https://orcid.org/0000-0003-3394-6030

УДК 004.023:621.039.76

М.А. ПАНТІН[*]

ІНФОРМАЦІЙНА СИСТЕМА ОЦІНОК РАДІАЦІЙНОГО ЗАБРУДНЕННЯ В УМОВАХ НЕПОВНИХ ДАНИХ НА НЕРЕГУЛЯРНІЙ МЕРЕЖІ СПОСТЕРЕЖЕНЬ

[*]Інститут проблем математичних машин і систем НАН України, м. Київ, Україна

Анотація. *Пропонується підхід до побудови просторової мапи радіаційного забруднення складних промислових об'єктів у випадках, коли детальне обстеження є обмеженим або небезпечним через високі дозові навантаження. Наявні вимірювання потужності еквівалентної дози отримуються лише в окремих доступних точках і формують нерегулярну мережу спостережень. Задачу розглянуто як обернену задачу реконструкції просторового розподілу активності за обмеженою кількістю дозиметричних вимірювань, що дозволяє явно враховувати фізичні обмеження моделі та геометрію об'єкта. Розроблено інформаційну систему, що охоплює всі основні етапи роботи з дозиметричними даними — від завантаження та формування початкового наближення до ітераційного відновлення розподілу активності та перевірки результатів за контрольними точками. Функціональну структуру описано з використанням IDEF0 (п'ять етапів обробки зі зворотними зв'язками). Архітектуру реалізовано як багаторівневу систему: вебрівень (Nginx, Gunicorn, Django), обчислювальний вузол (Python, NumPy, Pandas) і рівень зберігання даних (PostgreSQL), що взаємодіють через брокер повідомлень RabbitMQ. Відокремлення обчислювального ядра від вебрівня дозволяє використовувати систему як самостійну систему підтримки прийняття рішень або як обчислювальний модуль у складі інших інформаційних систем. Апробацію виконано на експериментальних даних Об'єкта «Укриття» Чорнобильської АЕС: за 7000 ітерацій еволюційного алгоритму досягнуто середньої абсолютної відсоткової похибки близько 13 % при відносній похибці вимірювань близько 17 %, що відповідає межі точності, зумовленої похибкою вхідних вимірювань.*
Ключові слова: *радіаційне картографування, еволюційний алгоритм, обернена задача, неповні дані, нерегулярна мережа спостережень, інформаційна система, система підтримки прийняття рішень, Об'єкт «Укриття».*

Abstract. *An approach is proposed for constructing a spatial radiation contamination map of complex industrial facilities in cases where detailed site survey is limited or hazardous due to high dose rates. Under such conditions, available equivalent dose rate measurements are obtained only at individually accessible points, forming an irregular observation network. The problem is treated as an inverse problem of reconstructing the spatial activity distribution from a limited number of dosimetric measurements, allowing for explicit consideration of the physical constraints of the model and the facility geometry. An information system that covers all major stages of dosimetric data processing — from data loading and initial approximation to iterative activity distribution reconstruction and result validation against control points — has been developed. The functional structure is described using IDEF0 (five processing stages with feedback loops). The architecture is implemented as a multi-tier system consisting of a web tier (Nginx, Gunicorn, Django), a computational node (Python, NumPy, Pandas), and a data storage tier (PostgreSQL), interacting via the RabbitMQ message broker. Decoupling the computational core from the web tier allows the system to operate either as a standalone decision support system or as a computational module within other information systems. Validation has been performed on experimental data from the Shelter Object of the Chornobyl NPP. After 7,000 iterations of the evolutionary algorithm, a mean absolute percentage error of approximately 13 % has been achieved at a relative measurement uncertainty of approximately 17 %, corresponding to the accuracy limit imposed by the input measurement error.*
Keywords: *radiation mapping, evolutionary algorithm, inverse problem, incomplete data, irregular observation network, information system, decision support system, Shelter Object.*

ISSN 1028-9763. Математичні машини і системи. 2026. № 3

DOI: 10.34121/1028-9763-2026-3-49-59

1. Вступ

Оцінка радіаційного забруднення промислових об’єктів та територій, що зазнали впливу ядерних аварій, залишається важливою прикладною задачею радіаційної безпеки. У багатьох практичних ситуаціях пряме детальне обстеження території неможливе через високі дозові навантаження, складну геометрію об’єкта або обмежений доступ до окремих зон. Типовим прикладом є Об’єкт «Укриття» (ОУ) — споруда, зведена над четвертим блоком Чорнобильської АЕС, — у межах якого дозиметричні вимірювання традиційно виконуються вибірково, на нерегулярній мережі позицій і лише в тих точках, де це припустимо з позицій радіаційного захисту [1, 2].

Задача побудови мапи забруднення у такій постановці є оберненою задачею: за обмеженим набором вимірювань потужності еквівалентної дози (ПЕД) необхідно відновити просторовий розподіл активності на поверхні об’єкта. Класичні методи просторової інтерполяції (обернених зважених відстаней, крайгінг, сплайни) орієнтовані на згладжування значень усередині області спостережень і не дають задовільних результатів, коли мережа розріджена, нерегулярна, а геометрія об’єкта зумовлює часткове екранування окремих ділянок від детекторів. Перспективною альтернативою є методи оптимізаційної реконструкції, зокрема еволюційні алгоритми, які дають змогу врахувати фізичні обмеження задачі та залучати апріорну інформацію про розподіл джерел [3, 4].

