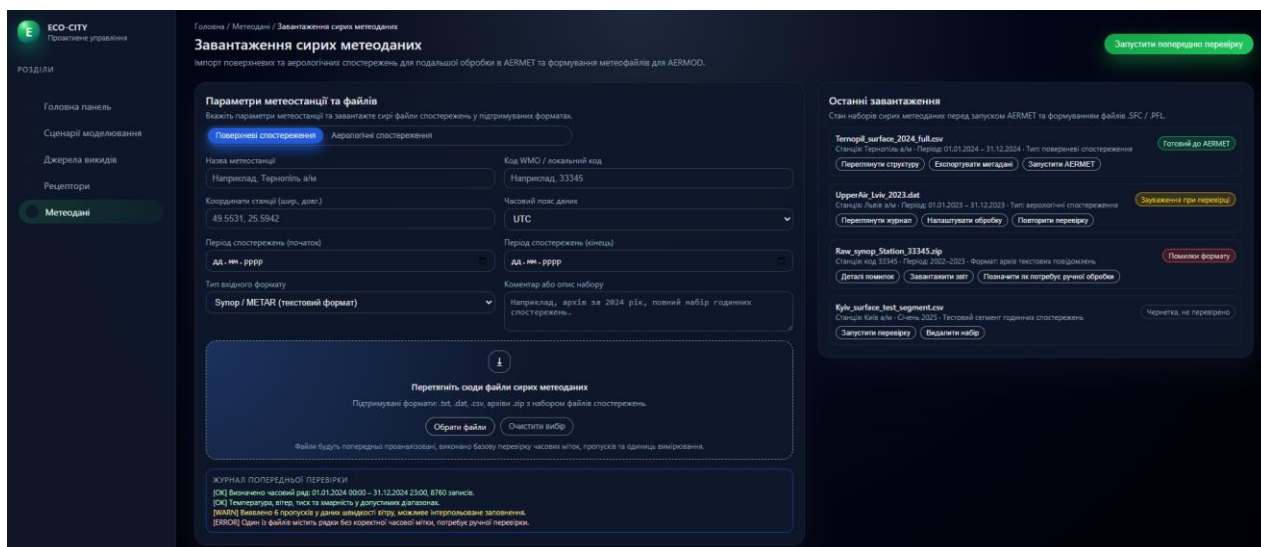


Рис. 4.22. Сторінка перегляду та опрацювання метеорологічних даних

На відміну від процедур первинного імпорту, даний компонент забезпечує глибокий змістовний аналіз структурованих годинних записів, виступаючи шлюзом контролю якості для регуляторної моделі. Це дозволяє нівелювати ризики використання некоректних даних ще до етапу ресурсоємних обчислень.

Функціональна організація модуля базується на дворівневій схемі аналізу. Верхній рівень надає агреговані метрики репрезентативності вибірки (повнота рядів, ідентифікатори станцій, загальний статус валідності), що дозволяє миттєво оцінити готовність набору даних (рис. 4.23). Детальний рівень реалізовано через табличний перегляд фізичних параметрів (швидкість вітру, температура, клас стійкості атмосфери) з інтегрованими маркерами якості. Така система автоматизованої індикації аномалій та пропусків забезпечує експерту інструментарій для прийняття обґрунтованого рішення щодо придатності метеорологічного сценарію для подальшого моделювання.

Модуль ModelingRunPage (рис. 4.24) виконує функцію центрального керуючого вузла, що забезпечує автоматизацію повного циклу сценарного моделювання: від фінальної конфігурації параметрів до запуску обчислювального ядра AERMOD.

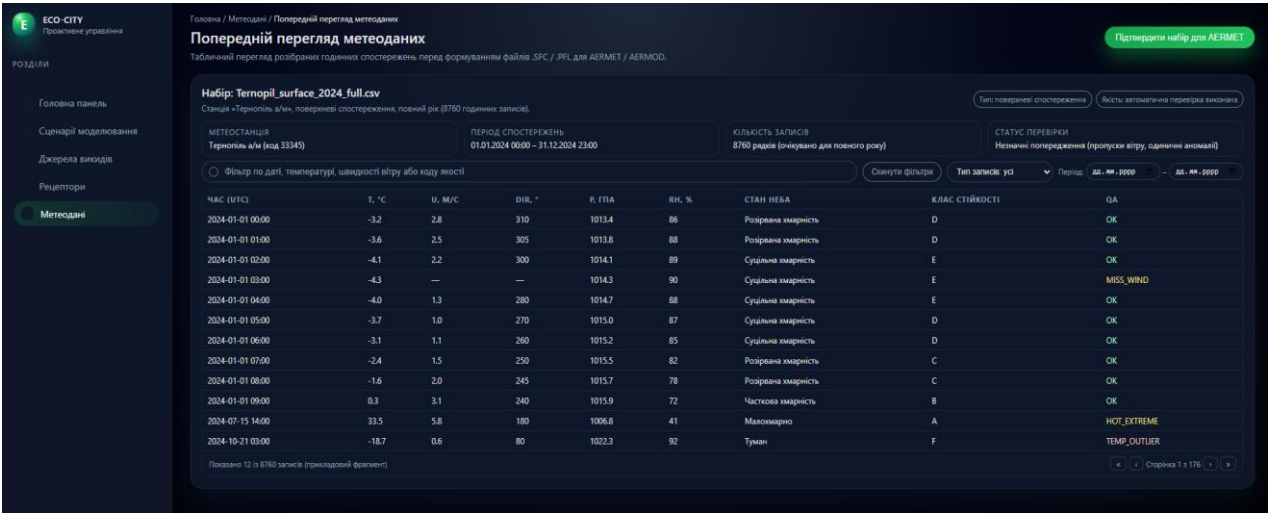


Рис. 4.23. Сторінка перегляду метеорологічних даних

Ключовим завданням модуля є трансляція семантичних структур онтології UESO у синтаксис керуючого файлу AERMOD.INP. Це створює необхідний шар абстракції, дозволяючи користувачеві оперувати високорівневими поняттями (сценарій, набір метеоданих, тип полютанта) без необхідності ручного втручання у низькорівневі налаштування регуляторної моделі.

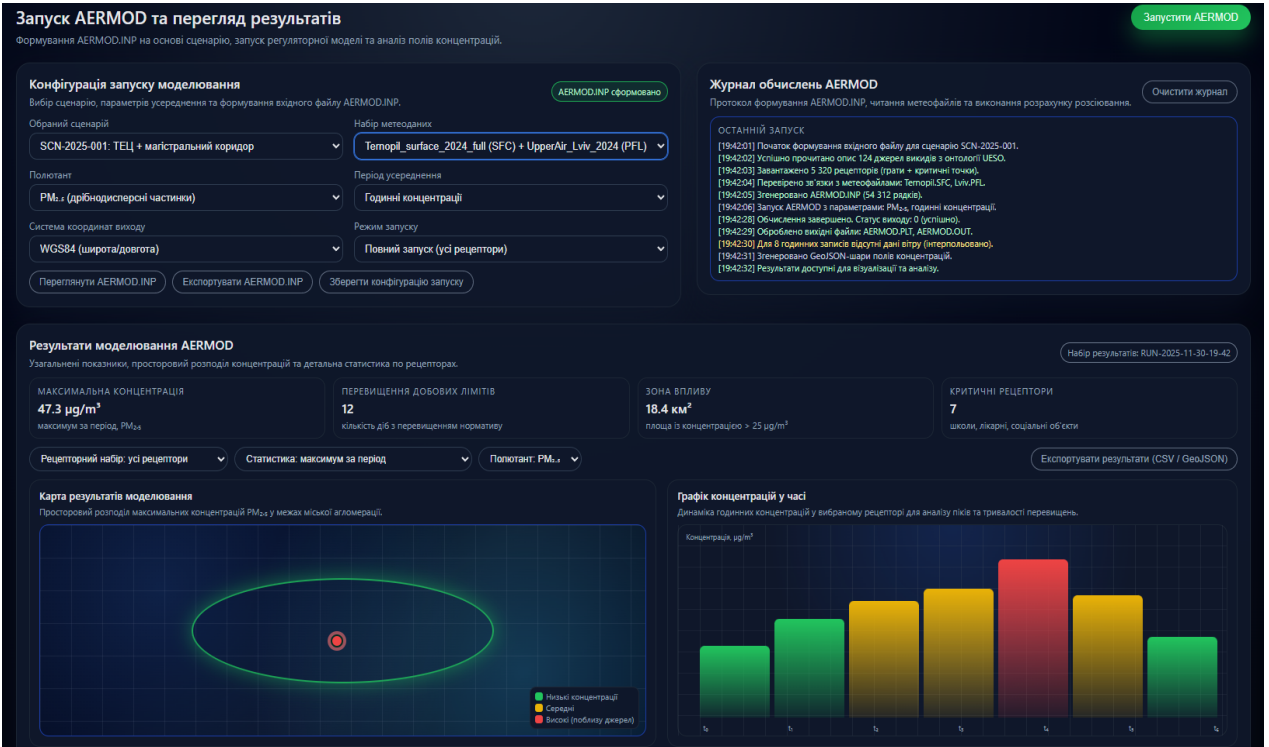


Рис. 4.24. Сторінка перебігу процесу моделювання ModelingRunPage

Система автоматично забезпечує валідацію вхідних даних (файлів .SFC/.PFL) та оркестрацію запуску розрахунків, гарантуючи коректність та відтворюваність результатів моделювання.

Забезпечення транспарентності та нормативної відповідності розрахунків реалізовано через інтегровану підсистему логування (Execution Audit Trail). Журнал автоматично фіксує повний операційний конвеєр: від парсингу онтологічних структур та генерації AERMOD.INP до бінарного виконання ядра та постпроцесингу результатів. Це гарантує повну простежуваність (traceability) експериментів, дозволяючи однозначно асоціювати отримані геопросторові дані з конкретною конфігурацією вхідних параметрів, що є обов'язковою умовою для верифікації регуляторних рішень.

Аналітичний блок модуля забезпечує багаторівневу інтерпретацію результатів моделювання (рис. 4.25). На макрорівні формується зведення інтегральних індикаторів екологічного ризику (максимальні концентрації, площа зон перевищення ГДК, кількість критичних рецепторів), що дозволяє оперативно оцінити загальну безпеку сценарію.

