

О. С. БИЧКОВ, д-р техн. наук, професор;

ORCID: 0000-0002-9378-9535

М. О. МЕЛЬНИК, аспірант,

ORCID: 0009-0000-1180-9487

Київський національний університет імені Тараса Шевченка

ITERATIVE REFINEMENT FUNNEL: АРХІТЕКТУРНИЙ ПАТЕРН ДЛЯ СИСТЕМ ІНТЕРАКТИВНОЇ ДІАГНОСТИКИ З АСИМЕТРИЧНОЮ ОБРОБКОЮ ВІДПОВІДЕЙ

Стаття присвячена розробці та формалізації нового архітектурного патерну Iterative Refinement Funnel (IRF), призначеного для побудови систем інтерактивної діагностики. Патерн адресує задачу ефективного пошуку правильного рішення серед великої кількості варіантів в умовах неповної інформації та інформації, яка поступово надходить. Запропонований патерн ґрунтується на концепції асиметричної обробки відповідей, яка враховує різницю в інформаційній цінності між підтвердженням (YES) та запереченням (NO). Підтвердження ознаки ініціює повний перерахунок моделі машинного навчання з оновленою проєкцією на карту, що самоорганізується, та переранжуванням гіпотез нейронною мережею, тоді як заперечення дозволяє продовжити поточну діагностичну фазу без витратних обчислень.

У роботі запропоновано дві взаємозамінні стратегії вибору питань – Cluster Heuristic та Differential Heuristic, реалізовані через патерн проєктування Strategy, що забезпечує адаптивність до різних сценаріїв діагностики та характеристик вхідних даних. Розроблено комплекс механізмів робастності, включаючи Critical NO для обробки заперечення патогномонічних ознак, Exploration Question для подолання ефекту локальних мінімумів, а також Soft Differential для зниження впливу помилкових або неточних відповідей користувача.

Експериментальна валідація на медичній базі даних, що містить 844 захворювання та 460 унікальних симптомів, демонструє досягнення точності Top-1 на рівні 83,3–84,1% при забезпеченні 150-кратного прискорення порівняно з класичним підходом Expected Information Gain. Додатково встановлено, що використання асиметричної обробки відповідей дозволяє істотно зменшити обчислювальні витрати без втрати якості результатів, що є важливим для систем реального часу. Запропонований патерн характеризується доменною незалежністю та може бути інтегрований у різні інтелектуальні системи, зокрема у задачах підтримки прийняття рішень, де необхідне ефективне інтерактивне звуження простору гіпотез. Крім того, формалізація патерну відкриває можливості для його подальшої автоматизованої оптимізації та розширення в межах гібридних архітектур штучного інтелекту.

Ключові слова: архітектурний патерн, інтерактивна діагностика, машинне навчання, асиметрична обробка, Expected Information Gain, робастність системи, клінічна підтримка прийняття рішень.

Вступ

Системи інтерактивної діагностики застосовуються в різних предметних областях: від медичної діагностики до технічної підтримки, юридичної класифікації та рекомендаційних сервісів. Ці застосування об'єднують необхідність знаходження правильного рішення серед великої кількості варіантів (сотні або тисячі) в умовах, коли інформація надходить поступово через діалог з користувачем.

Традиційний підхід Expected Information Gain (EIG) передбачає вибір на кожному кроці питання, що максимально зменшує ентропію розподілу ймовірностей [4], [6]. Однак цей підхід має суттєві обмеження. По-перше, обчислювальна складність $O(S \times D \times \log D)$, де S – кількість можливих питань, D – кількість варіантів відповіді, призводить до неприйнятних затримок у інтерактивних застосуваннях. По-друге, EIG передбачає симетричну обробку відповідей YES та NO, що не відповідає реальній природі діагностичного процесу.

Аналіз діагностичних сесій виявляє асиметрію: підтвердження ознаки (YES) додає нову позитивну інформацію, яка може змінити ранжування гіпотез; заперечення (NO) лише виключає один варіант, не додаючи конструктивної інформації. Ця асиметрія обумовлює доцільність різної обробки YES та NO відповідей, що є основою запропонованого патерну.

Постановка проблеми

Сучасні інформаційні системи дедалі частіше стикаються із задачею ідентифікації правильного рішення серед великої кількості варіантів в умовах неповної інформації. Медичні діагностичні системи мають обирати серед сотень чи тисяч захворювань, системи технічної підтримки – серед сотень типів несправностей, юридичні класифікатори – серед тисяч прецедентів та категорій [13], [14]. Ці задачі об'єднує неможливість отримати повну інформацію одразу: дані надходять поступово через діалог з користувачем, і кожне додаткове питання має свою вартість у термінах часу та когнітивного навантаження.

Класичний підхід до вирішення таких задач базується на обчисленні Expected Information Gain для кожного можливого питання з подальшим вибором питання, що максимально зменшує ентропію розподілу ймовірностей [4], [6]. Однак цей підхід має принципове обмеження: обчислювальна складність $O(S \times D \times \log D)$, де S – кількість можливих питань, D – кількість варіантів відповіді, робить його непридатним для інтерактивних застосувань з великим простором рішень. При $S = 500$ питань та $D = 1000$ варіантів кожен вибір питання потребує мільйонів операцій, що призводить до неприйнятних затримок у системах реального часу [22], [24].

Окрім обчислювальної складності, існує концептуальна проблема, яка залишається поза увагою класичних підходів. Емпіричний аналіз діагностичних сесій виявляє асиметрію між підтвердженням та запереченням ознаки [17]. Коли користувач підтверджує наявність ознаки, це додає нову позитивну інформацію, яка може змінити ранжування гіпотез. Натомість заперечення лише виключає один варіант, не додаючи конструктивної інформації про правильну відповідь. Існуючі архітектурні рішення не враховують цю асиметрію, обробляючи обидва типи відповідей однаково.

Потреба у спеціалізованому архітектурному патерні для систем інтерактивної діагностики обумовлена також відсутністю стандартизованого підходу до організації таких систем [16, 21]. Розробники змушені щоразу винаходити архітектуру заново, що призводить до неоптимальних рішень, складності підтримки та неможливості повторного використання напрацьованих рішень. Архітектурний патерн, що формалізує найкращі практики побудови діагностичних систем, дозволить суттєво скоротити час розробки та підвищити якість кінцевих рішень [18].

Аналіз останніх досліджень і публікацій

Проблема автоматизованої діагностики має довгу історію досліджень, що бере початок від експертних систем 1970-х років. Робота Shortliffe та Buchanan [2] над системою MYCIN продемонструвала можливість кодування медичних знань у вигляді продукційних правил. Однак експертні системи першого покоління потребували явного формулювання всіх правил експертами, що обмежувало їх масштабованість та адаптивність до нових предметних областей.