Разом із тим сам алгоритм реконструкції є лише обчислювальним ядром прикладної задачі. Для реального використання у практиці радіаційної безпеки потрібна інформаційна система, яка приймає експериментальні дані, виконує обчислення, забезпечує можливість експертного контролю та ітераційного коригування результатів, подає мапу у вигляді, зручному для ухвалення рішень, та припускає інтеграцію із зовнішніми системами моніторингу. Проєктування таких систем у наявних публікаціях висвітлено фрагментарно: зазвичай розглядається або обчислювальне ядро, або окремі підходи до візуалізації, тоді як цілісний опис архітектури, розподілу відповідальностей між компонентами та механізмів взаємодії із зовнішніми системами відсутній.

Метою роботи є опис інформаційної системи радіаційного картографування, призначеної для роботи в умовах неповних даних, зібраних на нерегулярній мережі спостережень, з акцентом на функціональній моделі, архітектурних рішеннях, що забезпечують відокремлення обчислювального ядра від рівня подання, та на результатах апробації на експериментальних даних ОУ Чорнобильської АЕС.

Наукова новизна роботи пов’язана з узгодженим архітектурним рішенням, яке дозволяє використовувати одну й ту саму обчислювальну основу як у вигляді автономної системи підтримки прийняття рішень (СППР), так і як компонента зовнішніх систем, без зміни інтерфейсів обміну даними.

2. Постановка задачі та вимоги до системи

Вхідними даними для системи є множина вимірювань потужності еквівалентної дози (ПЕД), кожне з яких характеризується трьома просторовими координатами точки вимірювання та зафіксованим показом детектора. Мережа спостережень не є регулярною: положення точок визначається не сіткою, а траєкторіями руху дозиметричних платформ, їхніми обмеженнями з безпеки та технічною доступністю окремих ділянок. Кількість вимірювань, як правило, на кілька порядків менша за кількість елементів поверхні об’єкта, на якій потрібно відновити розподіл активності, тобто задача є невизначеною.

Фізична модель, покладена в основу реконструкції, розглядає поверхню об’єкта як сукупність точкових ізотропних джерел із кроком 1 м. Зв’язок між показами віртуальних детекторів і активностями джерел виражається системою лінійних рівнянь із коефіцієнта-

ми, що відповідають закону обернених квадратів та умовам екранування (детальне виведення моделі наведено у [3, 4] і не відтворюється у цій роботі). Пошук розв'язку реалізовано модифікованим еволюційним алгоритмом з ініціалізацією, що може використовувати попередньо виконану просторову інтерполяцію.

Зважаючи на наведену постановку задачі, до інформаційної системи висунуто такі вимоги: приймати вхідні дані у стандартизованому форматі (координати точок та значення ПЕД) без прив'язки до конкретного засобу збирання; виконувати ресурсоємні обчислення асинхронно, не блокуючи користувача під час очікування результату; забезпечувати можливість повторних прогонів алгоритму з коригуванням параметрів за результатами експертного аналізу; подавати результати у формі, придатній як для безпосереднього перегляду людиною-оператором, так і для передавання у зовнішні інформаційні системи; припускати можливість експлуатації як автономно, так і у складі ширшої системи радіаційної безпеки. Перелічені вимоги зумовлюють багаторівневу архітектуру з асинхронною обробкою та розподілом відповідальностей між мережевим, обчислювальним і представницьким рівнями.

3. Функціональна модель системи

Для формалізованого опису функціональної структури системи застосовано метод IDEF0. Вибір методу зумовлений його здатністю подавати складні процеси у вигляді ієрархії взаємопов'язаних функцій з явним розмежуванням входів, виходів, керуючих впливів та ресурсів, що дозволяє узгоджено описати як інформаційні потоки, так і ролі виконавців [5, 6].

Контекстна діаграма розглядає систему як єдину функцію, що перетворює набір первинних дозиметричних вимірювань та модель території у мапу радіоактивного забруднення. Керуючими впливами виступають координати контрольних точок із відомою радіоактивністю та межі ділянки картографування. Ресурси системи охоплюють дозиметричне оснащення (за потреби збирання нових вимірювань), обчислювальну платформу та фахівців, що забезпечують супровід процесу.

На діаграмі декомпозиції (рис. 1) виділено п'ять функціональних етапів:

а) збір і підготовка первинних даних із використанням історичних записів та нових вимірювань;

б) розрахунок рівнів забруднення еволюційним алгоритмом на основі підготовленої тривимірної моделі об'єкта;

в) контроль і оцінка якості результатів за контрольними точками з можливістю повторного прогону алгоритму зі скоригованими параметрами;

г) перенесення перевірених значень активності на геометричну модель місцевості та формування попередньої мапи;

д) верифікація побудованої мапи за контрольними вимірюваннями і заданими межами ділянки. Між етапами передбачено зворотні зв'язки, що відповідає ітераційній природі задачі оберненої реконструкції.