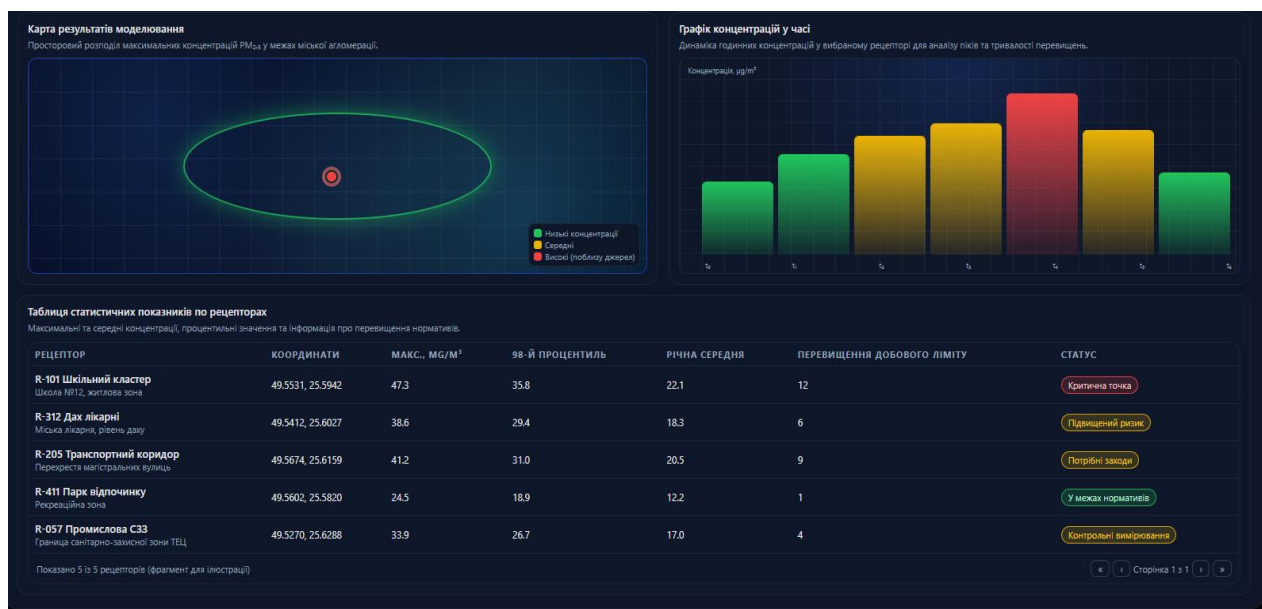


Рис. 4.25. Сторінка оцінювання статистичних показників по рецепторах

На мікрорівні інструментарій часового аналізу візуалізує динаміку

концентрацій у контрольних точках, дозволяючи ідентифікувати пікові епізоди забруднення та корелювати їх з метеорологічними умовами. Такий комплексний підхід створює емпіричне підґрунтя для розробки цільових компенсаційних заходів у рамках проактивного управління.

Таким чином, *ModelingRunPage* є не просто технічним інтерфейсом для запуску *AERMOD*, а інтегрованим робочим середовищем для сценарного аналізу, яке поєднує конфігурацію моделі, протокол її виконання та багатовимірну візуалізацію результатів. У загальній архітектурі системи *Eco-City* ця сторінка фіксує момент переходу від знання-орієнтованого формування вхідних даних до обґрунтованого, візуально підтриманого прийняття управлінських рішень щодо екологічної безпеки міста.

Модуль *MapViewPage* виступає центральним компонентом підсистеми геопросторової візуалізації, забезпечуючи трансформацію чисельних результатів моделювання *AERMOD* у візуальні аналітичні образи. Інтерфейс підтримує повний когнітивний цикл експерта: від вибору сценарію до ідентифікації критичних зон (4.26). Завдяки механізмам динамічної фільтрації параметрів (тип полютанта, часовий зріз, статистична метрика: максимум, процентиль, середнє), модуль дозволяє змінювати «оптику» дослідження, перетворюючи статичні карти на інтерактивний інструмент діагностики стану міської екосистеми.

Технологічну основу модуля становить реалізація принципу багат шарової суперпозиції даних. Візуалізація безперервного скалярного поля концентрацій за допомогою градієнтної заливки дозволяє оперативно локалізувати епіцентри забруднення. Для кількісної оцінки застосовано шар ізоліній, що дискретизує поле та окреслює межі зон перевищення нормативів. Інтеграція маркерів рецепторів забезпечує просторову прив'язку розрахункових величин до об'єктів критичної інфраструктури, що є необхідною умовою для переходу від абстрактного моделювання до прийняття адресних управлінських рішень.

Додатковий рівень аналітики реалізовано через механізм ідентифікації

та ранжування кластерів забруднення («гарячих точок»). Структурування виявлених аномалій у табличному вигляді із зазначенням координат центроїдів, площі та категорії ризику створює підґрунтя для операційного управління: формування пріоритетних планів реагування (обмеження емісії, оптимізація режимів роботи) та документальної фіксації результатів сценарного аналізу.

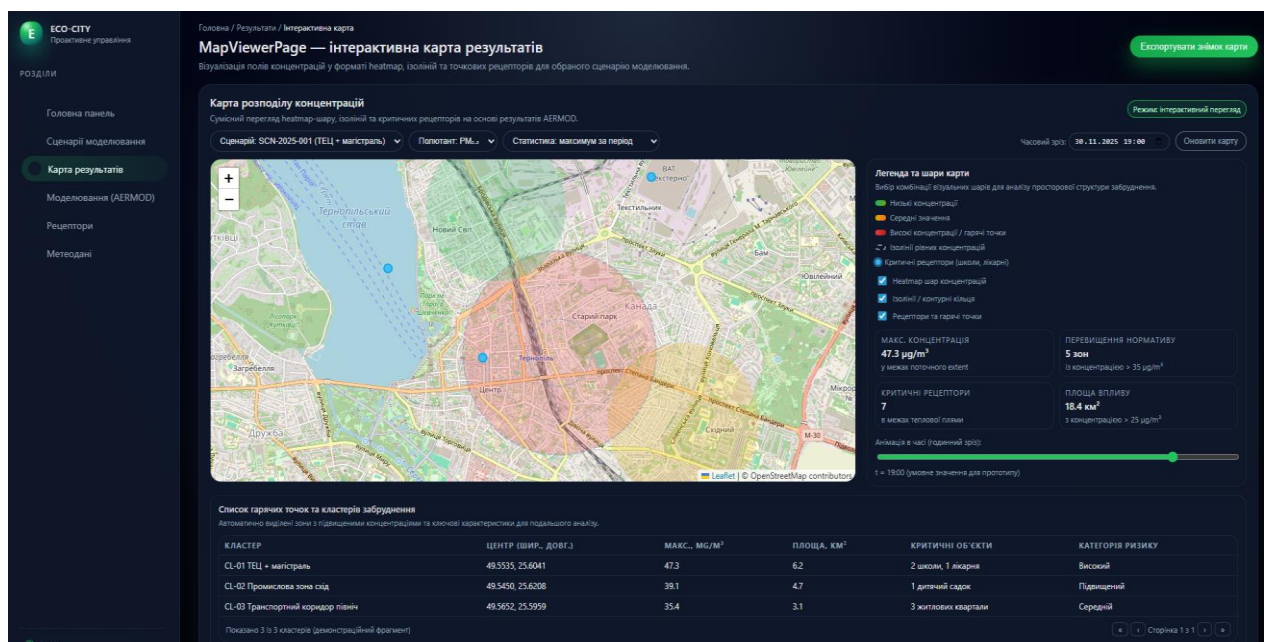


Рис. 4.26. Сторінка MapViewerPage у системі Eco-City

Модуль MapViewerPage реалізує функцію інтерактивного геопросторового інтерфейсу, що трансформує результати математичного моделювання у візуальні аналітичні моделі шляхом синтезу теплових карт, ізоліній та метрик забруднення. Це забезпечує підтримку динамічного сценарного аналізу, дозволяючи в реальному часі оцінювати просторові екологічні ризики при варіації параметрів джерел викидів або метеоумов, що є необхідною умовою проактивного управління.

Модуль DataRepositoryMonitorPage (рис. 4.27) реалізує функцію централізованого моніторингу гетерогенної інфраструктури збереження даних, консолідує метрики стану трьох ключових підсистем: реляційної бази (EcoCityDB), семантичного сховища (UESO Triple Store) та файлового

репозиторію результатів (ModelOutput Storage). Інтеграція цього інтерфейсу в загальний контур управління дозволяє оперативно оцінювати експлуатаційну готовність ресурсів (латентність, рівень утилізації дискового простору, статус доступності) перед ініціацією ресурсоємних циклів сценарного моделювання.

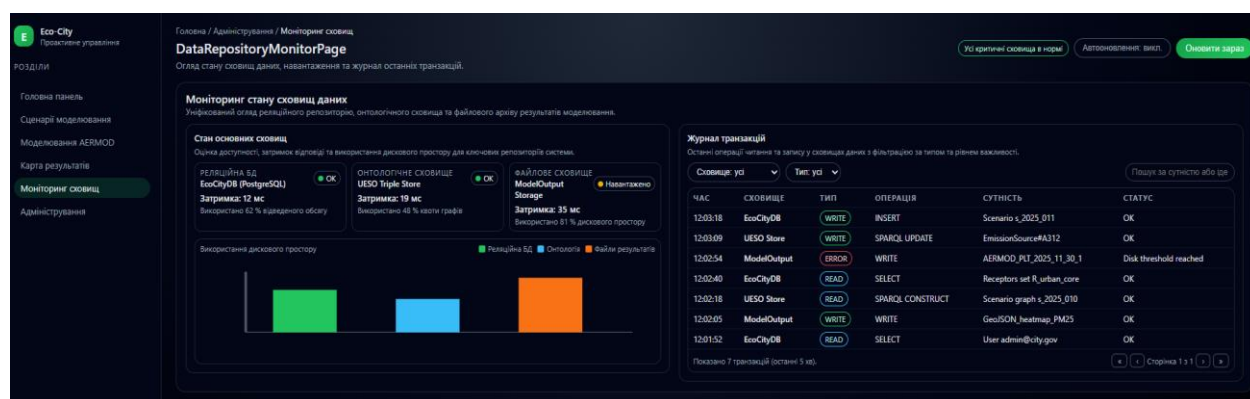


Рис. 4.27. Моніторинг та управління сховищем даних

У функціональному аспекті модуль виступає інструментом превентивної діагностики обчислювального конвеєра. Оскільки стабільність роботи препроцесорів (AERMET, BPIPPRM) та ядра AERMOD критично залежить від синхронізації сховищ, своєчасне виявлення деградації продуктивності або дисбалансів у накопиченні даних дозволяє попередити системні збої. Графічна візуалізація динаміки використання ресурсів забезпечує адміністратора аналітичною інформацією для планування ємності системи та оптимізації політик зберігання.

Другим концептуальним компонентом цієї сторінки є журнал транзакцій, який забезпечує протоколювання ключових операцій читання й запису в усіх типах сховищ. Кожний запис журналу містить часову мітку, тип сховища, характер операції (SELECT, INSERT, SPARQL UPDATE, запис вихідного файлу тощо), ідентифікатор сутності (сценарію, джерела викидів, набору рецепторів, вихідного GeoJSON чи PLT-файлу) та статус виконання. Наявність фільтрації за типом сховища, видом операції та текстового пошуку дозволяє швидко реконструювати пов'язану послідовність дій для

конкретного сценарію моделювання та виявляти потенційні «вузькі місця» або аномальні ситуації (наприклад, відмову запису результатів через перевищення дискової квоти). У дисертаційному контексті це забезпечує важливу властивість – трасованість усіх етапів життєвого циклу даних, що є критичною для відтворюваних наукових обчислень.

Це створює передумови для проактивного управління не тільки міським середовищем, але й самою інформаційно-обчислювальною інфраструктурою, на якій базуються моделі розсіювання, онтологія UESO та інтелектуалізовані сервіси підтримки прийняття рішень.

4.4.2. Тестування продуктивності системи

В таблиці 4.2 наведено часову ефективність використання в порівнянні з комерційним пакетом AERMOD View (Lakes Environmental), що є стандартом де-факто в галузі. Акцент зроблено на тому, що в AERMOD View більшість операцій виконуються через «майстри» та діалогові вікна вручну, тоді як у запропонованій системі вони автоматизовані на рівні бекенду.