Байєсівські мережі, формалізовані у роботі Pearl [3], запропонували математичний апарат для моделювання невизначеності та залежностей між змінними. Цей підхід застосовується у діагностичних системах, однак він не адресує питання ефективної організації інтерактивного діалогу з користувачем. Байєсівське оновлення передбачає симетричну обробку нової інформації незалежно від її характеру.

Методи активного навчання, систематизовані у оглядовій роботі Settles [5], досліджують стратегії вибору найбільш інформативних прикладів для навчання моделі. Концепція Information Gain, центральна для цього напрямку та обґрунтована у класичній роботі Shannon [6], безпосередньо застосовується до задачі вибору питань у діагностичних системах. Сучасні дослідження Li et al. [22] демонструють застосування цих концепцій у системах інтерактивного клінічного висновування на базі великих мовних моделей. Проте активне навчання фокусується на покращенні моделі, а не на архітектурі системи взаємодії з користувачем.

Класична робота Gamma, Helm, Johnson та Vlissides [1], відома як «Gang of Four», систематизувала патерни проектування об'єктно-орієнтованого програмування. Патерни Strategy, State та Template Method, описані у цій роботі, слугують будівельними блоками для реалізації діагностичних систем. Однак ці патерни адресують локальні задачі проектування, а не глобальну архітектуру системи.

Серед архітектурних патернів програмного забезпечення, систематизованих у роботах Buschmann et al. [8] та Shaw та Garlan [9], найближчим до проблематики діагностичних систем є патерн Blackboard, що передбачає співпрацю кількох незалежних джерел знань над спільним робочим простором. Цей патерн успішно застосовувався у системах розпізнавання мовлення та інтерпретації сигналів [10]. Однак Blackboard не специфікує механізми інтерактивного уточнення інформації та не враховує асиметрію між різними типами вхідних даних.

Патерн Model-View-Controller та його варіації [11] визначають розділення відповідальностей у інтерактивних застосуваннях, але не адресують специфіку діагностичного процесу з його ітеративним звуженням простору рішень. Мікросервісна архітектура [12] пропонує підходи до декомпозиції системи на незалежні сервіси, однак не визначає логіку їхньої взаємодії у контексті діагностики. Патерн Pipe-and-Filter [8] описує послідовну обробку даних, але не передбачає циклічності та зворотного зв'язку, характерних для діагностичного процесу.

Карти, що самоорганізуються (Self-Organizing Maps), запропоновані Kohonen [7], знайшли застосування для кластеризації та візуалізації багатовимірних даних. Сучасні дослідження Tripathi [19] та Javed et al. [20] демонструють ефективність SOM для кластерного аналізу медичних даних та часових рядів. У контексті діагностичних систем SOM можуть використовуватися для попередньої фільтрації простору варіантів, однак їх інтеграція в цілісну архітектуру інтерактивної діагностики залишається недостатньо дослідженою.

Останні роботи у галузі клінічних систем підтримки прийняття рішень (CDSS) демонструють значний прогрес у застосуванні машинного навчання [13] – [15]. Систематичні огляди Susanto et al. [17] та Kell et al. [24] аналізують вплив ML-систем на якість медичних рішень та визначають ключові виклики впровадження. Pumplun et al. [16] пропонують підхід Design Science для створення пояснюваних ML-систем у клінічній практиці. Дослідження Gao et al. [23] демонструє інтеграцію графів знань з великими мовними моделями для покращення діагностичних передбачень.

Окремий напрям складають роботи, присвячені відповідальному впровадженню AI у клінічну практику. Labkoff et al. [18] формулюють рекомендації щодо AI-enabled CDSS, а Giebel et al. [21] досліджують перспективи інтеграції таких систем з точки зору різних стейкхолдерів. Ці роботи звертають увагу на питання пояснюваності, робастності та довіри користувачів до діагностичних систем.

Таким чином, аналіз літератури засвідчує відсутність архітектурного патерну, який би комплексно адресував специфіку систем інтерактивної діагностики: ефективне звуження великого простору рішень, асиметричну обробку підтверджень та заперечень, гарантії збереження правильної відповіді серед кандидатів, та механізми захисту від помилок моделі та користувача. Ця прогалина обумовлює необхідність розробки нового архітектурного патерну, представленого у даній роботі.

Мета і задачі дослідження

Метою дослідження є розробка та формалізація архітектурного патерну для систем інтерактивної діагностики, що забезпечує ефективне звуження простору рішень з асиметрич-

ною обробкою відповідей, гнучким вибором стратегій та гарантіями збереження правильної відповіді у списку кандидатів.

Для досягнення поставленої мети в роботі передбачається розв'язання низки взаємопов'язаних задач. Насамперед здійснюється аналіз існуючих підходів до побудови систем інтерактивної діагностики та методів звуження простору гіпотез з метою виявлення їх обмежень. Далі обґрунтовується концепція асиметричної обробки відповідей користувача та формалізується її вплив на ефективність діагностичного процесу. На цій основі розробляється архітектурний патерн *Iterative Refinement Funnel* із визначенням його структурних компонентів, логіки взаємодії та принципів функціонування.

У межах дослідження також передбачається реалізація механізму вибору діагностичних питань на основі взаємозамінних стратегій із забезпеченням їх гнучкої інтеграції, а також синтез комплексу механізмів робастності для роботи в умовах неповної та неточної інформації. Особливу увагу приділено забезпеченню формальних гарантій збереження правильної гіпотези у процесі ітеративного звуження множини кандидатів. Завершальним етапом є проведення експериментальної валідації запропонованого підходу, оцінювання його ефективності за показниками точності та обчислювальної складності, а також порівняння з існуючими методами.

Результати дослідження

У теорії програмної інженерії прийнято розрізняти патерни проектування та архітектурні патерни за масштабом їхнього впливу на систему [1], [8], [9]. Патерни проектування, такі як описані у класичній роботі *Gamma* та співавторів [1], адресують локальні задачі на рівні окремих класів чи модулів: як створити об'єкт, як організувати взаємодію між двома компонентами, як інкапсулювати алгоритм. Архітектурні патерни натомість визначають глобальну структуру програмної системи, встановлюючи загальні принципи організації та взаємодії великих підсистем [8], [9].

Iterative Refinement Funnel належить саме до категорії архітектурних патернів. Він визначає не спосіб реалізації окремої функції, а цілісну концепцію побудови системи інтерактивної діагностики. Патерн встановлює чотири шари архітектури: *Funnel Layer* для попередньої фільтрації простору варіантів, *Ranking Layer* для точного ранжування кандидатів, *Interaction Layer* для організації діалогу з користувачем, та *Control Layer* для керування процесом та критеріями зупинки. Ці шари утворюють неподільну архітектурну цілісність, зміна якої потребує повного перепроектування системи.