4. Архітектурні рішення та розподіл компонентів

Архітектуру системи розглянуто з кількох точок зору: структурної (склад компонентів), поведінкової (їх взаємодія у часі) та ієрархічної (розподіл за рівнями). Такий опис відповідає підходам, закріпленим у стандарті ISO/IEC/IEEE 42010 [7], і дозволяє узгоджено подати як структуру, так і поведінку системи [8–10].

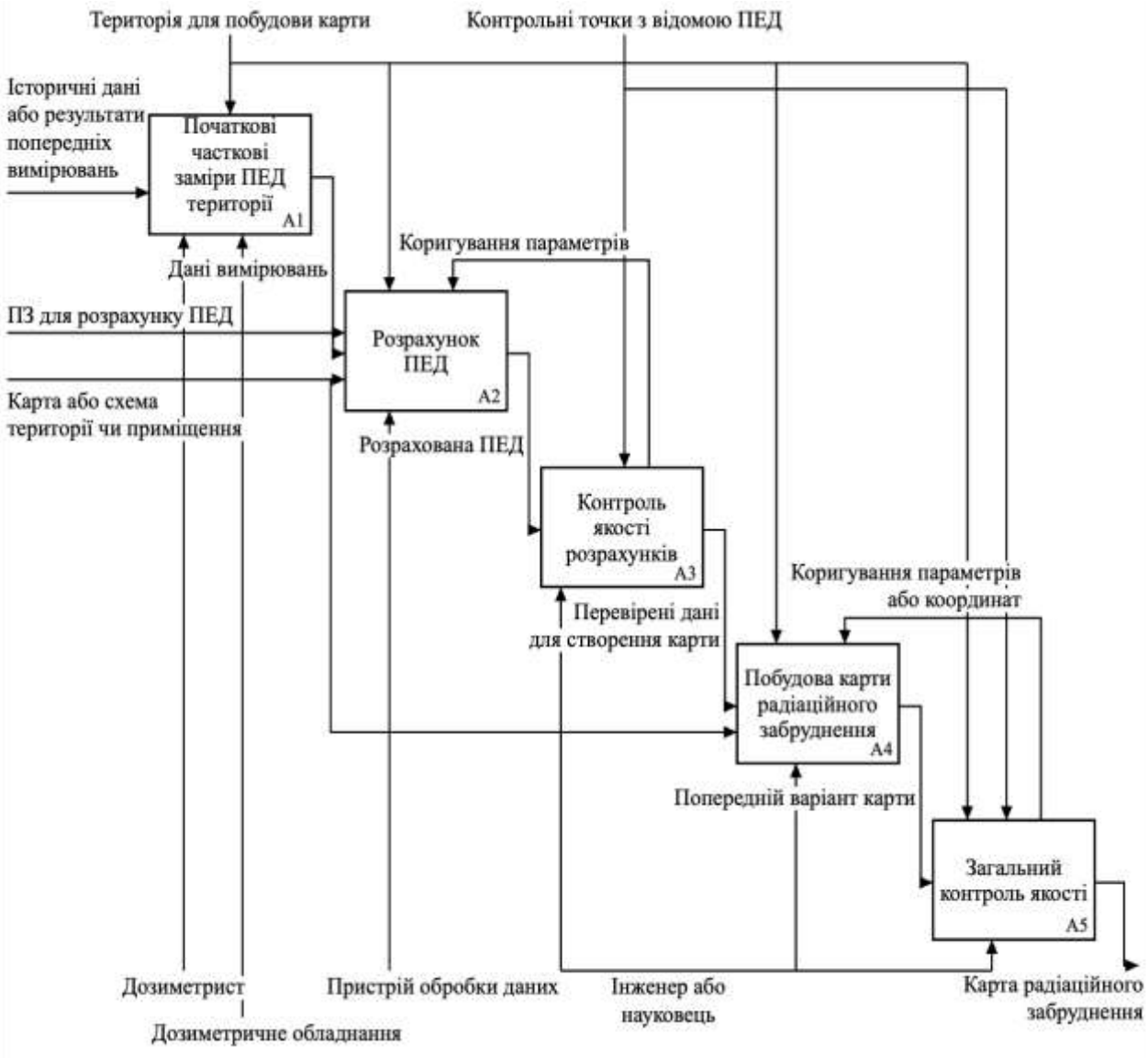

Рисунок 1 — Діаграма декомпозиції функціональної моделі системи (IDEF0)

4.1. Функціональні вимоги та ролі користувачів

Для опису функціональних вимог із позицій зовнішніх користувачів побудовано діаграму прецедентів UML (рис. 2) [11, 12]. У системі виділено чотири ролі: користувача, який ініціює обробку та переглядає результати; наукового співробітника, який виконує експертний контроль якості та ухвалює рішення щодо повторного прогону алгоритму; дозиметриста, що забезпечує надходження первинних вимірювань; адміністратора, відповідального за супровід системи. Основні прецеденти охоплюють повний цикл — від завантаження вхідних даних та задання параметрів до отримання, аналізу та експорту мапи забруднення. Зв’язки типу «include» використано для опису обов’язкових підпроцесів, тоді як зв’язки «extend» — для умовних сценаріїв, зокрема повторної обробки даних з урахуванням експертних коригувань.