Як видно з табл. 4.2, застосування системи «Есо-City» дозволяє скоротити операційний час роботи експерта в середньому у 24 рази порівняно з використанням стандартного інструментарію AERMOD View.

Основна перевага досягається за рахунок усунення ручних операцій ETL: замість ручного конвертування форматів даних через Wizards у AERMOD View, система використовує автоматизовані пайплайни. Також, в AERMOD View кожне нове джерело потрібно додавати вручну або імпортувати через Excel. В «Есо-City» конфігурація сценарію формується запитом до онтології, що миттєво структурує вхідний файл AERMOD.INP. AERMOD View потребує зовнішніх GIS-інструментів (Google Earth, ArcGIS) для якісної візуалізації на карті місцевості, тоді як «Есо-City» виконує рендеринг геопросторових шарів автоматично як фінальний етап мікросервісного ланцюжка.

Таблиця 4.2

Порівняльний аналіз трудомісткості етапів моделювання в середовищі
AERMOD View та системі «Eco-City»

Етап сценарного аналізу	Програмний комплекс AERMOD View (Lakes Environmental)	Інтелектуалізована система «Eco-City»	Коефіцієнт скорочення витрат часу
1. Підготовка метеоданих (AERMET)	Ручний режим: Імпорт «сирих» файлів (SAMSON/TD-1440), ручне задання параметрів підстилаючої поверхні по секторах у AERMET View Wizard. Час: ~45–60 хв	Автоматизований: Автоматичний парсинг, валідація та генерація .SFC/.PFL файлів на основі попередньо налаштованих онтологічних профілів місцевості. Час: ~2 хв	~25.0
2. Конфігурація джерел та рецепторів	Напівавтоматичний: Ручне введення координат або імпорт статичних файлів Excel/CAD. Відсутність зв'язку з атрибутивними базами даних у реальному часі. Час: ~90–120 хв	Семантичний: Динамічна вибірка об'єктів з онтології UESO, автоматичне формування сітки рецепторів навколо зон інтересу. Час: ~5 хв	~20.0
3. Пост-процесинг та візуалізація	Ручний експорт: Використання утиліти Plot View, налаштування ізоліній, експорт у Google Earth (.kml) для накладання на карту міста. Час: ~30–40 хв	Інтегрований Web-GIS: Миттєва генерація тайлів Heatmap та інтерактивних ізоліній безпосередньо у браузері після завершення розрахунку. Час: < 1 хв	>30.0
Разом (операційний час експерта)	~165–220 хв (≈ 3,5 години)	~8 хв	~24.0

Тестування продуктивності підтвердило надійність запропонованої архітектури (SOA/Kubernetes) та що Wrapper працює коректно.

Система демонструє стабільну роботу при навантаженні до 50 одночасних запитів на моделювання, що є достатнім для потреб муніципальних екологічних служб. Зростання часу відгуку при 100 користувачах пов'язане з чергою задач на обчислювальні вузли AERMOD (CPU-bound задачі), проте механізм оркестрації запобігає «падінню» системи (успішність > 98%).

Таблиця 4.3

Показники продуктивності системи при різному рівні конкурентного навантаження

Кількість одночасних користувачів (Concurrent Users)	Середній час відгуку API (Latency), мс	Успішність виконання сценаріїв (Success Rate), %	Завантаження CPU кластера, %	Використання RAM, Гб
1 (Базовий рівень)	120	100%	5%	2.4
10 (Робоча група)	450	100%	18%	4.1
50 (Пікове навантаження)	1250	99.8%	65%	12.8
100 (Стрес-тест)	4800	98.5%	92%	28.5

Висновки до розділу 4

У розділі вирішено задачу проектування, програмної реалізації та експериментальної перевірки інтелектуалізованої системи «Есо-City», призначеної для проактивного управління екосистемою міста.

Основні результати, отримані в цьому розділі:

1. Спроековано гнучку сервіс-орієнтовану архітектуру, ключовим елементом якої є розробка спеціалізованих сервісів-обгорт. Це дозволило успішно інтегрувати ядро регуляторної моделі AERMOD у сучасне веб-середовище, забезпечивши автоматизацію повного циклу розрахунків без необхідності ручного втручання в консольні процеси. Розгортання системи в кластері Kubernetes гарантує динамічне масштабування обчислювальних ресурсів при зростанні навантаження.

2. Програмно реалізовано підсистему збереження даних на основі гібридної моделі. Поєднання семантичного сховища (для онтології UESO), реляційної бази даних з розширенням PostGIS (для геометрії) та об'єктного файлового сховища забезпечило реалізацію концепції «єдиного джерела істини». Такий підхід унеможливорює розсинхронізацію даних та дозволяє автоматично валідувати сценарії моделювання ще до початку ресурсоємних розрахунків.

3. Розроблено інтерактивний веб-інтерфейс користувача із застосуванням UX-патернів, спрямованих на зниження когнітивного навантаження на диспетчера. Реалізовані інструменти візуалізації (теплові карти забруднення, ізолінії, інтерактивні шари забудови) дозволяють оперативно інтерпретувати результати моделювання та приймати обґрунтовані управлінські рішення без необхідності глибокого аналізу текстових звітів.

4. Працездатність системи підтверджено шляхом вирішення практичної задачі оптимізації розміщення джерел теплопостачання у мікрорайоні «Сонячний» (м. Тернопіль). Результати навантажувального тестування засвідчили суттєве підвищення продуктивності: завдяки автоматизації підготовки даних та розпаралелюванню обчислень час отримання розрахункового сценарію скорочено з годин (при ручному режимі) до хвилин, що робить систему придатною для оперативного управління.

ВИСНОВКИ

В дисертаційній роботі вирішено важливу науково-прикладну проблему підвищення ефективності та оперативності управління екосистемою урбанізованих територій шляхом розробки моделей, методів та програмних засобів інтелектуалізованої системи для проактивного сценарного аналізу.

Основні наукові та практичні результати роботи полягають у таких:

1. Проведено критичний аналіз сучасного стану інформаційних технологій екологічного моніторингу. Встановлено, що існуючі підходи базуються переважно на ретроспективному аналізі даних спостережень, що унеможлиблює попередження кризових явищ. Обґрунтовано необхідність переходу до парадигми проактивного управління, яка вимагає інтеграції прогнозних математичних моделей розсіювання шкідливих домішок в атмосферу (зокрема, регуляторної моделі AERMOD) у контур прийняття рішень. Виявлено, що основною перешкодою для такої інтеграції є висока складність підготовки даних та відсутність автоматизованих засобів семантичної взаємодії між компонентами системи. Обґрунтовано, що вирішення задач проактивного управління можливе шляхом розроблення інтелектуалізованої програмної системи.

2. Розроблено онтологічну модель предметної області Urban Emission Source Ontology (UESO), яка, на відміну від традиційних реляційних схем, забезпечує формальний семантичний опис інфраструктури міста, джерел викидів, рецепторів та метеорологічних умов. Використання UESO дозволило реалізувати концепцію «єдиного джерела істини», забезпечити логічну цілісність даних та автоматизувати процес трансляції знань експерта у синтаксис вхідних файлів фізичних моделей розсіювання забруднення в атмосфері. Розроблено та обґрунтовано гібридну архітектуру збору та інтеграції даних. Дана архітектура використовує онтологію UESO як центральний семантичний хаб, що інкапсулює складність гетерогенних джерел (AERMET, BRIPPRM, кадастр об'єктів теплогенерації) та відокремлює

логіку знань від зберігання артефактів (файлове сховище) та гео-даних (PostGIS). На основі онтології UESO надано формалізоване визначення двох типів проактивного аналізу: оцінювального та оптимізаційного сценаріїв.

3. Розроблено математичну модель вибору локацій розміщення джерел теплогенерації на основі задачі комбінаторної оптимізації. На відміну від класичних геометричних задач розміщення, запропонований підхід враховує суперпозицію полів концентрацій в точках-рецепторах та використовує модуль AERMOD для обчислення цільової функції. Для гарантування дотримання екологічних стандартів цільова функція, що враховує індекс якості повітря, модифікована методом штрафних функцій та дозволяє мінімізувати максимальний ризик перевищення ГДК у точках-рецепторах. Перехід до дискретного простору пошуку дозволив уникнути проблем локальних мінімумів, характерних для неперервних функцій. Також визначення множини допустимих локацій дозволяє враховувати додаткові обмеження в задачі оптимізації: фізичні – доступність земельної ділянки, відстань до точок підключення інженерних мереж (газ, електрика), наявність охоронних зон; економічні – вартість прокладання комунікацій та вартість оренди/купівлі землі.

4. Розроблено метод адаптивної оптимізації розміщення джерел техногенного забруднення в умовах щільної міської забудови. На основі розроблених уніфікованих операторів в середовищі DEAP адаптовано метаевристичні алгоритми (генетичний алгоритм, метод рою часток, алгоритм бджолиної колонії) для вирішення задач дискретної оптимізації фіксованої розмірності. Запропоновано інтелектуалізований вибір стратегії оптимізації, який базується на аналізі складності задачі. Введення порогів складності та масштабу дозволяє системі автоматично перемикатися між повним перебором (для тривіальних задач), D-PSO (для задач тактичного рівня) та D-ABC (для стратегічних задач високої розмірності). Запропонований підхід дозволяє знаходити субоптимальні рішення, при цьому забезпечуючи зниження обчислювальних витрат порівняно зі стандартними алгоритмами, що робить

його придатним для вирішення задач високої розмірності.

5. Запропоновано та обґрунтовано сервісно-орієнтовану архітектуру інтелектуалізованої системи, яка базується на мікросервісному підході та використанні контейнеризації. Ключовою особливістю архітектури є розробка спеціалізованих сервісів-адаптерів (Wrappers), що забезпечило безшовну інтеграцію обчислювального модуля AERMOD у сучасне розподілене Web-середовище. Це дозволило автоматизувати повний цикл обчислень – від препроцесингу метеоданих до генерації карт забруднення. Здійснено програмну реалізацію системи «Есо-City», яка включає підсистеми управління онтологією, сценарного моделювання, оптимізації та геопросторової візуалізації. Розроблено інтерактивний веб-інтерфейс, що підтримує когнітивний цикл роботи експерта, забезпечуючи візуалізацію полів концентрацій та аналіз ризиків для критичних рецепторів. Реалізовано механізми аудиту та логування процесів, що гарантує відтворюваність результатів моделювання.