Принципова відмінність архітектурного патерну від патерну проектування полягає у вартості зміни рішення [9], [11]. Якщо заміна патерну *Strategy* на *State* у межах одного модуля потребує локального рефакторингу, то перехід від архітектури *IRF* до альтернативної архітектури означає переписування системи з нуля. Це пояснюється тим, що архітектурний патерн визначає структуру коду, модель даних, протоколи взаємодії компонентів, стратегію масштабування та підходи до тестування.

IRF як архітектурний патерн використовує всередині себе класичні патерни проектування [1]. Зокрема, вибір стратегії питань реалізовано через патерн *Strategy*, управління фазами діагностичного циклу — через патерн *State*, а координацію компонентів — через *Template Method*. Ці патерни проектування є інструментами реалізації архітектурного рішення, а не його сутністю.

Архітектурний патерн *IRF* не обмежується медичною діагностикою та може бути застосований у будь-якій предметній області, де необхідно ефективно ідентифікувати правильне рішення серед великої кількості варіантів через інтерактивний діалог з користувачем.

У сфері технічної підтримки інформаційних систем патерн дозволяє організувати процес *troubleshooting*, де простір варіантів складається з сотень типів несправностей, а ознаками виступають симптоми проблеми, які повідомляє користувач. Асиметрична обробка відповідей є особливо релевантною в цьому контексті: підтвердження конкретного симптому суттєво звужує простір пошуку, тоді як заперечення має менший діагностичний вплив.

Юридична класифікація становить ще один перспективний домен застосування. Система може допомагати класифікувати позови або визначати релевантні прецеденти на основі ознак

справи. Простір варіантів у цьому випадку охоплює тисячі юридичних категорій та прецедентів, а процес уточнення відбувається через послідовні питання про обставини справи.

Рекомендаційні системи з інтерактивним підбором також можуть бути побудовані на архітектурі IRF. На відміну від традиційних рекомендаційних систем, що базуються на історії взаємодій, система на основі IRF активно опитує користувача для швидкого звуження простору товарів чи контенту до найбільш релевантних варіантів.

Експериментальна валідація патерну на системі медичної діагностики Dr.Case продемонструвала його ефективність: досягнуто точність Top-1 на рівні 83–84% при 150-кратному прискоренні порівняно з класичним підходом Expected Information Gain. Ці результати підтверджують практичну цінність запропонованого архітектурного рішення та його готовність до застосування в інших предметних областях.

Представимо опис патерну у форматі каталогу Gang of Four [1], що забезпечує стандартизований спосіб документування архітектурних рішень.

a) Назва та класифікація

Назва: Iterative Refinement Funnel (IRF)

Класифікація: Поведінковий архітектурний патерн

Також відомий як: Asymmetric Diagnostic Loop, YES-Restart Pattern

b) Призначення (Intent)

Організувати ефективний ітеративний процес звуження простору рішень у системах інтерактивної діагностики. Забезпечити асиметричну обробку відповідей користувача, де підтвердження (YES) ініціює повний перерахунок моделі, а заперечення (NO) дозволяє продовжити поточну фазу без витратних обчислень. Підтримати гнучку заміну стратегій вибору питань та надати механізми робастності.

c) Мотивація (Motivation)

Розглянемо типову задачу діагностики: система має визначити правильну відповідь з-поміж D можливих варіантів на основі S ознак через інтерактивний діалог. Класичний підхід EIG обчислює для кожної ознаки очікуване зменшення ентропії, що потребує $O(S \times D \times \log D)$ операцій на вибір одного питання – неприйнятно для інтерактивних застосувань.

Спостереження за реальними сесіями виявляє важливу закономірність: коли користувач підтверджує ознаку, це часто радикально змінює ранжування – модель може висунути на перше місце зовсім іншу гіпотезу. Натомість, заперечення рідко призводить до суттєвих змін – поточний лідер зазвичай залишається на першому місці.

Ця асиметрія підказує архітектурне рішення: виконувати дорогі обчислення лише при YES, а при NO – продовжувати з поточним станом. Додатково, замість глобального пошуку оптимального питання можна використовувати локальні евристики, прив'язані до поточної TOP-1 гіпотези

d) Застосовність (Applicability)

Патерн слід застосовувати, коли: необхідно знайти правильну відповідь серед великої кількості варіантів ($D > 100$) через інтерактивне опитування; відповіді YES та NO мають принципово різну інформаційну цінність; обчислювальна складність EIG є неприйнятною; потрібна гнучкість у виборі стратегії залежно від контексту; важливо забезпечити гарантії Candidate Recall $\geq 99,5\%$

e) Структура (Structure)

Архітектура організована у вигляді функціональних шарів. FUNNEL LAYER відповідає за розбиття простору варіантів на кластери (SpacePartitioner) та забезпечення гарантій покриття (CandidateGuarantor). RANKING LAYER виконує точне ранжування у межах кластера (PrecisionRanker) з механізмом кешування (RankerCache). QUESTION SELECTION реалізує вибір стратегії питань через класичний патерн Strategy [1]. ANSWER PROCESSING інкапсулює асиметричну логіку обробки. ORCHESTRATION координує взаємодію компонентів.

f) Учасники (Participants)

На рис. 1 представлено діаграму компонентів архітектури патерну.

CycleController: Центральний оркестратор, що координує взаємодію компонентів та керує життєвим циклом діагностичної сесії.

QuestionSelector (інтерфейс): Визначає контракт для стратегій вибору питань: `select()` для вибору наступного питання, `reset()` для скидання стану при RESTART.

ClusterHeuristic: Стратегія, що послідовно питає про ознаки поточної TOP-1 гіпотези, потім про ознаки її кластера. Ефективна коли TOP-1 чітко домінує.

DifferentialHeuristic: Стратегія, що питає про ознаки, які відрізняють TOP-1 від TOP-2. Ефективніша коли TOP-1 \approx TOP-2.

AsymmetricAnswerProcessor: Інкапсулює асиметричну логіку: YES \rightarrow FULL_RESTART (інвалідація кешу, перерахунок), NO \rightarrow CONTINUE_PHASE.

SpacePartitioner: Розбиває простір варіантів на кластери схожих елементів.

PrecisionRanker: Виконує точне ранжування варіантів у межах кластера.

RankerCache: Кешує результати ранжування з інвалідацією при YES.

g) Взаємодія (Collaborations)

На рис. 2 представлено діаграму послідовності, що ілюструє взаємодію компонентів.