Рисунок 2 — Діаграма прецедентів

4.2. Взаємодія компонентів

Діаграми послідовностей охоплюють два основні сценарії. При надсиланні вхідних даних (HTTP POST, рис. 3) запит клієнта проходить через зворотний проксі-сервер Nginx і сервер застосунку Gunicorn до вебшару Django, де здійснюється первинна валідація. Подальша обробка виконується асинхронно через публікацію завдання у брокері повідомлень RabbitMQ. Обчислювальний процес мовою Python забирає завдання з черги, виконує реконструкцію з використанням бібліотек NumPy та Pandas, формує інтерактивну мапу у форматі HTML засобами Plotly й зберігає результат у файловому сховищі. При запиті на перегляд (HTTP GET) вебшар звертається до попередньо згенерованого файла та повертає його клієнту; завдяки відсутності асинхронної обробки у цьому сценарії забезпечується мінімальна затримка відгуку.

Усі UML-діаграми, наведені у статті, підготовлено з використанням інструмента PlantUML [13], що забезпечує текстоорієнтований опис діаграм із наступною автоматичною генерацією графічних представлень відповідно до офіційної специфікації UML [14]. Такий підхід спрощує супровід архітектурної документації разом із програмним кодом у системі контролю версій та гарантує відтворюваність діаграм.

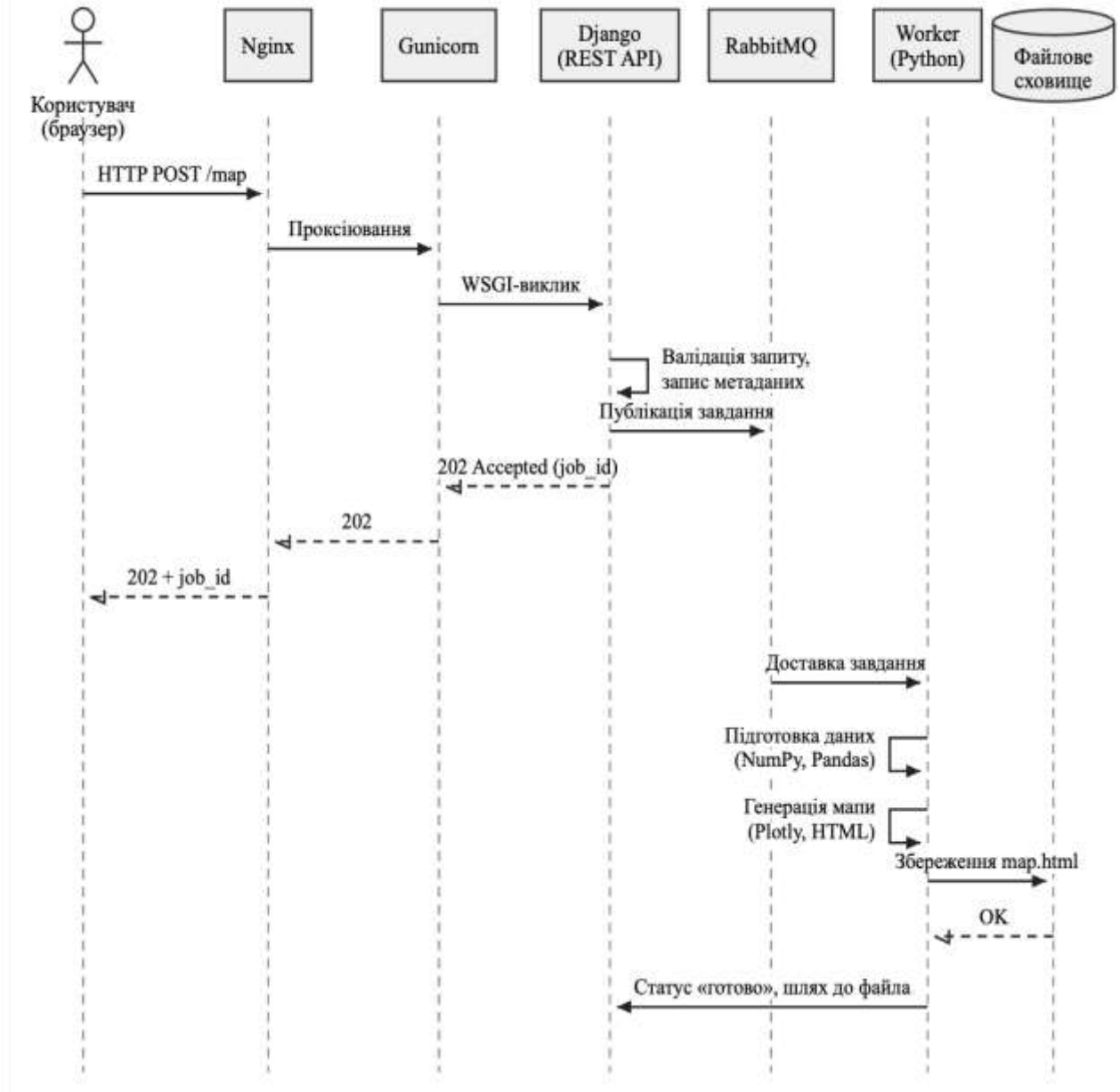

Рисунок 3 — Діаграма послідовностей: обробка вхідних даних (HTTP POST)