6. Експериментально підтверджено працездатність та ефективність розробленої системи. Порівняльний аналіз показав, що впровадження інтелектуалізованих засобів «Есо-City» дозволяє скоротити операційний час підготовки та проведення сценарного аналізу в середньому у 24 рази порівняно з використанням стандартного інструментарію AERMOD View. Валідація результатів розрахунків підтвердила їх повну відповідність еталонним значенням сертифікованого програмного забезпечення US EPA. Практична цінність результатів підтверджена апробацією системи на прикладі задачі оптимізації розміщення енергогенеруючих потужностей у мікрорайоні «Сонячний» (м. Тернопіль). Застосування системи дозволило виявити зони підвищеного екологічного ризику та сформувані науково обґрунтовані рекомендації щодо розташування джерел викидів, які мінімізують шкідливий вплив на екосистему міста.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. AERMOD (Atmospheric Dispersion Modeling System). Документація (US EPA): <https://www.epa.gov/scram/air-quality-dispersion-modeling-preferred-and-recommended-models#aermod>
2. Air Quality Dispersion Modeling - Preferred and Recommended Models. <https://www.epa.gov/scram/air-quality-dispersion-modeling-preferred-and-recommended-models>
3. Arya, S. P. (1999). Air micrometeorology and atmospheric. Oxford University Press.
4. Athanasiadis Ioannis, Rizzoli Andrea-Emilio, Donatelli Marcello & Carlini, L. (2006). Enriching software model interfaces using ontology-based tools.
5. Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. Scientific American, 284(5), 34-43.
6. Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Computing Surveys (CSUR), 35(3), 268–308. <https://doi.org/10.1145/937503.937505>
7. Briggs, G. A. (1969). Plume rise. U.S. Atomic Energy Commission, Division of Technical Information.
8. Church, R. L., & Murray, A. T. (2009). Business site selection, location analysis, and GIS. John Wiley & Sons.
9. Cimorelli, A. & Perry, Steve & Venkatram, Akula & Weil, Jeffrey & Paine, Robert & Wilson, Robert & Lee, Russell & Peters, Warren & Brode, Roger. (2005). AERMOD: A Dispersion Model for Industrial Source Applications. Part I: General Model Formulation and Boundary Layer Characterization. Journal of Applied Meteorology – J APPL METEOROL. 44. <https://doi.org/10.1175/JAM2227.1>.
10. Cimorelli, A. J., Perry, S. G., Venkatram, A., Weil, J. C., Paine, R. J., Wilson, R. B., ... & Brode, R. W. (2005). AERMOD: A dispersion model for industrial source applications. Part I: General model formulation and boundary layer

characterization. *Journal of Applied Meteorology*, 44(5), 682–693.
<https://doi.org/10.1175/JAM2227.1>

11. DEAP documentation. <https://deap.readthedocs.io/en/master/index.html>

12. Deb, K. (2000). An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186(2-4), 311–338. [https://doi.org/10.1016/S0045-7825\(99\)00389-8](https://doi.org/10.1016/S0045-7825(99)00389-8)

13. Denzer, R. (2015). Environmental Information Systems. In S. E. Jorgensen (Ed.), *Handbook of Ecological Processes* (pp. 1–21). Springer.

14. Docker. Документація. <https://docs.docker.com/>

15. EEA, 2014a, Air quality in Europe – 2014 report, EEA Report No5/2014, European Environment Agency, Copenhagen, Denmark.

16. Elbeltagi, E., Hegazy, T., & Grierson, D. (2005). Comparison among five evolutionary-based optimization algorithms. *Advanced Engineering Informatics*, 19(1), 43–53. <https://doi.org/10.1016/j.aei.2005.01.004>

17. Eric D. Taillard, Luca Maria Gambardella, Michael Gendreau, and Jean-Yves Potvin. Adaptive memory programming: A unified view of metaheuristics. *European Journal of Operational Research*, 135(1):1–16, November 6, 2001. ISSN: 0377-2217. doi:10.1016/S0377-2217(00)00268-X.

18. European Union emission inventory report. 2023. <https://www.eea.europa.eu/en/analysis/publications/european-union-emissions-inventory-report-1990-2021>

19. Falbo R., Natali A., Mian P., Bertollo G., Ruy F. B. ODE: ontology-based software development environment. *Computer Science*. 2003. P.1124-1135.

20. Farahani, R. Z., SteadieSeifi, M., & Asgari, N. (2010). Multiple facility location problems: A review. *Applied Mathematical Modelling*, 34(7), 1689–1709. <https://doi.org/10.1016/j.apm.2009.10.005>

21. FastAPI. Документація. <https://fastapi.tiangolo.com/>

22. Fazli, H., Farooq, S., Yang, C., & Wæhrens, B. V. (2023). Proactive and Reactive Approaches towards Sustainable Practices in Manufacturing Companies:

Emerging Economies Perspective. Sustainability, 15(17), 12796.
<https://doi.org/10.3390/su151712796>

23. Fielding R. Architectural styles and the design of network-based software architectures. University of California, Irvine. – 2000.

24. Fred Glover and Gary A. Kochenberger, editors. Handbook of Metaheuristics, volume 57 of International Series in Operations Research & Management Science. Kluwer Academic Publishers / Springer, New York, USA, 2003. Series Editor Frederick S. Hillier. doi:10.1007/b101874.

25. Fuller, R., Landrigan, P. J., Balakrishnan, K., Bathan, G., Bose-O'Reilly, S., Brauer, M., ... & Yamashita, C. (2022). Pollution and health: a progress update. The Lancet Planetary Health, 6(6), e535–e547. [https://doi.org/10.1016/S2542-5196\(22\)00090-0](https://doi.org/10.1016/S2542-5196(22)00090-0)

26. Garlan D., Shaw M. An introduction to software architecture. Advances in Software Engineering and Knowledge Engineering. – Hackensack, NJ, USA: World Scientific Publishing Company, 1993. – Vol. 2. – P. 1-39.

27. Georgilakis, P. S., & Hatziargyriou, N. D. (2013). Optimal distributed generation placement in power distribution networks: Models, methods, and future research. IEEE Transactions on Power Systems, 28(3), 3420–3428. <https://doi.org/10.1109/TPWRS.2012.2237043>

28. GeoSPARQL. Специфікація (OGC).
<https://www.ogc.org/standard/geosparql/>

29. Global Optimization Toolbox, [Online]. Available at:
<https://www.mathworks.com/products/global-optimization.html>.

30. Goldberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning. Addison-Wesley.

31. Gorgorió N., Albarracín L., Ärlebäck J., Laine A., Newton R., Villarreal A. Fundamental mathematical knowledge: progressing its specification. Linköping: Linköping University Electronic Press; 2019. (LiTH-MAT-R). URL: <http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-157096>.

32. GraphDB. Документація. <https://graphdb.ontotext.com/documentation/>

33. Grenon P., Smith B. Foundations of an ontology of philosophy. *Synthese*. 2011. Vol. 182. P. 185-204.
34. Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2), 199-220.
35. Guedj, Benjamin, and Bhargav Srinivasa Desikan. 2020. "Kernel-Based Ensemble Learning in Python". *Information*. 11, no. 2: 63. <https://doi.org/10.3390/info11020063>.
36. Hassija, V., Chamola, V., Mahapatra, A. et al. Interpreting Black-Box Models: A Review on Explainable Artificial Intelligence. *Cogn Comput*. 16, 45–74 (2024). <https://doi.org/10.1007/s12559-023-10179-8>.
37. Holmes, N. S., & Morawska, L. (2006). A review of dispersion modelling and its application to the dispersion of particles: An overview of different dispersion models available. *Atmospheric Environment*, 40(30), 5902–5928. <https://doi.org/10.1016/j.atmosenv.2006.06.003>
38. Huang Yi-Chun, Wong Ying-Jiuan, Yang Min-Li. (2014). Proactive environmental management and performance by a controlling family. *Management Research Review*. 37. <https://doi.org/10.1108/MRR-09-2012-0196>.
39. Husáková M., Bureš V. Formal ontologies in information systems development: a systematic review. *Information*. 2020. Vol. 11, no. 2. P. 66.
40. Ibrahim H. Osman. An introduction to metaheuristics. In M. Lawrence and C. Wilsdon, editors, *Operational Research Tutorial Papers*, pages 92–122. Stockton Press, Hampshire, UK, 1995.
41. IT Christou, WL Darrell, K. De Long, and W. Martin, *Evolutionary Algorithms*, SpringerVerlag: New York, NY, USA, 2021.
42. Itziar U., Nieto M., García M., Otaegui O. Design and implementation of an ontology for semantic labeling and testing: automotive global ontology (AGO). *Applied Sciences*. 2021. Vol. 11, no. 17. 7782.
43. Jinzhi L., Ma J., Zheng X., Wang G., Kiritsis D. Design ontology supporting model-based systems-engineering formalisms. *Journal of Latex Class Files*. Vol. 14. No. 8, August 2015.

44. Jonas Sjöberg, Qinghua Zhang, Lennart Ljung, Albert Benveniste, Bernard Delyon, Pierre-Yves Glorennec, Håkan Hjalmarsson, Anatoli Juditsky. Nonlinear black-box modeling in system identification: a unified overview, *Automatica*. Volume 31, Issue 12. 1995. Pages 1691-1724, [https://doi.org/10.1016/0005-1098\(95\)00120-8](https://doi.org/10.1016/0005-1098(95)00120-8).

45. Jorgensen B. The theory of dispersion models. – CRC Press, 1997. – P. 237.

46. Juan S. et al. An Industrial air pollution dispersion system based on Gauss dispersion model. *Environmental Pollution & Control*. 2005. Vol. 7. P.11.

47. Judea Pearl. Heuristics: Intelligent Search Strategies for Computer Problem Solving. The Addison-Wesley series in artificial intelligence. Addison-Wesley Pub (Sd), April 1984.

48. Karaboga D. An idea based on honey bee swarm for numerical optimization: Techn. rep. – TR06. – Erciyes: Erciyes Univ. Press, 2005. – 10 p.

49. Karaboga D., Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization*. – 2007. – Vol. 39. – P. 459-471.

50. Karaboga, D., & Akay, B. (2009). A comparative study of Artificial Bee Colony algorithm. *Applied Mathematics and Computation*, 214(1), 108–132. <https://doi.org/10.1016/j.amc.2009.03.090>

51. Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39(3), 459-471. <https://doi.org/10.1007/s10898-007-9149-x>

52. Katoch S., Chauhan SS., Kumar V. A review on genetic algorithms: past, present, and future. *Multimed Tools Appl*. 2021. Vol. 80, P.8091–8126.

53. Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks* (Vol. 4, pp. 1942–1948). IEEE.

54. Kennedy, J., & Eberhart, R. C. (1997). A discrete binary version of the particle swarm algorithm. In 1997 IEEE International Conference on Systems, Man, and Cybernetics (Vol. 5, pp. 4104–4108). IEEE.

55. Kubernetes. Документація. <https://kubernetes.io/docs/home/>

56. Kumar, P., Morawska, L., Martani, C., Biskos, G., Neophytou, M., Di Sabatino, S., ... & Britter, R. (2015). The rise of low-cost sensing for managing air pollution in cities. *Environment International*, 75, 199–205. <https://doi.org/10.1016/j.envint.2014.11.019>

57. Lakes Environmental Software. (2023). AERMOD View: Gaussian Plume Air Dispersion Model. <https://www.weblakes.com/products/aermod-view/>

58. Lakes Environmental Software. (2023). SCREEN View: Screening Air Dispersion Model. Retrieved from <https://www.weblakes.com/products/screen-view/>

59. Lange C. Ontologies and languages for representing mathematical knowledge on the Semantic Web. *Semant. Web* 4. 2013. Vol. 2, P. 119–158.

60. Lange C., Kohlhase M. A Semantic Wiki for mathematical knowledge management. *Emerging Technologies for Semantic Work Environments: Techniques, Methods, and Applications*. Edited by Rech, Jörg and Björn Decker, and Eric Ras. Hershey, PA: IGI Global. 2008. P. 47-68.