Основний момент взаємодії — асиметрична реакція на відповіді. При NO кеш залишається валідним, система продовжує поточну фазу без витратних обчислень. При YES відбувається інвалідація кешу, повний перерахунок ранжування та скидання фази стратегії.

h) Наслідки (Consequences)

Преваги:

Радикальне прискорення (до 150x) завдяки складності $O(1)$ замість $O(S \times D \times \log D)$. Економія обчислень завдяки виклику `ranker` лише при YES. Гнучкість через Strategy Pattern – можливість динамічної зміни стратегії. Гарантія Candidate Recall $\geq 99,5\%$. Захист від помилок через механізми робастності.

Компроміси:

Втрата 1-2% точності порівняно з EIG. Збільшена кількість питань (хоча загальний user experience покращується завдяки миттєвій реакції). Залежність від якості початкового ранжування.

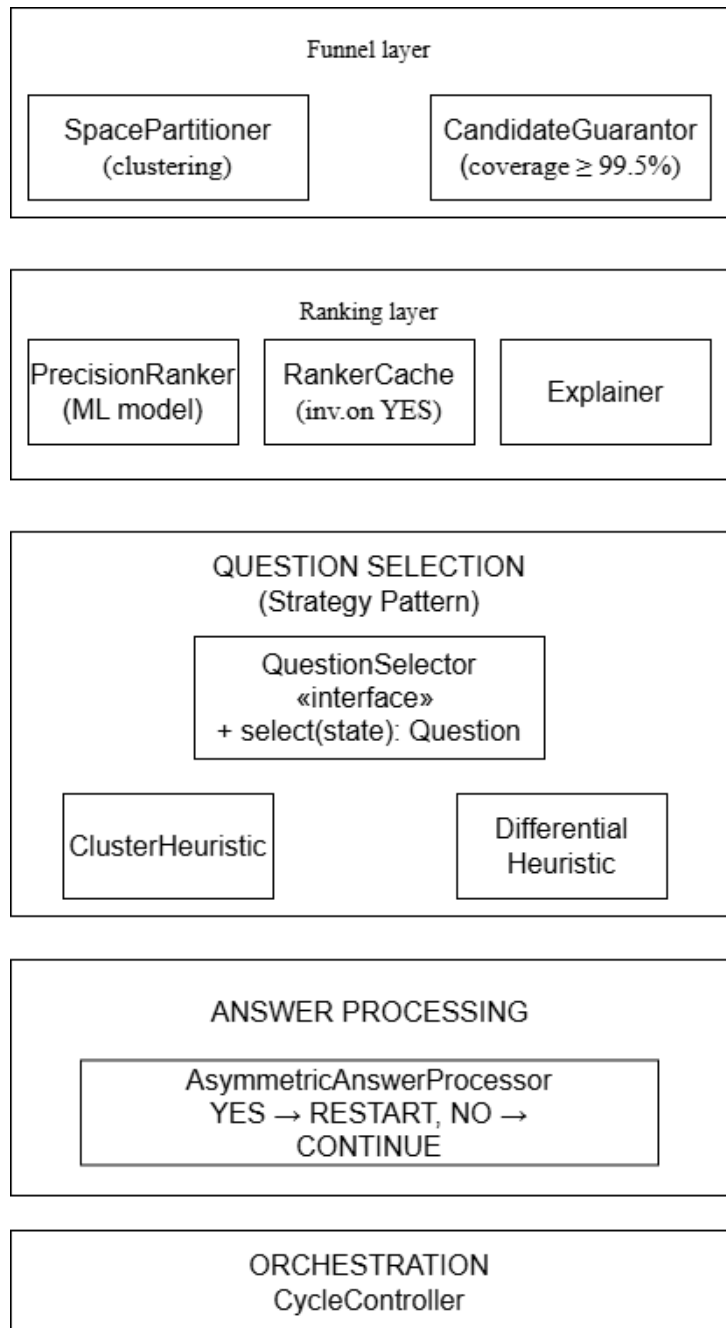


Рис. 1. Діаграма компонентів архітектури патерну IRF

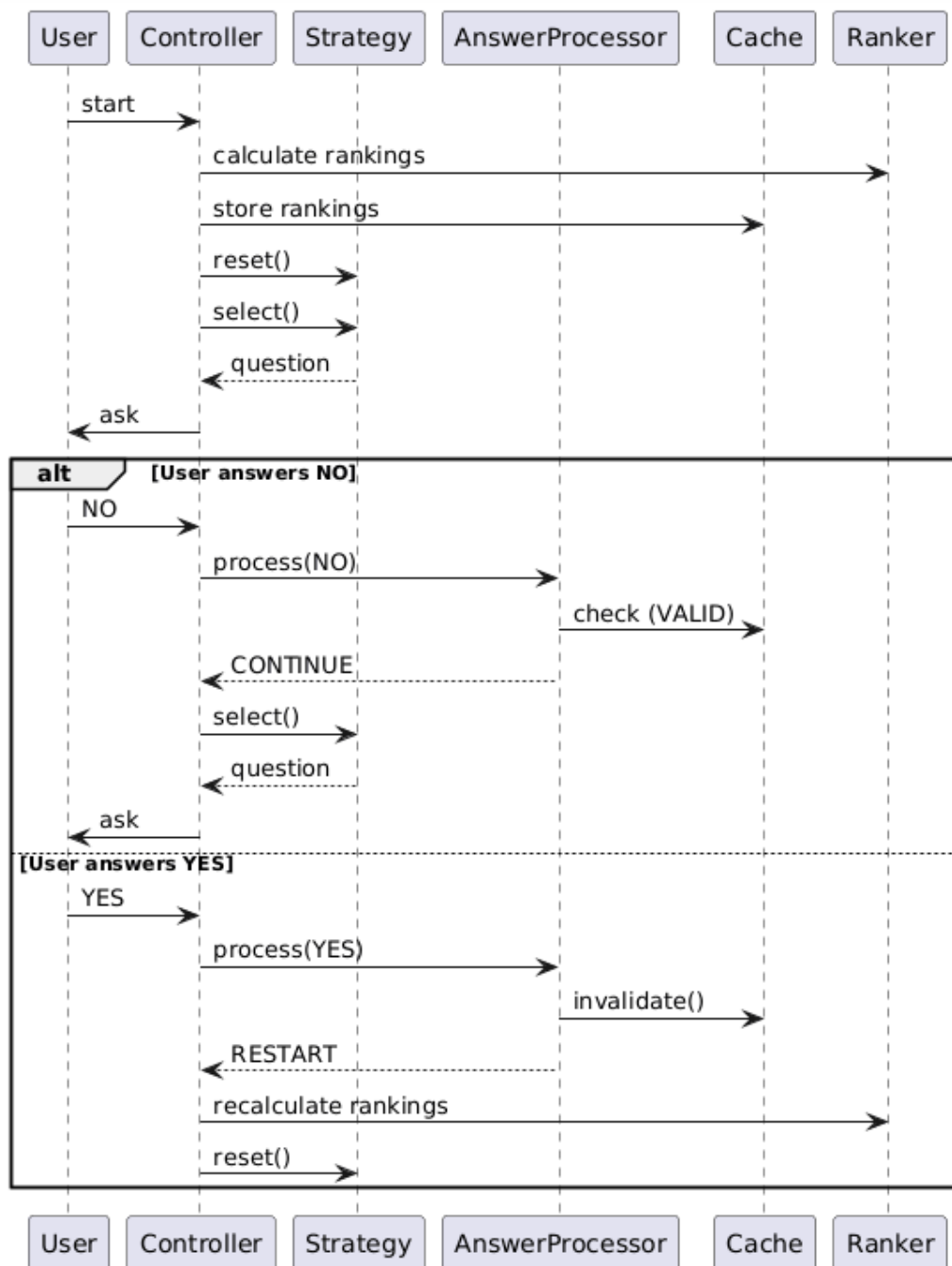


Рис. 2. Діаграма послідовності

i) Пов'язані патерни (Related Patterns)