4.3. Статична структура обчислювального ядра

Статичну структуру обчислювального ядра подано на діаграмі класів (рис. 4), де виділено три логічні рівні. Рівень предметної моделі містить класи Point (воксель розрахункової області з координатами й поточним значенням активності), Sensor (дозиметричний датчик із виміряним значенням та посиланням на точки, які він «спостерігає») та Surface (агрегуючий контейнер, що об’єднує множини точок та датчиків). Рівень обробки містить модуль початкової ініціалізації Step0Init, який формує стартове наближення методом найменших квадратів, та модуль EvolutionaryReconstruction, що реалізує ітераційне уточнення розподілу активності. Допоміжний рівень відповідає за завантаження геометричної моделі, даних сенсорів і табличне подання через DataFrame-структури. Поділ на модулі узгоджується з принципом розподілу відповідальностей і сприяє повторному використанню компонентів [15].

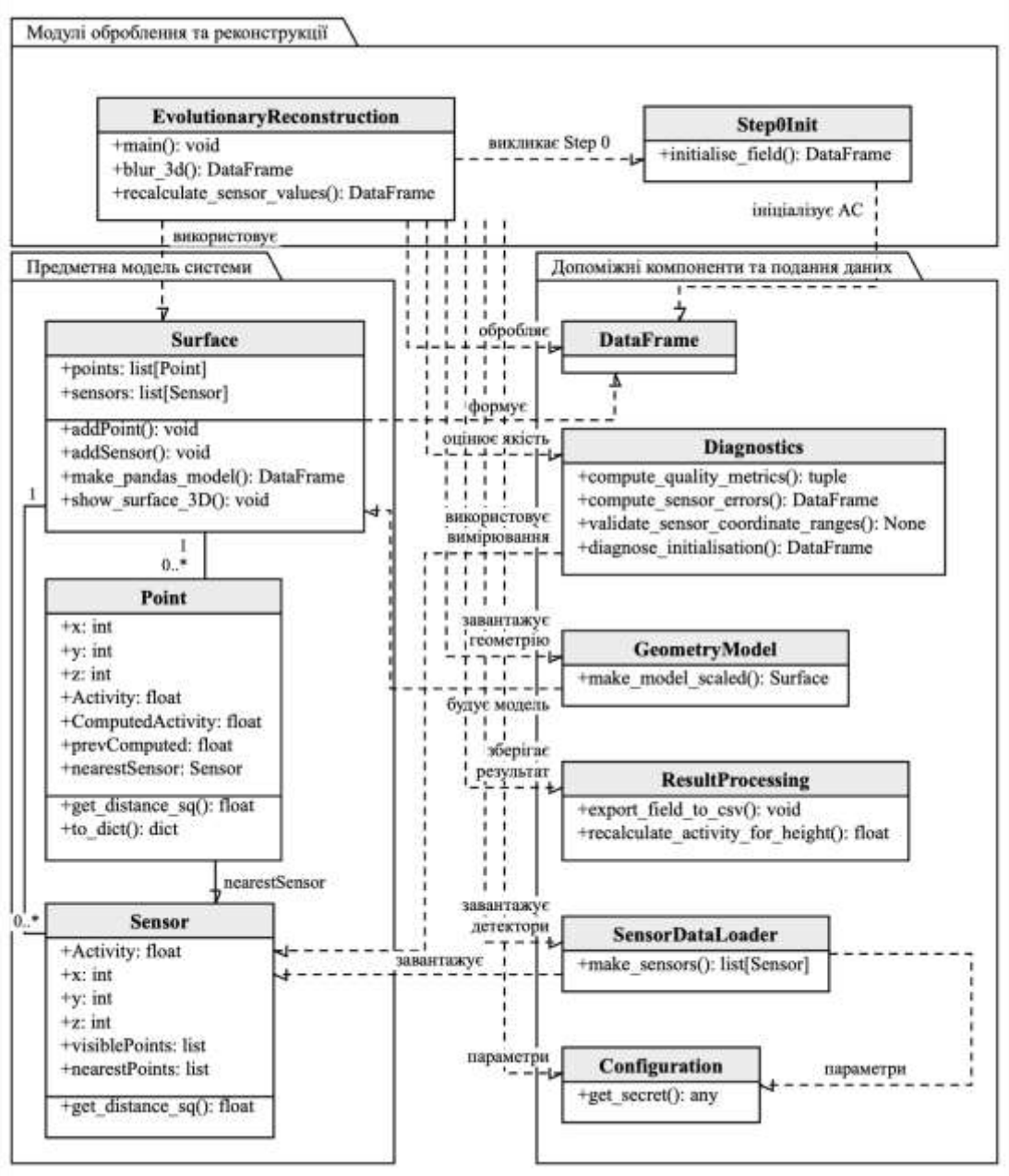

Рисунок 4 — Діаграма класів

4.4. Розгортання компонентів та інтероперабельність

Діаграма розгортання (рис. 5) фіксує розподіл компонентів за вузлами виконання. Виділено три вузли — вебсервер, обчислювальний вузол та сховище, кожен з яких виконує функцію, що може масштабуватися або замінюватися незалежно. Вебсервер приймає мережеві запити (Nginx), обробляє їх на рівні застосунку (Gunicorn + Django REST API) та передає обчислювальні завдання у чергу. Обчислювальний вузол несе ресурсоємну частину — Python-процес із модулями Step0Init та EvolutionaryReconstruction, а також локальний брокер RabbitMQ. Сховище поєднує реляційну базу даних PostgreSQL для метаданих завдань (запит, статус, часові мітки, довідники) та файлову систему для вхідних вимірювань (CSV/JSON) і згенерованих результатів (HTML, JSON).