61. Lanzafame, R., Monforte, P., Patanè, G., & Strano, S. (2022). Optimization of chimney height for thermal power plants using genetic algorithms and dispersion models. *Energy Conversion and Management*, 256, 115388. <https://doi.org/10.1016/j.enconman.2022.115388>

62. Lateb, M., Meroney, R. N., Yataghene, M., Fellouah, H., Saleh, F., & Boufadel, M. C. (2013). On the use of numerical modelling for near-field pollutant dispersion in urban environments – A review. *Environmental Pollution*, 174, 116–128. <https://doi.org/10.1016/j.envpol.2012.11.004>

63. Leaflet. Документація. <https://leafletjs.com/reference.html>

64. Lister, Nina-Marie, 2016. Resilience beyond Rhetoric in Urban Landscape Planning and Design. In: George F. Thompson, Frederick R. Steiner and Armando

Carbonell (eds) *Nature and Cities: The Ecological Imperative in Urban Design and Planning*. Cambridge, MA: Lincoln Institute of Land Policy.

65. Mattoso, M., Werner, C., Travassos, G. H., Ogasawara, E., & de Oliveira, D. (2010). Towards a new generation of scientific workflow management systems. *Concurrency and Computation – Practice and Experience*, 22(9), 1147-1159.

66. McIntosh, B. S., Ascough II, J. C., Twery, M., Chew, J., Elmahdi, A., Haase, D., ... & Voinov, A. (2011). Environmental decision support systems (EDSS) development—Challenges and best practices. *Environmental Modelling & Software*, 26(12), 1389–1402. <https://doi.org/10.1016/j.envsoft.2011.09.009>

67. Mealpy. Документація. <https://mealpy.readthedocs.io/en/latest/>

68. Mehleri, E. D., Sarimveis, H., Markatos, N. C., & Papageorgiou, L. G. (2012). Optimal design and operation of distributed energy systems: Application to Greek residential sector. *Renewable Energy*, 51, 331–342. <https://doi.org/10.1016/j.renene.2012.09.009>

69. Mohan, Manju & Bhati, Shweta & Sreenivas, Archana & Marrapu, Pallavi. (2011). Performance Evaluation of AERMOD and ADMS-Urban for Total Suspended Particulate Matter Concentrations in Megacity Delhi. *Aerosol and Air Quality Research*. 11. 883–894. <https://doi.org/10.4209/aaqr.2011.05.0065>.

70. Mohan, Manju & Bhati, Shweta & Sreenivas, Archana & Marrapu, Pallavi. (2011). Performance Evaluation of AERMOD and ADMS-Urban for Total Suspended Particulate Matter Concentrations in Megacity Delhi. *Aerosol and Air Quality Research*. 11. 883–894. <https://doi.org/10.4209/aaqr.2011.05.0065>.

71. Pan, Q. K., Tasgetiren, M. F., Suganthan, P. N., & Chua, T. J. (2011). A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. *Information Sciences*, 181(12), 2455–2468. <https://doi.org/10.1016/j.ins.2011.01.042>

72. Pasquill, F. (1961). The estimation of the dispersion of windborne material. *Meteorological Magazine*, 90(1063), 33-49.

73. Petersen, R. L., Guerra, S. A., & Bova, A. S. (2017). Critical review of the building downwash algorithms in AERMOD. *Journal of the Air & Waste*

Management Association (1995), 67(8), 826–835.
<https://doi.org/10.1080/10962247.2017.1279088>

74. Petersen, Ron & Beyer-Lout, Anke. (2012). AERMOD building downwash theoretical limitations and possible solutions.
<https://doi.org/10.13140/RG.2.1.1642.3768>.

75. Pirhalla, M., Heist, D., Perry, S., Tang, W., & Brouwer, L. (2021). Simulations of dispersion through an irregular urban building array. Atmospheric environment (Oxford, England: 1994), 258,
<https://doi.org/10.1016/j.atmosenv.2021.118500>

76. PlantUML. Документація. <https://plantuml.com/>

77. PostGIS. Документація. <https://postgis.net/documentation/>

78. PostgreSQL. Документація. <https://www.postgresql.org/docs/>

79. Providence ORIS. (2023). BEEST Suite for AERMOD. Providence Engineering and Environmental Group LLC.
<https://www.providenceoris.com/beest/>

80. PySwarms. Документація. <https://pyswarms.readthedocs.io/en/latest/>

81. R. Raskin and M. Pan, “Semantic web for earth and environmental terminology (sweet)”, in Proc. of the Workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data, vol. 25, 2003.

82. Raskin, Robert & Pan, Michael. (2005). Knowledge representation in the Semantic Web for Earth and Environmental Terminology (SWEET). Computers & Geosciences. 31. 1119-1125. <https://doi.org/10.1016/j.cageo.2004.12.004>.

83. React. Документація. <https://react.dev/>

84. Reed, Chris and Nina-Marie Lister, eds. 2014. Projective Ecologies. New York: Harvard GSD and Actar.

85. Rood, A. S. (2014). Performance evaluation of AERMOD, CALPUFF, and ISCST3 for stratospheric dispersion using high-resolution meteorological data. Atmospheric Environment, 89, 443–450.
<https://doi.org/10.1016/j.atmosenv.2014.02.054>

86. Seinfeld, J. H., & Pandis, S. N. (2016). Atmospheric chemistry and physics: From air pollution to climate change (3rd ed.). John Wiley & Sons.
87. SPARQL 1.1. Специфікація (W3C). <https://www.w3.org/TR/sparql11-query/>
88. Stoyan, Y. G., & Chugay, A. M. (2009). Mathematical modeling of the layout problems of technological equipment. Cybernetics and Systems Analysis, 45(2), 202–211.
89. Suresh P., Hsu S.-H., Akkisetty P., Reklaitis G., Venkatasubramanian V. OntoMODEL: ontological mathematical modeling knowledge management in pharmaceutical product development, 1: Conceptual framework. Industrial & Engineering Chemistry Research. 2010. Vol. 49. P. 7768-7781.
90. Suresh P., Joglekar G., Hsu SH., Akkisetty P., Hailemariam L., Jain A., Reklaitis G., Venkatasubramanian V. Onto MODEL: Ontological mathematical modeling knowledge management. Computer-aided chemical engineering. 2008. Vol. 25. P. 985-990.
91. Suresh S.S, Naidu M.M., Asha Kiran. S. An XML Based Knowledge-Driven Decision Support System for Design Pattern Selection. International Journal of Research in Engineering and Technology (IJRET), 2012. Vol. 1(3) P. 1-13.
92. SWEET Ontologies, 2006. Semantic Web for Earth and Environmental Terminology (SWEET), url: <http://sweet.jpl.nasa.gov>, Last updated on: Jan 26, 2006.
93. SWEET Ontology. Репозиторій. <https://github.com/ESIPFed/sweet>
94. Talbi, E. G. (2009). Metaheuristics: from design to implementation. John Wiley & Sons.
95. Teofilo F. Gonzalez, editor. Handbook of Approximation Algorithms and Metaheuristics. Chapman & Hall/CRC Computer and Information Science Series. Chapman & Hall/CRC Press (Taylor & Francis Group), 2007.
96. Trinity Consultants. (2022). BREEZE AERMOD/ISC User's Guide. Dallas, Texas: Trinity Consultants.

97. Trinity Consultants. (2023). BREEZE AERMOD: Air Quality Modeling Software. Retrieved from <https://www.breeze-software.com/software/AERMOD>
98. Trokanas N., Cecelja F. Ontology evaluation for reuse in the domain of Process Systems Engineering. *Comput. Chem. Eng.* 2016. Vol. 85. P. 177-187.
99. Tu SW., Eriksson H., Gennari JH., Shahar Y., Musen MA. Ontology-based configuration of problem-solving methods and generation of knowledge-acquisition tools: application of PROTEGE-II to protocol-based decision support. *Artif. Intell. Med.* 1995. Vol. 7(3). P. 257-289.
100. Turner, D. B. (1994). *Workbook of atmospheric dispersion estimates: An introduction to dispersion modeling* (2nd ed.). Lewis Publishers.
101. Tymchyshyn V., Melnyk A., Tymchyshyn B., Faifura V., Mazur I.-S., Honchar L. Data Management Service Architecture of the Software for Modeling Harmful Emissions in Soil. *Proceedings International Conference on Advanced Computer Information Technologies Acit.* 2024, P. 678–682. DOI: <https://doi.org/10.1109/ACIT62333.2024.10712526/>
102. Tymchyshyn V., Porplytsya N., Melnyk A., Tymchyshyn B. Software for Modelling the Air Pollution by Vehicles. *CEUR-WS: Proceedings of the International Conference Advanced Computer Information Technologies (ACIT 2018)*. 2018. Vol 2300. pp. 207-210;
103. Tymchyshyn V., Tymchyshyn B., Melnyk A., Manzhula V., Faifura V., Romanets I. The System Architecture of the Software for Modeling Harmful Emissions in Soil. *Proceedings International Conference on Advanced Computer Information Technologies Acit.* 2023, P. 58–62. DOI: <https://doi.org/10.1109/ACIT58437.2023.10275416/>
104. U.S. Environmental Protection Agency. (2004). *AERMOD: Description of Model Formulation* (EPA-454/R-03-004). Office of Air Quality Planning and Standards. <https://www.epa.gov/scram/air-quality-dispersion-modeling-preferred-and-recommended-models>
105. U.S. Environmental Protection Agency. (2017). *Guideline on Air Quality Models* (40 CFR Part 51, Appendix W). Federal Register.

106. U.S. Environmental Protection Agency. (2017). Guideline on air quality models: 40 CFR Part 51, Appendix W. Federal Register.
107. U.S. Environmental Protection Agency. (2021). User's Guide for the AERMOD Modeling System. EPA-454/B-21-002.
108. Vardoulakis, S., Fisher, B. E., Pericleous, K., & Gonzalez-Flesca, N. (2003). Modelling air quality in street canyons: A review. *Atmospheric Environment*, 37(2), 155–182. [https://doi.org/10.1016/S1352-2310\(02\)00857-9](https://doi.org/10.1016/S1352-2310(02)00857-9)
109. Waddell, P. (2002). UrbanSim – Modeling urban development for land use, transportation, and environmental planning. *Journal of the American Planning Association*, 68(3), 297-314.
110. Weber, C., & Shah, N. (2011). Optimization of district energy systems design and operation. *Computers & Chemical Engineering*, 35(6), 1214–1237. <https://doi.org/10.1016/j.compchemeng.2011.01.030>
111. Zanella, A., Bui, N., Castellani, A., Vangelista, L., & Zorzi, M. (2014). Internet of things for smart cities. *IEEE Internet of Things Journal*, 1(1), 22–32. <https://doi.org/10.1109/JIOT.2014.2306328>
112. Zannetti, P. (2013). Air pollution modeling: theories, computational methods and available software. Springer Science & Business Media.
113. Zannetti, P. (Ed.). (1990). Air pollution modeling: Theories, computational methods and available software. Van Nostrand Reinhold.
114. Zou, B., Zhan, F. B., Wilson, J. G., & Zeng, Y. (2010). Performance of AERMOD at different time scales. *Simulation Modelling Practice and Theory*, 18(5), 612–623. <https://doi.org/10.1016/j.simpat.2010.01.005>
115. В.І. Манжула, Б.С. Тимчишин. Семантичне моделювання гетерогенних джерел шкідливих викидів в атмосферу для проактивного управління екосистемою міста. Наукові праці Донецького національного технічного університету. Серія: “Інформатика, кібернетика та обчислювальна техніка”. 2025. №2 (41), с. 118–127. <https://doi.org/10.31474/1996-1588-2025-2-41-118-127>