Strategy [1]: IRF використовує Strategy для інкапсуляції взаємозамінних алгоритмів вибору питань.

State [1]: Кожна стратегія внутрішньо використовує State для управління фазами.

Template Method [1]: CycleController визначає скелет алгоритму, делегуючи конкретні кроки до стратегій.

Cache-Aside: RankerCache реалізує Cache-Aside з явною інвалідацією при YES.

j) Стратегії вибору питань

Cluster Heuristic базується на кластерній структурі простору варіантів. Основна ідея — систематичне дослідження найбільш релевантної області: спочатку ознаки TOP-1 гіпотези, потім ознаки її кластера, потім перехід до наступної гіпотези.

На рис. 3 представлено діаграму станів.

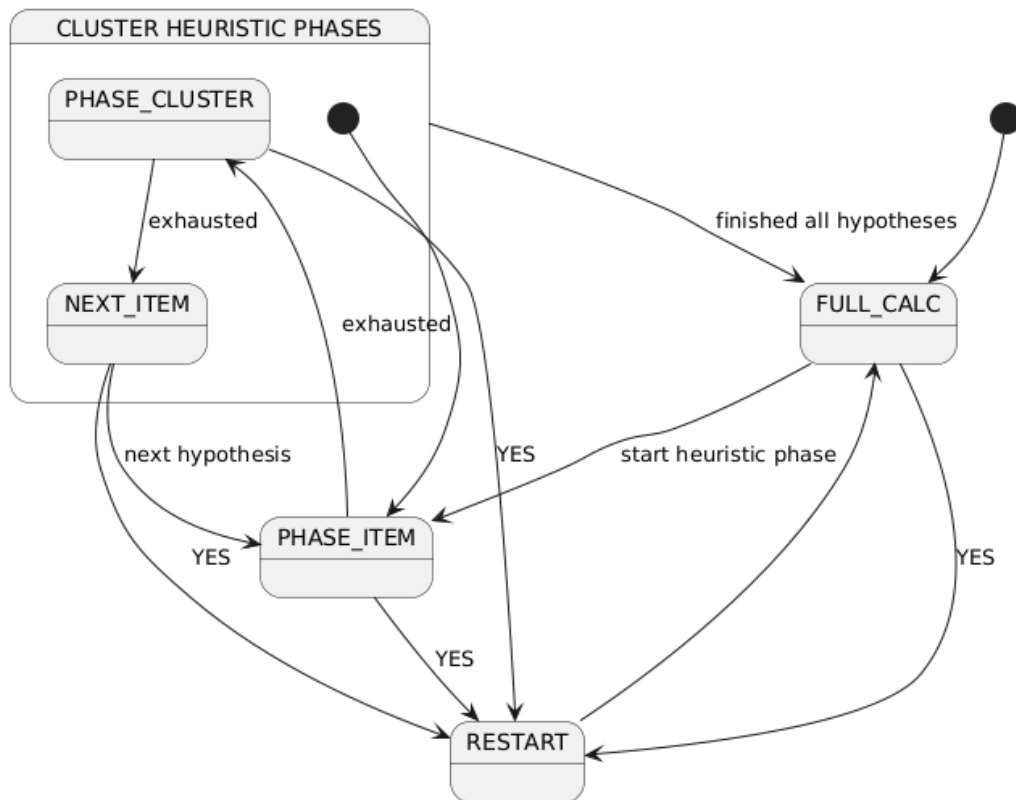


Рис. 3. Діаграма станів Cluster Heuristic

При отриманні YES на будь-якому етапі відбувається RESTART: інвалідація кешу, перерахунок ранжування, скидання фази. При NO система продовжує поточну фазу.

Differential Heuristic фокусується на розрізненні близьких гіпотез. Замість дослідження всіх ознак, стратегія питає лише про ознаки, що відрізняють TOP-1 від TOP-2.

На рис. 4 представлено діаграму станів.