Зовнішня взаємодія відбувається через єдину точку входу — вебсервер. Фахівець із радіаційної безпеки звертається до системи через браузер та отримує інтерактивну мапу; зовнішня інформаційна система (наприклад, платформа радіаційного моніторингу, у складі

якої розроблена система використовується як обчислювальний модуль) взаємодіє із системою через той самий REST API, обмінюючись структурованими даними. Обчислювальне ядро не знає, хто саме ініціював запит, і не нав'язує конкретного способу подання результатів: на вхід приймається набір вимірювань, на виході формується масив пар «координата — значення ПЕД», що далі трансформується або у HTML-візуалізацію, або у структуровану відповідь. Відокремлення обчислювального ядра від рівня подання забезпечує два режими використання системи — як автономної СППР та як обчислювального компонента у складі зовнішніх систем.

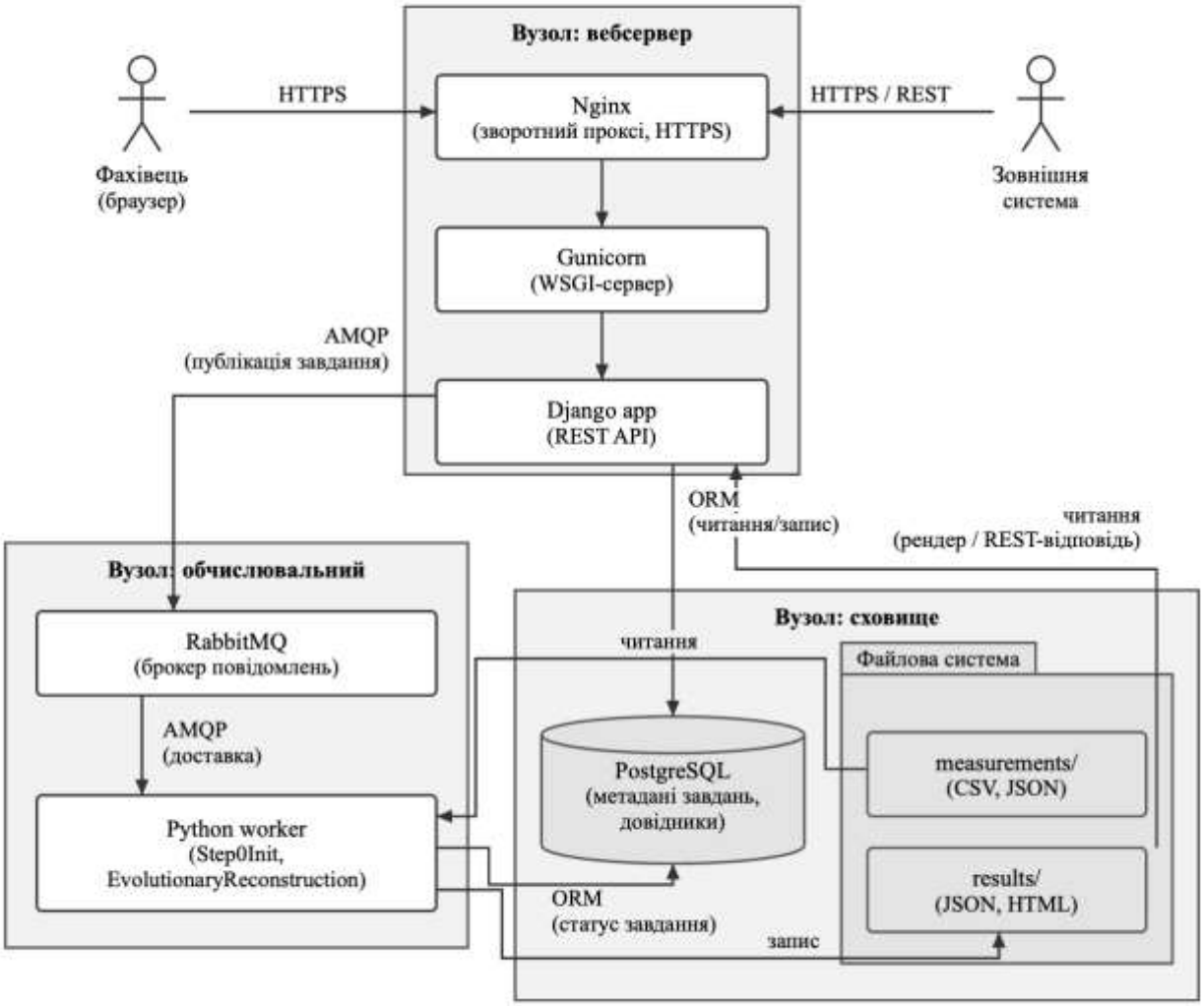

Рисунок 5 — Діаграма розгортання компонентів системи

5. Технологічний стек реалізації

Технологічний стек обрано з урахуванням практичних вимог до системи: надійності компонентів, сумісності з багаторівневою архітектурою та зручності обробки наукових даних у середовищі Python. Вебрівень побудовано на фреймворку Django із сервером застосунку Gunicorn за стандартом WSGI та зворотним проксі-сервером Nginx. Така комбінація є типовою для сучасних вебзастосунків, розроблених із використанням Python, і широко представлена в офіційній документації та практичних керівництвах [16, 17]. Nginx завдяки подієво-орієнтованій моделі обробки ефективно працює з великою кількістю одночасних з'єднань [18].

Для асинхронної обробки використано брокер повідомлень RabbitMQ, що реалізує модель черг і дає змогу відокремити вебрівень від ресурсоємних обчислень та незалежно

масштабувати обчислювальні компоненти [19, 20]. Метадані завдань і довідники зберігаються у реляційній базі PostgreSQL; взаємодія організована через Django ORM, що ізолює прикладний код від конкретного діалекту SQL.

Чисельні обчислення реалізовано з використанням бібліотек NumPy та Pandas, які забезпечують ефективну роботу з багатовимірними масивами, табличними структурами й векторизованими операціями [21, 22]. Візуалізація результатів формується у вигляді інтерактивних HTML-сторінок засобами бібліотеки Plotly [23]; згенеровані файли зберігаються у файловому сховищі, що дозволяє повторно використовувати підготовлені мапи без повторного виконання ресурсоємних обчислень і зменшує час відповіді системи.

Сукупно обраний стек забезпечує модульність системи на рівні, що виходить за межі внутрішньої організації програмного коду: модуль еволюційної реконструкції не містить прив'язки ні до джерела вхідних даних, ні до способу подання результатів, що й дозволяє описані у попередньому підрозділі два режими використання.

6. Апробація на даних Об'єкта «Укриття»