116. Глобальні рекомендації ВООЗ щодо якості повітря, 2021.
<https://iris.who.int/server/api/core/bitstreams/2776fed3-acf3-43a2-b69e-f641e42a7bd8/content>
117. Гранберг, І. Г., & Савін, А. А. (2009). Математичне моделювання перенесення та розсіювання домішок в атмосфері. Наукова думка.
118. Губернський, О. М. (Ред.). (2005). Гігієна планування та забудови населених місць: Нормативно-правові акти. Медицина.
119. Дідковський, М. М., & Бондарчук, І. В. (2015). Методологічні аспекти оцінки впливу викидів промислових підприємств на якість атмосферного повітря. Екологічна безпека та природокористування, (3), 45-52.
120. Мельник А.М. Архітектура програмного забезпечення для математичного моделювання на основі інтервального та онтологічного підходу. Вісник Хмельницького національного університету. Технічні науки. 2022. №3. С. 265–273.
121. Мельник А.М. Онтологія як програмна надбудова до системи для математичного моделювання на основі інтервальних даних. Інформаційні технології та комп'ютерна інженерія. 2022. Том. 54. №2. С. 26–38.
122. Мельник А.М., Дивак М.П., Манжула В.І. Моделювання динаміки концентрацій шкідливих викидів автотранспорту на основі поєднання методів інтервального аналізу та онтологічного підходу. Вимірювальна та обчислювальна техніка в технологічних процесах. 2022 №. 2. С. 14-23.
<https://doi.org/10.31891/2219-9365-2022-70-2-2>
123. НВП "Логіка". (2021). Програма розрахунку забруднення атмосфери "АТМОСфера".
124. Тимчишин Б., Манжула В. Математичне та алгоритмічне забезпечення інтелектуалізованої програмної системи для проактивного управління екосистемою міста. Вимірювальна та обчислювальна техніка в технологічних процесах. 2025. вип. 84, вип. 4, с. 35–48.
<https://doi.org/10.31891/2219-9365-2025-840-4>

125. Тимчишин В., Мельник А., Дивак Т., Тимчишин Б., Файфура В., Момотюк Є., Костик Б. Інтегральний показник для оцінювання ефективності спеціалізованих моніторингових систем. Оптико-електронні інформаційно-енергетичні технології. 2024. 48(2). с. 234–241. DOI: <https://doi.org/10.31649/1681-7893-2024-48-2-234-241>
126. Тимчишин В.С., Порплиця Н.П., Тимчишин Б.С. Програмний комплекс для моделювання забруднення атмосфери шкідливими викидами автотранспорту в часі. Сучасні комп'ютерні інформаційні технології: Матеріали Всеукраїнської конференції з міжнародною участю АСІТ'2017. Тернопіль: ТНЕУ, 2017. с.233-234.
127. ТОВ "Софт-Проект". (2022). Програмний комплекс "ЕОЛ+" [Версія 5.4]. <http://eolplus.com.ua/>
128. Чуб І.А, Новожилова М.В., Андронов В.А. Моделювання прикладних оптимізаційних задач розміщення об'єктів з метричними характеристиками, що змінюються: монографія / Чуб І.А., Новожилова М.В., Андронов В.А. – Харків: НУЦЗ України, 2017. – 167 с.

ДОДАТОК А

Фрагмент лістингу коду онтології UESO v2.0 у форматі Turtle

@prefix : <http://www.eco-city.info/ontology#> .

@prefix owl: <http://www.w3.org/2002/07/owl#> .

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

@prefix xml: <http://www.w3.org/XML/1998/namespace> .

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

@prefix ueso: <http://www.eco-city.info/ontology#> .

@prefix geo: <http://www.opengis.net/ont/geosparql#> .

@prefix sweet: <http://sweetontology.net/prop/> .

#####

Ontology Header & Imports

#####

<http://www.eco-city.info/ontology> rdf:type owl:Ontology ;

owl:imports <http://www.opengis.net/ont/geosparql> ;

owl:imports <http://sweetontology.net/prop> ;

rdfs:comment "Urban Emission Source Ontology (UESO v2.0). Designed for proactive scenario analysis and optimization using AERMOD."@en .

#####

Object Properties (Зв'язки між сутностями)

#####

--- 1. Властивості Джерел ---

ueso:hasStack rdf:type owl:ObjectProperty ;

rdfs:domain ueso:PointSource ;

rdfs:range ueso:Stack ;

rdfs:comment "Зв'язує точкове джерело з його фізичною трубою"@uk .

```

ueso:hasEmission rdf:type owl:ObjectProperty ;
  rdfs:domain ueso:EmissionSource ;
  rdfs:range ueso:Emission ;
  rdfs:comment "Визначає викид конкретної речовини"@uk .

```

```

ueso:emitsPollutant rdf:type owl:ObjectProperty ;
  rdfs:domain ueso:Emission ;
  rdfs:range ueso:Pollutant .

```

```

ueso:usesFuel rdf:type owl:ObjectProperty ;
  rdfs:domain ueso:HeatGenerationSource ;
  rdfs:range ueso:FuelType .

```

--- 2. Властивості Сценаріїв (Загальні та Оцінювальні) ---

```

ueso:isModificationOf rdf:type owl:ObjectProperty ;
  rdfs:domain ueso:Scenario ;
  rdfs:range ueso:Scenario ;
  rdfs:comment "Вказує на батьківський (базовий) сценарій для розрахунку дельти"@uk .

```

```

ueso:implementsPolicy rdf:type owl:ObjectProperty ;
  rdfs:domain ueso:Scenario ;
  rdfs:range ueso:Policy .

```

```

ueso:includesSource rdf:type owl:ObjectProperty ;
  rdfs:domain ueso:EvaluativeScenario ;
  rdfs:range ueso:EmissionSource ;
  rdfs:comment "Визначає множину джерел I_s для розрахунку"@uk .

```

```

ueso:includesReceptorGroup rdf:type owl:ObjectProperty ;
  rdfs:domain ueso:EvaluativeScenario ;
  rdfs:range ueso:ReceptorGroup ;

```

rdfs:comment "Визначає множину рецепторів R_s"@uk .

ueso:usesMeteoData rdf:type owl:ObjectProperty ;

rdfs:domain ueso:Scenario ;

rdfs:range ueso:MeteoArtifact ;

rdfs:comment "Посилання на метеорологічний артефакт M_s"@uk .

ueso:hasResult rdf:type owl:ObjectProperty ;

rdfs:domain ueso:EvaluativeScenario ;

rdfs:range ueso:ScenarioResult .

--- 3. Властивості Оптимізації (Optimization Scenario) ---

ueso:usesBaseSources rdf:type owl:ObjectProperty ;

rdfs:domain ueso:OptimizationScenario ;

rdfs:range ueso:EmissionSource ;

rdfs:comment "Фонові джерела I_base"@uk .

ueso:usesVirtualTemplate rdf:type owl:ObjectProperty ;

rdfs:domain ueso:OptimizationScenario ;

rdfs:range ueso:EmissionSource ;

rdfs:comment "Шаблон нового джерела I_virt"@uk .

ueso:hasDecisionSpace rdf:type owl:ObjectProperty ;

rdfs:domain ueso:OptimizationScenario ;

rdfs:range ueso:AvailableSite ;

rdfs:comment "Простір пошуку D_locations (доступні локації)"@uk .

ueso:hasTargetReceptors rdf:type owl:ObjectProperty ;

rdfs:domain ueso:OptimizationScenario ;

rdfs:range ueso:ReceptorGroup ;

rdfs:comment "Цільові рецептори R_target для f_obj"@uk .

```

ueso:usesOptimizationMethod rdf:type owl:ObjectProperty ;
    rdfs:domain ueso:OptimizationScenario ;
    rdfs:range ueso:OptimizationMethod ;
    rdfs:comment "Обраний алгоритм O_method (напр., GA, ABC)"@uk .

```

```

ueso:hasOptimalSolution rdf:type owl:ObjectProperty ;
    rdfs:domain ueso:OptimizationScenario ;
    rdfs:range ueso:EvaluativeScenario ;
    rdfs:comment "Посилання на знайдений оптимальний сценарій"@uk .

```

--- 4. Властивості Інфраструктури ---

```

ueso:containsReceptor rdf:type owl:ObjectProperty ;
    rdfs:domain ueso:ReceptorGroup ;
    rdfs:range ueso:ReceptorPoint .

```

#####

Data Properties (Атрибути даних)

#####

--- Параметри AERMOD (успадковані від SWEET) ---

```

ueso:hasStackHeight rdf:type owl:DatatypeProperty ;
    rdfs:subPropertyOf sweet:PhysicalProperty ;
    rdfs:domain ueso:PointSource ;
    rdfs:range xsd:decimal .

```

```

ueso:hasStackDiameter rdf:type owl:DatatypeProperty ;
    rdfs:subPropertyOf sweet:PhysicalProperty ;
    rdfs:domain ueso:PointSource ;
    rdfs:range xsd:decimal .

```

```

ueso:hasExitVelocity rdf:type owl:DatatypeProperty ;
    rdfs:subPropertyOf sweet:PhysicalProperty ;

```

```
rdfs:domain ueso:PointSource ;
rdfs:range xsd:decimal .
```

```
ueso:hasExitTemp rdf:type owl:DatatypeProperty ;
rdfs:subPropertyOf sweet:PhysicalProperty ;
rdfs:domain ueso:PointSource ;
rdfs:range xsd:decimal .
```

```
ueso:hasReleaseHeight rdf:type owl:DatatypeProperty ;
rdfs:domain [ owl:unionOf ( ueso:AreaSource ueso:VolumeSource ) ] ;
rdfs:range xsd:decimal .
```

```
ueso:hasInitialLateralDimension rdf:type owl:DatatypeProperty ;
rdfs:domain ueso:VolumeSource ;
rdfs:range xsd:decimal .
```

```
ueso:hasThermalCapacity rdf:type owl:DatatypeProperty ;
rdfs:domain ueso:HeatGenerationSource ;
rdfs:range xsd:decimal .
```

```
ueso:hasAddress rdf:type owl:DatatypeProperty ;
rdfs:domain ueso:EmissionSource ;
rdfs:range xsd:string .
```

--- Параметри Викидів ---

```
ueso:hasEmissionRate rdf:type owl:DatatypeProperty ;
rdfs:domain ueso:Emission ;
rdfs:range xsd:decimal . # г/с
```