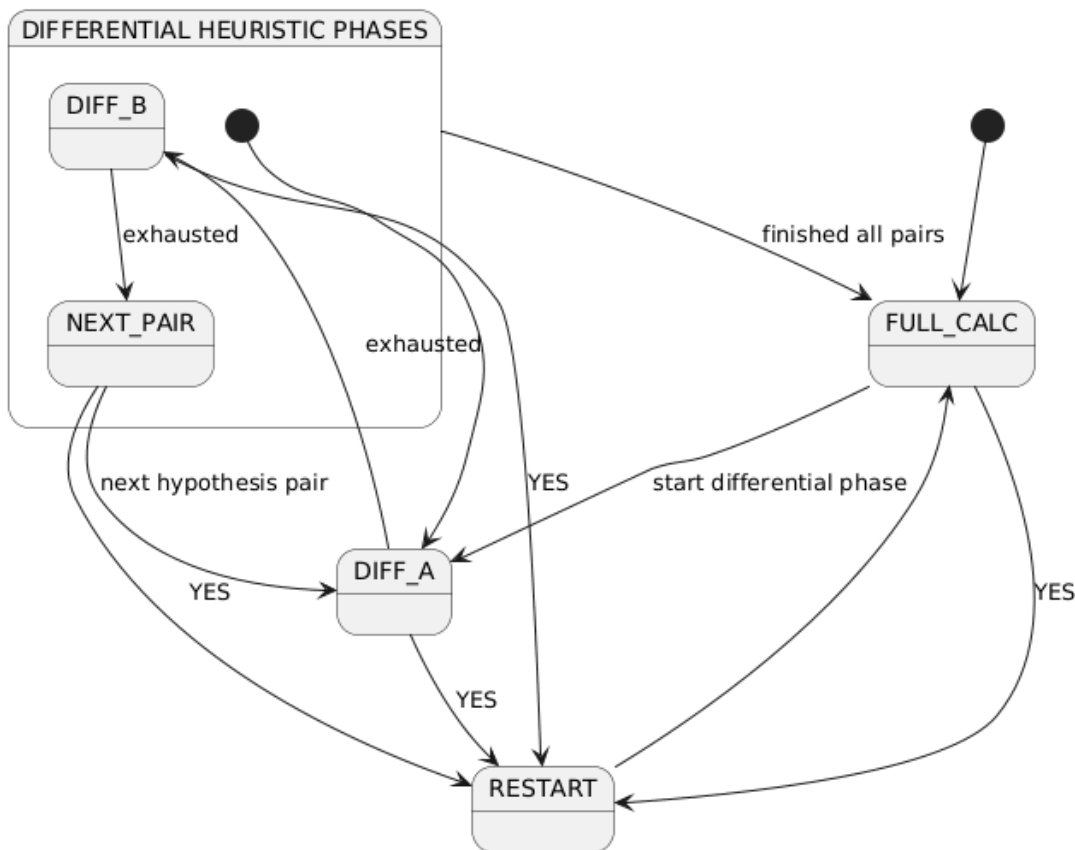


Рис. 4. Діаграма станів Differential Heuristic

Базова асиметрія YES→RESTART, NO→CONTINUE має обмеження у граничних випадках. Патерн передбачає три захисні механізми.

Critical NO

Не всі NO однаково слабкі. Заперечення критичної ознаки (з низьким coverage у кластері) має бути оброблене як сильний сигнал. Якщо coverage < tcrit (типово 30%), система тригерить RESTART замість CONTINUE.

Exploration Question

Якщо модель помилилася на початку і поставила хибну гіпотезу на перше місце, система може «закопатися» в хибну гілку. Після N послідовних NO (типово N=5) система вставляє глобальне питання про ознаку з високим coverage, що не належить поточному кластеру. Це дозволяє «вистрибнути» з локального мінімуму.

Soft Differential

Differential Heuristic чутлива до помилок користувача. Soft Differential вимагає мінімум 2 NO для спростування гіпотези, що захищає від випадкових помилок.

На рис. 5 представлено порівняння стратегій.

На рис. 6 представлено узагальнену діаграму обробки відповідей.