Працездатність системи перевірено на експериментальних даних, отриманих під час дозиметричного обстеження покрівлі ОУ Чорнобильської АЕС з використанням безпечного візка системи основних кранів нового безпечного конфайнмента. Вимірювання ПЕД виконувалися детектором БДБГ-09; діапазон зафіксованих значень склав від 230 до 28 490 мкЗв/год, відносна похибка вимірювань становила близько 17 % [1, 2]. Особливістю набору даних є його неповнота: точки спостережень утворюють нерегулярну мережу, прив'язану до траєкторій переміщення візка, і охоплюють лише частину поверхні об'єкта.

Первинні дані були завантажені у систему через REST API, після чого виконано просторову інтерполяцію та побудовано теплову карту ПЕД як базову ілюстрацію розподілу поля над об'єктом. На наступному етапі запущено еволюційний алгоритм реконструкції з ініціалізацією початкової популяції на основі фізичних законів. Для оцінки якості застосовано три метрики: похибку апроксимації (САП), середню абсолютну відсоткову похибку (САВП) та коефіцієнт детермінації (R^2), обчислені на наборі віртуальних детекторів, що відтворюють реальні точки вимірювань.

Аналіз збіжності показав, що значення САВП монотонно спадає зі зростанням кількості ітерацій, причому після близько 7000 ітерацій подальше поліпшення стає несуттєвим на тлі природної похибки вимірювань. Значення САВП зменшується з ≈19 % на початковій ітерації до ≈13 % наприкінці й лежить у межах власної відносної похибки детектора (≈17 %), що відповідає досягненню межі точності, зумовленої похибкою вхідних вимірювань. Спроби продовжити оптимізацію з метою зменшення САВП нижче цього рівня призводять до перенавчання під конкретний набір вимірювань без істотного поліпшення узагальнювальних властивостей моделі.

Отриманий результат підтверджує придатність системи до роботи в умовах нерегулярних та неповних вхідних даних. Архітектурні рішення — асинхронна обробка через чергу та відокремлення обчислювального ядра від вебрівня — дали можливість виконати тривалі розрахунки без блокування користувацького інтерфейсу, а ітераційна структура процесу дозволила зіставити проміжні результати з експертною оцінкою й за потреби повторити прогін зі скоригованими параметрами.

7. Висновки

У роботі розроблено інформаційну систему радіаційного картографування, орієнтовану на умови неповних і нерегулярних даних. Запропонована функціональна модель та архітектурні рішення дозволяють виконувати реконструкцію розподілу активності з урахуванням обмежень доступності вимірювань.

Апробація на даних Об'єкта «Укриття» Чорнобильської АЕС підтвердила працездатність системи: за 7000 ітерацій еволюційного алгоритму досягнуто середньої абсолютної відсоткової похибки близько 13 % при відносній похибці вимірювань близько 17 %, що відповідає досягненню межі точності, зумовленої похибкою вхідних вимірювань. Отримані результати дозволяють позиціонувати систему як інструмент підтримки прийняття рішень для планування робіт і маршрутизації робототехнічних платформ у зонах обмеженого доступу.