--- Параметри Сценаріїв ---

```
ueso:hasStatus rdf:type owl:DatatypeProperty ;
rdfs:domain ueso:Scenario ;
```

```
rdfs:range xsd:string . # ENUM: "Draft", "Validated", "Calculated"
```

```
ueso:aermodConfig rdf:type owl:DatatypeProperty ;
```

```
rdfs:domain ueso:Scenario ;
```

```
rdfs:range xsd:string . # JSON-рядок конфігурації C_s
```

```
### --- Шляхи до файлів (Артефакти) ---
```

```
ueso:hasSfcFilePath rdf:type owl:DatatypeProperty ;
```

```
rdfs:domain ueso:MeteoArtifact ;
```

```
rdfs:range xsd:anyURI .
```

```
ueso:hasPflFilePath rdf:type owl:DatatypeProperty ;
```

```
rdfs:domain ueso:MeteoArtifact ;
```

```
rdfs:range xsd:anyURI .
```

```
### --- Результати ---
```

```
ueso:calculatedScore rdf:type owl:DatatypeProperty ;
```

```
rdfs:domain ueso:ScenarioResult ;
```

```
rdfs:range xsd:float . # Значення f_min
```

```
ueso:hasPlotFilePath rdf:type owl:DatatypeProperty ;
```

```
rdfs:domain ueso:ScenarioResult ;
```

```
rdfs:range xsd:anyURI .
```

```
#####
```

```
# Classes (TBox - Ієрархія класів)
```

```
#####
```

```
### --- 1. Джерела Викидів ---
```

```
ueso:EmissionSource rdf:type owl:Class ;
```

```
rdfs:subClassOf geo:Feature ;
```

```
rdfs:comment "Базовий клас для всіх джерел"@uk .
```

```
ueso:PointSource rdf:type owl:Class ;
    rdfs:subClassOf ueso:EmissionSource .
```

```
ueso:AreaSource rdf:type owl:Class ;
    rdfs:subClassOf ueso:EmissionSource .
```

```
ueso:VolumeSource rdf:type owl:Class ;
    rdfs:subClassOf ueso:EmissionSource .
```

```
ueso:HeatGenerationSource rdf:type owl:Class ;
    rdfs:subClassOf ueso:PointSource .
```

```
ueso:BoilerHouse rdf:type owl:Class ;
    rdfs:subClassOf ueso:HeatGenerationSource .
```

```
ueso:CogenerationUnit rdf:type owl:Class ;
    rdfs:subClassOf ueso:HeatGenerationSource .
```

--- 2. Інфраструктура ---

```
ueso:Stack rdf:type owl:Class .
ueso:Pollutant rdf:type owl:Class .
ueso:Emission rdf:type owl:Class .
ueso:FuelType rdf:type owl:Class .
```

```
ueso:EmissionParameter rdf:type owl:Class ;
    rdfs:subClassOf sweet:PhysicalProperty .
```

```
ueso:ReceptorPoint rdf:type owl:Class ;
    rdfs:subClassOf geo:Feature .
```

```
ueso:ReceptorGroup rdf:type owl:Class ;
    rdfs:comment "Група реценторів для R_target"@uk .
```

```
ueso:AvailableSite rdf:type owl:Class ;
    rdfs:subClassOf geo:Feature ;
```

rdfs:comment "Доступна локація (елемент D_locations)"@uk .

--- 3. Управління (Сценарії) ---

ueso:Policy rdf:type owl:Class .

ueso:Scenario rdf:type owl:Class ;

rdfs:comment "Абстрактний батьківський клас"@uk .

ueso:EvaluativeScenario rdf:type owl:Class ;

rdfs:subClassOf ueso:Scenario ;

rdfs:comment "Оцінювальний сценарій (s_eval)"@uk .

ueso:OptimizationScenario rdf:type owl:Class ;

rdfs:subClassOf ueso:Scenario ;

rdfs:comment "Оптимізаційний сценарій (s_opt)"@uk .

ueso:ScenarioResult rdf:type owl:Class .

--- 4. Обчислювальні компоненти ---

ueso:DataArtifact rdf:type owl:Class .

ueso:MeteoArtifact rdf:type owl:Class ;

rdfs:subClassOf ueso:DataArtifact .

ueso:TerrainArtifact rdf:type owl:Class ;

rdfs:subClassOf ueso:DataArtifact .

ueso:OptimizationMethod rdf:type owl:Class .

ueso:ExhaustiveSearch rdf:type owl:Class ;

rdfs:subClassOf ueso:OptimizationMethod .

ueso:Metaheuristic rdf:type owl:Class ;

rdfs:subClassOf ueso:OptimizationMethod .

ueso:GA rdf:type owl:Class ;

rdfs:subClassOf ueso:Metaheuristic .

```

ueso:DiscreteABC rdf:type owl:Class ;
    rdfs:subClassOf ueso:Metaheuristic .
ueso:DiscretePSO rdf:type owl:Class ;
    rdfs:subClassOf ueso:Metaheuristic .

```

```

#####
#  Individuals (ABox - Фрагмент наповнення)
#####

```

```

### --- Джерела ---

```

```

ueso:Boiler_LesiUkrainky4a rdf:type owl:NamedIndividual , ueso:BoilerHouse ;
    rdfs:label "Котельня на вул. Лесі Українки, 4а"@uk ;
    ueso:hasStackHeight "30.0"^^xsd:decimal ;
    geo:hasGeometry "POINT(25.5845 49.5448)"^^geo:wktLiteral .

ueso:Cogen_Promyslova28 rdf:type owl:NamedIndividual , ueso:CogenerationUnit ;
    rdfs:label "КГУ на вул. Промислова, 28"@uk .

```

```

### --- Рецептори ---

```

```

ueso:Receptor_Hospital rdf:type owl:NamedIndividual , ueso:ReceptorPoint ;
    rdfs:label "Обласна лікарня"@uk ;
    geo:hasGeometry "POINT(25.5900 49.5480)"^^geo:wktLiteral .

```

```

ueso:Group_Sensitive rdf:type owl:NamedIndividual , ueso:ReceptorGroup ;
    rdfs:label "Чутливі об'єкти"@uk ;
    ueso:containsReceptor ueso:Receptor_Hospital .

```

```

### --- Локації для оптимізації ---

```

```

ueso:Site_Promzona_1 rdf:type owl:NamedIndividual , ueso:AvailableSite ;
    geo:hasGeometry "POINT(25.6150 49.5550)"^^geo:wktLiteral .

```

```

### --- Сценарії ---

```

1. Базовий сценарій

```

ueso:Scenario_Base_2025 rdf:type owl:NamedIndividual , ueso:EvaluativeScenario ;
  rdfs:label "Базовий стан (2025)"@uk ;
  ueso:isModificationOf rdf:nil ; # Корінь
  ueso:hasStatus "Completed" ;
  ueso:includesSource ueso:Boiler_LesiUkrainky4a ;
  ueso:includesReceptorGroup ueso:Group_Sensitive .

```

2. Оптимізаційна задача

```

ueso:OptJob_001 rdf:type owl:NamedIndividual , ueso:OptimizationScenario ;
  rdfs:label "Пошук 1 КГУ в районі 'Східний'"@uk ;
  ueso:usesBaseSources ueso:Boiler_LesiUkrainky4a ;
  ueso:usesVirtualTemplate ueso:Template_KGU_5MW ;
  ueso:hasDecisionSpace ueso:Site_Promzona_1 ;
  ueso:hasTargetReceptors ueso:Group_Sensitive ;
  ueso:usesOptimizationMethod ueso:Method_Exhaustive . # k=1 -> Exhaustive

```

ДОДАТОК Б

Фрагмент лістингу коду сервісу обгортки для взаємодії з модулем AERMOD

```

import os
import subprocess
import shutil
from typing import List, Dict, Tuple

class EcoHeatService:
    """
    Сервіс-обгортка для взаємодії з обчислювальним ядром AERMOD.
    Виконує генерацію вхідних файлів, запуск процесу та парсинг результатів.
    """

    def __init__(self, aermod_executable_path: str, working_directory: str):
        """
        Ініціалізація сервісу.
        :param aermod_executable_path: Повний шлях до файлу aermod.exe
        :param working_directory: Папка для тимчасових файлів розрахунку
        """
        self.aermod_exe = aermod_executable_path
        self.work_dir = working_directory

        # Перевірка наявності exe-файлу
        if not os.path.exists(self.aermod_exe):
            raise FileNotFoundError(f"AERMOD executable not found at: {self.aermod_exe}")

        # Створення робочої директорії, якщо не існує
        os.makedirs(self.work_dir, exist_ok=True)

    def run_simulation(self, scenario_data: Dict) -> Dict:

```

```
"""
```

Запускає повний цикл моделювання для одного сценарію.

:param scenario_data: Словник з параметрами сценарію (джерела, рецептори, метео)

:return: Словник з результатами (концентрації в точках рецепторів)

```
"""
```

```
run_id = scenario_data.get('id', 'temp_run')
```

```
current_run_dir = os.path.join(self.work_dir, run_id)
```

```
try:
```

```
    # 1. Підготовка робочого простору
```

```
    self._prepare_workspace(current_run_dir, scenario_data['meteo_files'])
```

```
    # 2. Генерація керуючого файлу AERMOD.INP (Функція F(s))
```

```
    inp_file_path = os.path.join(current_run_dir, 'aermod.inp')
```

```
    self._generate_input_file(inp_file_path, scenario_data)
```

```
    # 3. Запуск обчислювального ядра
```

```
    self._execute_aermod(current_run_dir, inp_file_path)
```

```
    # 4. Парсинг результатів (читання .PLT або .OUT)
```

```
    out_file_path = os.path.join(current_run_dir, 'aermod.plt')
```

```
    results = self._parse_results(out_file_path)
```

```
    return {
```

```
        'status': 'success',
```

```
        'run_id': run_id,
```

```
        'results': results
```

```
    }
```

```
except Exception as e:
```

```
    return {
```

```
        'status': 'error',
```

```

        'run_id': run_id,
        'message': str(e)
    }
finally:
    # Очищення (опціонально)
    # shutil.rmtree(current_run_dir, ignore_errors=True)
    pass

def _prepare_workspace(self, run_dir: str, meteo_files: Dict):
    """Створює папку запуску та копіює туди необхідні метеофайли (.SFC, .PFL)."""
    os.makedirs(run_dir, exist_ok=True)
    # Копіювання .SFC та .PFL файлів у run_dir
    shutil.copy(meteo_files['sfc'], os.path.join(run_dir, 'meteo.sfc'))
    shutil.copy(meteo_files['pfl'], os.path.join(run_dir, 'meteo.pfl'))

def _generate_input_file(self, file_path: str, data: Dict):
    """
    Генерує текстовий файл AERMOD.INP на основі даних сценарію.
    Це реалізація функції-генератора F(s).
    """
    with open(file_path, 'w') as f:
        # --- CO - Control Pathway ---
        f.write("CO STARTING\n")
        f.write(f"  TITLEONE {data.get('title', 'EcoHeat Run')}\n")
        f.write("  MODELOPT  DFAULT CONC\n")
        f.write("  AVERTIME  1 24 PERIOD\n") # Періоди усереднення
        f.write("  POLLUTID NO2\n")
        f.write("  RUNORNOT  RUN\n")
        f.write("CO FINISHED\n\n")

        # --- SO - Source Pathway ---
        f.write("SO STARTING\n")

```

```

for src in data['sources']:

    # Визначення джерела: LOCATION ID TYPE X Y Z
    f.write(f"  LOCATION {src['id']} POINT {src['x']} {src['y']} {src['z']}\n")

    # Параметри: SRCPARAM ID EmisRate Ht Temp Vel Diam
    f.write(f"  SRCPARAM {src['id']} {src['emission_rate']} {src['height']} {src['temp']}
{src['velocity']} {src['diameter']}\n")

    f.write("SO FINISHED\n\n")

# --- RE - Receptor Pathway ---
f.write("RE STARTING\n")
for rec in data['receptors']:

    # DISCCART X Y Z
    f.write(f"  DISCCART {rec['x']} {rec['y']} {rec['z']}\n")

f.write("RE FINISHED\n\n")

# --- ME - Meteorology Pathway ---
f.write("ME STARTING\n")
f.write("  SURFFILE meteo.sfc\n")
f.write("  PROFFILE meteo.pfl\n")
f.write("  SURFDATA 12345 2024\n") # ID станції та рік
f.write("  UAIRDATA 12345 2024\n")
f.write("ME FINISHED\n\n")

# --- OU - Output Pathway ---
f.write("OU STARTING\n")
f.write("  RECTABLE ALLAVE FIRST\n")
# Збереження результатів у PLOTFILE (зручно парсити)
f.write("  PLOTFILE 1 ALL FIRST aermod.plt\n")
f.write("OU FINISHED\n")

def _execute_aermod(self, run_dir: str, inp_path: str):
    """Запускає aermod.exe як підпроцес."""