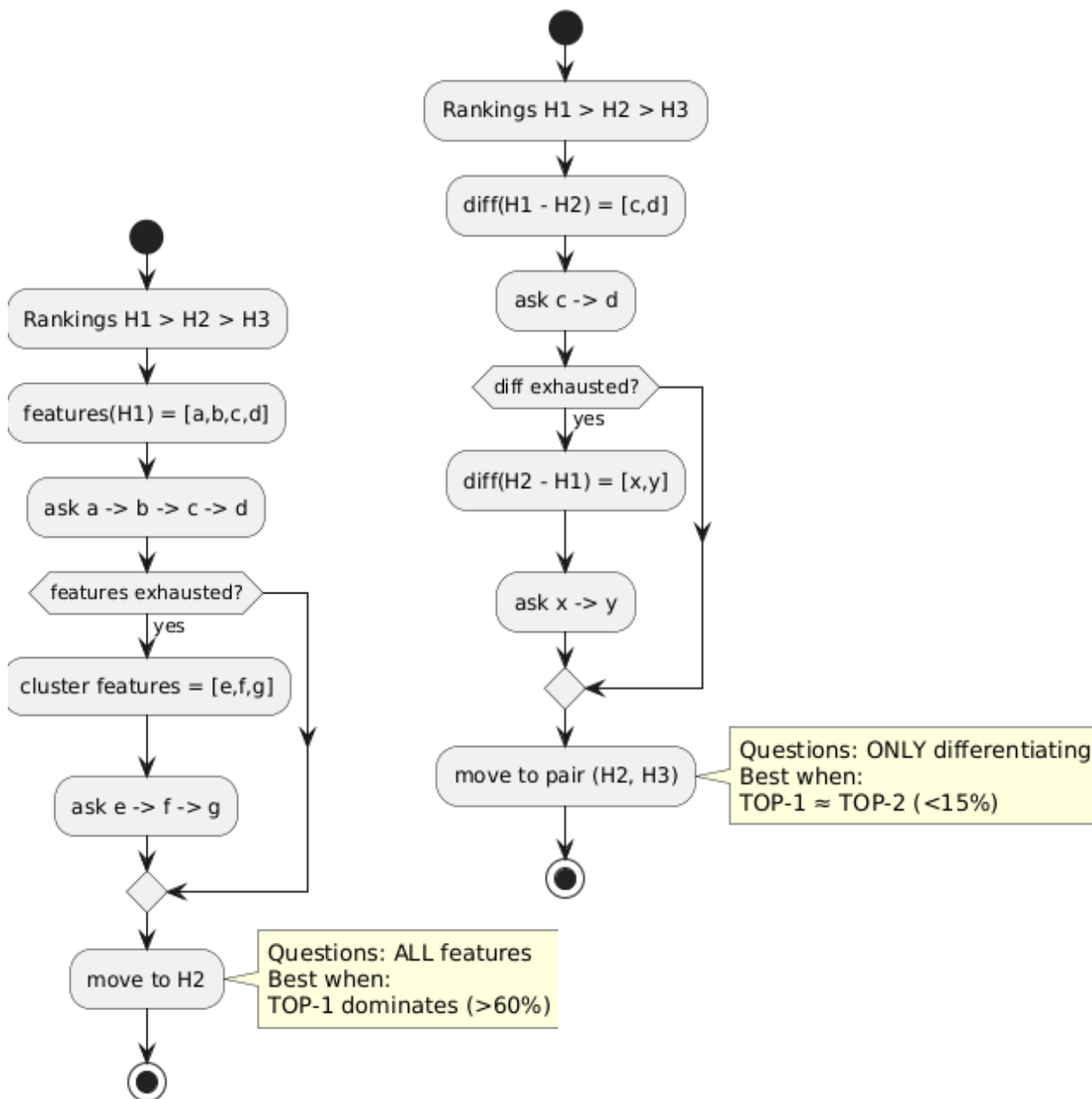


Рис. 5. Порівняння Cluster та Differential Heuristic

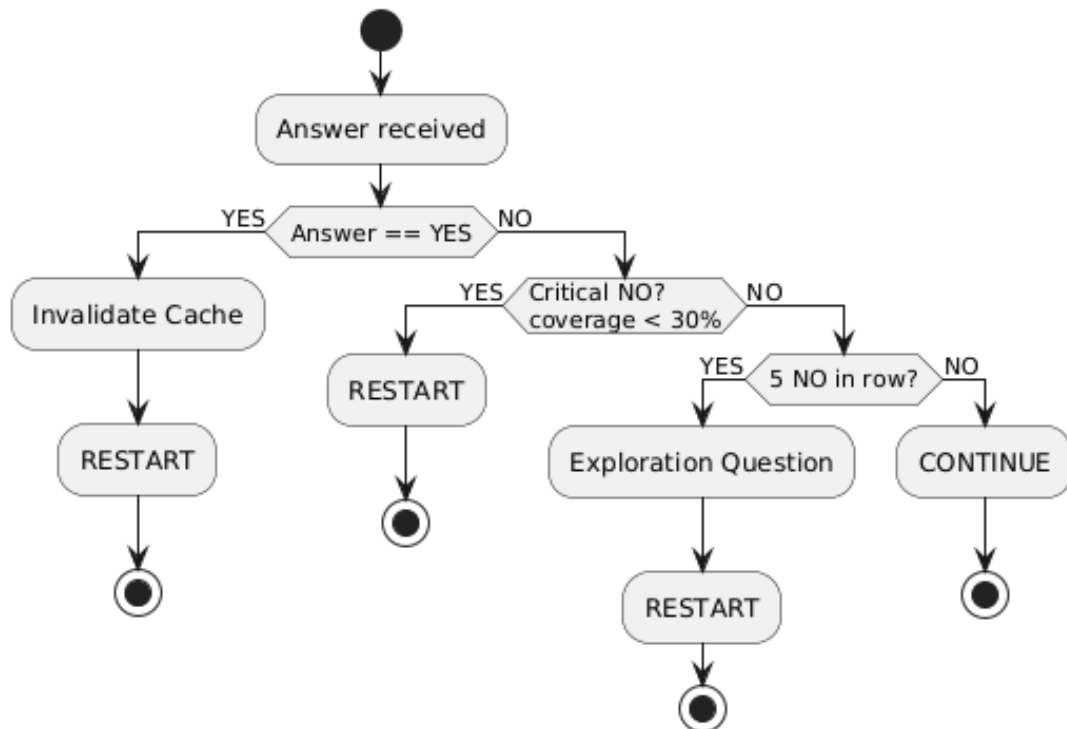


Рис. 6. Діаграма обробки відповідей з механізмами робастності

к) Реалізація

У цьому розділі наведено ключові фрагменти коду, що ілюструють реалізацію основних компонентів патерну.

Контракт стратегії вибору питань визначає два методи: `select()` для вибору наступного питання та `reset()` для скидання стану при `RESTART`.

```

from abc import ABC, abstractmethod
from typing import Optional
class QuestionSelector(ABC):
    """Інтерфейс для стратегій вибору питань."""
    @abstractmethod
    def select(self, state: SessionState) -> Optional[str]:
        """Обрати наступне питання або None якщо вичерпано."""
        pass
    @abstractmethod
    def reset(self) -> None:
        """Скинути стан при RESTART."""
        pass
  
```

Основна особливість патерну – різна обробка `YES` та `NO` відповідей.

```

class AsymmetricAnswerProcessor:
    """Асиметрична обробка: YES → RESTART, NO → CONTINUE."""
    def process(self, answer: bool, symptom: str,
                state: SessionState) -> RestartAction:
        if answer: # YES
            state.confirmed.add(symptom)
            state.invalidate_cache()
            return RestartAction.FULL_RESTART
        else: # NO
            state.denied.add(symptom)
            if self._is_critical(symptom, state):
  
```

```

return RestartAction.FULL_RESTART
return RestartAction.CONTINUE
Контролер реалізує патерн Template Method, координуючи взаємодію компонентів.
class CycleController:
    """Оркестратор діагностичного циклу."""
    def run_cycle(self, initial_symptoms: Set[str]) -> Result:
        state = SessionState(confirmed=initial_symptoms)
        state.rankings = self.ranker.predict(state)
        self.selector.reset()
        while not self.stopping.should_stop(state):
            question = self.selector.select(state)
            if question is None:
                break
            answer = self.get_user_answer(question)
            action = self.processor.process(answer, question, state)
            if action == RestartAction.FULL_RESTART:
                state.rankings = self.ranker.predict(state)
                self.selector.reset()
        return Result(diagnosis=state.top_hypothesis)

```

l) Теоретичні засади та аналіз

Класичний EIG визначається як різниця між поточною ентропією та очікуваною ентропією після отримання відповіді. Принципова відмінність IRF – відмова від Байєсівського оновлення при NO. При YES множина відомих ознак оновлюється, і модель генерує повністю нове ранжування. При NO ранжування не оновлюється. Оскільки IRF не використовує Байєсівське оновлення, EIG не є релевантним для даної архітектури, що дозволяє замінити $O(S \times D \times \log D)$ на $O(1)$.

У табл. 1 наведено порівняння обчислювальної складності.

Таблиця 1

Порівняння обчислювальної складності

Операція	Класичний EIG	Патерн IRF
Вибір питання	$O(S \times D \times \log D)$	$O(1)$
Для $S=500, D=1000$	~5 000 000 операцій	~1 операція
Час на питання	100–500 мс	<1 мс
Виклики ranker	Кожну ітерацію	Тільки при YES

m) Експериментальна валідація

Для валідації ефективності патерну проведено експерименти на системі медичної діагностики Dr.Case, побудованій на архітектурі IRF. База даних містить 844 захворювання та 460 симптомів. Тестування проведено на 1000 випадків з типовою неповнотою інформації (30–50% симптомів) та шумом.

Таблиця 2

Результати експериментального порівняння

Метрика	Global EIG	IRF Cluster	IRF Differential
Top-1 Accuracy	85,1%	83,3%	84,1%
Candidate Recall	99,7%	99,5%	99,5%
Середня к-ть питань	4,2	11,8	9,2
Час на питання	~150 мс	<1 мс	<1 мс
Прискорення	1x	~150x	~150x

Результати демонструють, що патерн IRF забезпечує 150-кратне прискорення при втраті точності лише 1-2%. Candidate Recall $\geq 99,5\%$ гарантує присутність правильної відповіді у списку кандидатів.

У таблиці 3 наведено рекомендації щодо вибору стратегії.

Таблиця 3

Рекомендації щодо вибору стратегії

Умова	ClusterHeuristic	DifferentialHeuristic
TOP-1 conf > 60%	Рекомендовано	Надлишково
TOP-1 \approx TOP-2 (gap < 15%)	Неефективно	Рекомендовано
Мінімізація питань	Більше питань	Менше питань

Висновки та перспективи подальших досліджень

У роботі представлено архітектурний патерн Iterative Refinement Funnel (IRF) для систем інтерактивної діагностики. Патерн описано у форматі каталогу Gang of Four [1], що забезпечує стандартизований спосіб документування та полегшує його впровадження.