Подальші напрями досліджень пов'язані з інтеграцією системи з роботизованими засобами радіаційної розвідки, розширенням набору стратегій ініціалізації еволюційного алгоритму та додаванням режиму інкрементної актуалізації мапи за новими вимірюваннями.

СПИСОК ДЖЕРЕЛ

1. Saveliev M.V., Krasnov V.A., Levchenko A.P. et al. Measuring the equivalent dose rate over the Shelter object after completion of the New Safe Confinement. *Nuclear power and the environment*. 2020. Vol. 4, Issue 19. P. 50–56. DOI: https://doi.org/10.31717/2311-8253.20.4.6.
2. Михайлов О.В., Савельєв М.В., Пантін М.А. Результати контролю потужності еквівалентної дози γ-випромінювання у приміщеннях об'єкта «Укриття» ЧАЕС. *Ядерна енергетика та довкілля*. 2024. Вип. 2, N 30. P. 51–60. DOI: https://doi.org/10.31717/2311-8253.24.2.6.
3. Saveliev M.V., Pantin M.A., Skiter I.S. et al. Implementation of Novel Evolutional Algorithm for 3-Dimensional Radiation Mapping and Gamma-Field Reconstruction within the Chornobyl Sarcophagus. *MDPI Algorithms*. 2023. Vol. 16, Issue 4. P. 204. DOI: https://doi.org/10.3390/a16040204.
4. Pantin M.A., Saveliev M.V., Skiter I.S. An approach of use evolutional algorithms for radiation mapping. *Проблеми зняття з експлуатації об'єктів ядерної енергетики та відновлення навколишнього середовища. INUDECO 22*: Матеріали VII Міжнар. конф. (27.04.2022). Slavutych, Ukraine, 2022. Also available online. URL: https://inudeco.pro/ (збірник INUDECO 2022).
5. Defense Acquisition University. Systems Engineering Fundamentals. Fort Belvoir, VA: Defense Acquisition University Press, 2001. 222 p.
6. Yourdon E. Modern Structured Analysis. Englewood Cliffs, NJ: Prentice Hall, 1989. 672 p. ISBN 978-0-13-598624-0.
7. ISO/IEC/IEEE 42010:2022. Systems and software engineering — Architecture description. Geneva: ISO, 2022. ISBN 978-0-539-24940-8. 2022.
8. Bass L., Clements P., Kazman R. Software Architecture in Practice. 3. Boston: Addison-Wesley, 2013. 624 p. ISBN 978-0-321-81573-6.
9. Shaw M., Garlan D. Software Architecture: Perspectives on an Emerging Discipline. Upper Saddle River: Prentice Hall, 1996. 269 p. ISBN 978-0-13-182957-2.
10. Kruchten P. Architectural Blueprints — The "4+1" View Model of Software Architecture. *IEEE Software*. 1995. Vol. 12, Issue 6. P. 42–50. DOI: https://doi.org/10.1109/52.469759.
11. Booch G., Rumbaugh J., Jacobson I. The Unified Modeling Language User Guide. 2. Addison-Wesley, 2005.
12. Fowler M. UML Distilled: A Brief Guide to the Standard Object Modeling Language. 3. Addison-Wesley, 2004.
13. PlantUML Documentation. *PlantUML*. URL: https://plantuml.com/ (accessed: 22.01.2026).
14. Unified Modeling Language (UML) Specification. *Object Management Group (OMG) Specifications*. URL: https://www.omg.org/spec/UML/ (accessed: 22.01.2026).
15. Sommerville I. Software Engineering. 10. Boston: Pearson Education Limited, 2015. 816 p. ISBN 978-0-13-394303-0.

16. Django Documentation. *Django Project*. 2024. URL: https://docs.djangoproject.com/en/stable/ (accessed: 22.01.2026).
17. Eby P.J. PEP 3333 — Python Web Server Gateway Interface (WSGI). Python Software Foundation, 2010. URL: https://peps.python.org/pep-3333/ 2010.
18. DeJonghe D. NGINX Cookbook: Advanced Recipes for High-Performance Load Balancing. 2. Sebastopol, CA: O'Reilly Media, 2022. 524 p. ISBN 978-1-0981-2624-7.
19. Hohpe G., Woolf B. Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Boston: Addison-Wesley Professional, 2003. 736 p. URL: https://martinfowler.com/books/eip.html.
20. Videla A., Williams J.J.W. RabbitMQ in Action: Distributed Messaging for Everyone. Shelter Island, NY: Manning Publications, 2012. 320 p. URL: https://www.manning.com/books/rabbitmq-in-action.
21. Harris C.R., Millman K.J., Walt S.J. van der et al. Array programming with NumPy. *Nature*. 2020. Vol. 585. P. 357–362. DOI: https://doi.org/10.1038/s41586-020-2649-2.
22. McKinney W. Data Structures for Statistical Computing in Python. *Proc. of the 9th Python in Science Conference (SciPy)*. 2010. P. 56–61. URL: https://doi.org/10.25080/Majora-92bf1922-00a.
23. Plotly Python Documentation. *Plotly*. 2024. URL: https://plotly.com/python/ (accessed: 22.01.2026).

Стаття надійшла до редакції 24.04.2026 / прийнята до друку 13.08.2026