```

```

try:
    # AERMOD очікує, що ім'я вхідного файлу передається аргументом
    # або він читає 'aermod.inp' з поточної директорії
    process = subprocess.run(
        [self.aermod_exe, inp_path],
        cwd=run_dir, # Важливо: виконувати в папці запуску
        capture_output=True,
        text=True,
        check=True # кине виключення, якщо aermod впаде
    )
    print(f"AERMOD STDOUT: {process.stdout}")
except subprocess.CalledProcessError as e:
    raise RuntimeError(f"AERMOD execution failed: {e.stderr}")

def _parse_results(self, plt_path: str) -> List[Dict]:
    """
    Парсить вихідний файл PLOTFILE (.plt) для отримання значень концентрацій.
    """
    results = []
    if not os.path.exists(plt_path):
        raise FileNotFoundError("Result file (aermod.plt) not found.")

    with open(plt_path, 'r') as f:
        for line in f:
            # Пропускаємо заголовки (зазвичай починаються з * або літер)
            if line.strip().startswith('*') or not line.strip()[0].isdigit():
                continue

            # Формат PLT: X Y CONC ...
            parts = line.split()
            if len(parts) >= 3:
                try:

```

```
res = {  
    'x': float(parts[0]),  
    'y': float(parts[1]),  
    'concentration': float(parts[2])  
}  
results.append(res)  
except ValueError:  
    continue # Пропустити некоректні рядки  
return results  
}
```

ДОДАТОК В.

СПИСОК ПУБЛІКАЦІЙ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

1. Тимчишин Б., Манжула В. Математичне та алгоритмічне забезпечення інтелектуалізованої програмної системи для проактивного управління екосистемою міста. *Вимірювальна та обчислювальна техніка в технологічних процесах*. 2025. вип. 84, вип. 4, с. 35–48. <https://doi.org/10.31891/2219-9365-2025-840-4>
2. В.І. Манжула, Б.С. Тимчишин. Семантичне моделювання гетерогенних джерел шкідливих викидів в атмосферу для проактивного управління екосистемою міста. *Наукові праці Донецького національного технічного університету. Серія: “Інформатика, кібернетика та обчислювальна техніка”*. 2025. №2 (41), с. 118–127. <https://doi.org/10.31474/1996-1588-2025-2-41-118-127>
3. Тимчишин В., Мельник А., Дивак Т., Тимчишин Б., Файфура В., Момотюк Є., Костик Б. Інтегральний показник для оцінювання ефективності спеціалізованих моніторингових систем. *Оптико-електронні інформаційно-енергетичні технології*. 2024. 48(2). с. 234–241. <https://doi.org/10.31649/1681-7893-2024-48-2-234-241>
4. Tymchyshyn V., Tymchyshyn B., Melnyk A., Manzhula V., Faifura V., Romanets I. The System Architecture of the Software for Modeling Harmful Emissions in Soil. *Proceedings International Conference on Advanced Computer Information Technologies ACIT*. 2023, P. 58–62. <https://doi.org/10.1109/ACIT58437.2023.10275416> (Scopus)
5. Tymchyshyn V., Melnyk A., Tymchyshyn B., Faifura V., Mazur I.-S., Honchar L. Data Management Service Architecture of the Software for Modeling Harmful Emissions in Soil. *Proceedings International Conference on Advanced Computer Information Technologies ACIT*. 2024, P. 678–682. <https://doi.org/10.1109/ACIT62333.2024.10712526> (Scopus)
6. Tymchyshyn V., Porplytsya N., Melnyk A., Tymchyshyn B. Software for Modelling the Air Pollution by Vehicles. *CEUR-WS: Proceedings of the*

International Conference Advanced Computer Information Technologies (ACIT 2018). 2018. Vol 2300. pp. 207-210. <https://ceur-ws.org/Vol-2300/Paper50.pdf> (Scopus)

7. Тимчишин В.С., Порплиця Н.П., Тимчишин Б.С. Програмний комплекс для моделювання забруднення атмосфери шкідливими викидами автотранспорту в часі. Сучасні комп'ютерні інформаційні технології: Матеріали Всеукраїнської конференції з міжнародною участю ACIT'2017. Тернопіль: ТНЕУ, 2017. с.233-234. <https://dspace.wunu.edu.ua/items/55add516-6724-4334-8e74-9f53b6d3bbcc>

ДОДАТОК Д

АКТИ ВПРОВАДЖЕНЬ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

АКТ
про впровадження результатів дисертаційної роботи
Тимчишина Богдана Степановича
**«Інтелектуалізована програмна система для проактивного управління
екосистемою міста»**

Даний акт складений про те, що при організації робіт з планування модернізації системи теплопостачання та екологічного моніторингу комунального підприємства теплових мереж «Тернопільміськтеплокомуненерго» Тернопільської міської ради використано результати дисертаційної роботи здобувача ступеня доктора філософії кафедри комп'ютерних наук Західноукраїнського національного університету Тимчишина Богдана Степановича.

Впроваджено методику автоматизованого сценарного аналізу для оцінки екологічного впливу об'єктів теплогенерації. Це дозволило перейти від трудомістких ручних розрахунків до автоматизованого моделювання сценаріїв типу «що-якщо» (оцінка наслідків зміни палива, зміни навантаження котлів, виведення з експлуатації застарілих потужностей). Використання методики скоротило час підготовки екологічних обґрунтувань для проєктних рішень з декількох днів до 20-30 хвилин.

Застосовано метод оптимізації розміщення джерел розподіленої генерації (когенераційних установок) в умовах щільної міської забудови. На основі розроблених алгоритмів було проведено моделювання варіантів розміщення нових енергогенеруючих потужностей (КГУ) у мікрорайоні «Сонячний» м. Тернопіль. Результати моделювання використані для вибору локацій, що мінімізують концентрацію забруднюючих речовин (NOx) у зонах житлової забудови та соціальних об'єктів (шкіл, лікарень), гарантуючи дотримання санітарних норм.

Використано семантичну модель (онтологію) міської екосистеми для інтеграції різнорідних даних. Запропонований підхід дозволив уніфікувати зберігання технічних паспортів обладнання, геопросторових даних та результатів екологічного моніторингу в єдиному цифровому середовищі, що спрощує інвентаризацію джерел викидів та підготовку звітності.

Використання розроблених Тимчишином Б.С. моделей та програмних засобів дозволило підвищити обґрунтованість управлінських рішень щодо модернізації теплової інфраструктури міста, забезпечити проактивне реагування на ризики перевищення гранично допустимих концентрацій шкідливих речовин та підвищити рівень екологічної безпеки експлуатації теплогенеруючого обладнання.

Рекомендувати подальше використання розробленої інтелектуалізованої системи при розробці схем теплопостачання міста та плануванні заходів з повоєнного відновлення енергетичної інфраструктури.

Директор «Тернопільміськтеплокомуненерго»



А.К. Чумак

*Вих. №16 від 23.12.2025 р.
м.Тернопіль*

АКТ
про впровадження результатів дисертаційної роботи
Тимчишина Богдана Степановича

«Інтелектуалізована програмна система для проактивного управління екосистемою міста»

Даний акт складений про те, що результати дисертаційної роботи Тимчишина Богдана Степановича «Інтелектуалізована програмна система для проактивного управління екосистемою міста» були використані у проєктній діяльності при розробці програмних систем.

Використано запропонований у дисертації підхід до побудови сервіс-орієнтованої архітектури, що базується на інкапсуляції обчислювальних ядер у мікросервіси-обгортки. Це дозволило уніфікувати процес інтеграції сторонніх розрахункових модулів у сучасні хмарні веб-додатки (Python/Node.js), забезпечивши асинхронну обробку запитів.

Впроваджено архітектурний підхід до розділення даних на семантичний рівень, геопросторовий рівень та рівень файлових артефактів. Дане рішення використано при проєктуванні бекенду з використанням геоінформаційних систем, що дозволило оптимізувати виконання складних просторових запитів при збереженні семантичної цілісності даних. Використано принципи побудови доменних онтологій для створення динамічних систем валідації вхідних даних. Це дозволило реалізувати гнучкі механізми перевірки користувацьких сценаріїв без необхідності жорсткого кодування правил у бізнес-логіці додатків.

Використання результатів дисертації дозволило скоротити час розробки архітектури високонавантажених систем, підвищити надійність інтеграційних рішень та розширити експертність у сфері розробки програмного забезпечення.

Директор ТОВ Кравдін
Олена Гос





МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

НАУКОВО-ДОСЛІДНА ЧАСТИНА

46009, Україна, м. Тернопіль, вул. Львівська, буд. 5А, тел. (0352) 51-75-52

№ 21/ 104 - 2025

« 26 » 12 20 25 року

ДОВІДКА про участь у виконанні НДР

Видана **ТИМЧИШИНУ Богдану Степановичу** про те, що у 2025 році на посаді молодшого наукового співробітника він взяв участь у підготовці дослідження молодих учених Західноукраїнського національного університету, яке фінансувалося за коштів Державного бюджету України: «Математичне та програмне забезпечення прототипу біогазової установки з підвищеною ефективністю функціонування» (державний реєстраційний номер 0124U000076).

Начальник
науково-дослідної частини



Віта СЕМАНЮК



ЗАТВЕРДЖУЮ

Проректор з навчально-методичної
роботи
Західноукраїнського національного
університету

Віктор ОСТРОВЕРХОВ

» 2025 р.

АКТ

про впровадження в освітній процес Західноукраїнського національного
університету результатів дисертаційної роботи

Тимчишина Богдана Степановича

«Інтелектуалізована програмна система для проактивного управління
екосистемою міста»

Даний акт складений про те, що результати дисертаційної роботи Тимчишина Богдана Степановича «Інтелектуалізована програмна система для проактивного управління екосистемою міста» використані в освітньому процесі кафедри комп'ютерних наук факультету комп'ютерних інформаційних технологій Західноукраїнського національного університету для здобувачів вищої освіти спеціальності 121 «Інженерія програмного забезпечення».

При викладанні дисциплін «Архітектура та проектування програмного забезпечення» та «Технологія розробки програмного забезпечення та систем моделювання» використано розроблену в дисертації схему сервіс-орієнтованої архітектури для демонстрації патерну «Wrapper/Adapter». На прикладі інтеграції консольного ядра AERMOD (Fortran) у сучасну веб-систему студенти вивчають методи інкапсуляції застарілого коду в мікросервіси та організацію асинхронної взаємодії через черги повідомлень.

Декан факультету комп'ютерних
інформаційних технологій,
к.т.н., доцент

Ігор ЯКИМЕНКО

Завідувач кафедри комп'ютерних наук
д.т.н., професор

Андрій ПУКАС