Основою роботи патерну є принцип асиметричної обробки відповідей: YES тригерить повний перерахунок (RESTART), NO дозволяє продовжити без витратних обчислень (CONTINUE). Це забезпечує 150-кратне прискорення при втраті точності лише 1-2%.

Запропоновано дві взаємозамінні стратегії – Cluster Heuristic та Differential Heuristic – реалізовані через класичний патерн проектування Strategy [1]. Розроблено механізми робастності: Critical NO, Exploration Question, Soft Differential.

Патерн належить до категорії архітектурних патернів, визначаючи глобальну структуру системи інтерактивної діагностики. Він є domain-agnostic та застосовний у медичній діагностиці, IT-підтримці, юридичній класифікації та рекомендаційних системах.

Експериментальна валідація на системі Dr.Case (844 захворювання, 460 симптомів) підтвердила практичну цінність запропонованого рішення: досягнуто точність Top-1 на рівні 83-84% з гарантією Candidate Recall $\geq 99,5\%$.

Подальшим напрямком дослідження є розробка узагальненого патерну.

Внесок авторів

Максим МЕЛЬНИК – збір і перевірка емпіричних даних, програмне забезпечення; Олексій БИЧКОВ – концептуалізація, методика.

Декларація про штучний інтелект

Штучний інтелект не використовувався.

Конфлікт інтересів

Робота виконана відповідно до принципів академічної доброчесності, етичних норм проведення наукових досліджень та вимог редакційної політики щодо запобігання конфлікту інтересів.

Список використаної літератури

1. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design patterns: Elements of reusable object-oriented software*. Addison-Wesley.
2. Shortliffe, E. H., & Buchanan, B. G. (1975). *A model of inexact reasoning in medicine*. *Mathematical Biosciences*, 23, 351–379.
3. Pearl, J. (1988). *Probabilistic reasoning in intelligent systems*. Morgan Kaufmann.
4. MacKay, D. J. C. (2003). *Information theory, inference, and learning algorithms*. Cambridge University Press.
5. Settles, B. (2009). *Active learning literature survey (CS Technical Report 1648)*. University of Wisconsin–Madison.

6. Shannon, C. E. (1948). *A mathematical theory of communication*. *Bell System Technical Journal*, 27, 379–423.
7. Kohonen, T. (2001). *Self-organizing maps (3rd ed.)*. Springer.
8. Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., & Stal, M. (1996). *Pattern-oriented software architecture: A system of patterns*. Wiley.
9. Shaw, M., & Garlan, D. (1996). *Software architecture: Perspectives on an emerging discipline*. Prentice Hall.
10. Englemore, R., & Morgan, T. (1988). *Blackboard systems*. Addison-Wesley.
11. Fowler, M. (2002). *Patterns of enterprise application architecture*. Addison-Wesley.
12. Newman, S. (2015). *Building microservices: Designing fine-grained systems*. O'Reilly Media.
13. Elhaddad, O. M., Alfares, A. A., Alshahrani, S. M., & Bahkali, S. A. (2024). AI-driven clinical decision support systems: An ongoing pursuit of potential. *Cureus*, 16(4), e57728. <https://doi.org/10.7759/cureus.57728>
14. Park, J., Kim, D., Lee, S., Choi, Y., & Kim, H. (2024). XAI-based clinical decision support systems: A systematic review. *Applied Sciences*, 14(15), 6638. <https://doi.org/10.3390/app14156638>
15. Kim, S. H., Lee, J. H., Park, Y. S., et al. (2024). Machine learning-based clinical decision support system for treatment recommendation and overall survival prediction of hepatocellular carcinoma: A multi-center study. *npj Digital Medicine*, 7(4). <https://doi.org/10.1038/s41746-023-00976-8>
16. Pumplun, L., Peters, F., Gawlitza, J. F., & Buxmann, P. (2023). Bringing machine learning systems into clinical practice: A design science approach to explainable machine learning-based clinical decision support systems. *Journal of the Association for Information Systems*, 24(4), 953–979. <https://doi.org/10.17705/1jais.00820>
17. Susanto, A. P., Lyell, D., Widyantoro, B., Berkovsky, S., & Magrabi, F. (2023). Effects of machine learning-based clinical decision support systems on decision-making, care delivery, and patient outcomes: A scoping review. *Journal of the American Medical Informatics Association*, 30(12), 2050–2063. <https://doi.org/10.1093/jamia/ocad180>
18. Labkoff, S., Oladimeji, B., Kannry, J., et al. (2024). Toward a responsible future: recommendations for AI-enabled clinical decision support. *Journal of the American Medical Informatics Association*, 31(11), 2730–2739. <https://doi.org/10.1093/jamia/ocae209>
19. Tripathi, K. (2024). The novel hierarchical clustering approach using self-organizing map with optimum dimension selection. *Health Care Science*, 3, 88–100. <https://doi.org/10.1002/hcs2.90>
20. Javed, A., Rizzo, D. M., Lee, B. S., & Gramling, R. (2024). SOMTimeS: Self organizing maps for time series clustering and its application to serious illness conversations. *Data Mining and Knowledge Discovery*, 38(3), 813–839. <https://doi.org/10.1007/s10618-023-00979-9>
21. Giebel, G. D., Raszke, P., Nowak, H., et al. (2025). Improving AI-based clinical decision support systems and their integration into care from the perspective of experts: Interview study among different stakeholders. *JMIR Medical Informatics*, 13(1), e69688. <https://doi.org/10.2196/69688>
22. Li, S. S., Balachandran, V., Feng, S., et al. (2024). MEDIQ: Question-asking LLMs and a benchmark for reliable interactive clinical reasoning. *Advances in Neural Information Processing Systems*, 37, 28858–28888.
23. Gao, Y., Li, R., Croxford, E., et al. (2025). DR.KNOWS: Leveraging medical knowledge graphs into large language models for diagnosis prediction: Design and application study. *JMIR AI*, 1, e58670. <https://doi.org/10.2196/58670>
24. Kell, G., Roberts, A., Ferrari, D., et al. (2024). Question answering systems for health professionals at the point of care: A systematic review. *Journal of the American Medical Informatics Association*, 31(4), 1009–1024. <https://doi.org/10.1093/jamia/ocae015>

O. Bychkov, M. Melnyk

ITERATIVE REFINEMENT FUNNEL: AN ARCHITECTURAL PATTERN FOR INTERACTIVE DIAGNOSTIC SYSTEMS WITH ASYMMETRIC RESPONSE PROCESSING

*The article focuses on the development and formalization of a novel architectural pattern, *Iterative Refinement Funnel* (IRF), intended for the design of interactive diagnostic systems. The pattern addresses the task of efficiently identifying the correct solution among a large set of alternatives under conditions of incomplete and incrementally acquired information. The proposed approach is based on the concept of asymmetric response processing, which accounts for the observed difference in informational value between confirmation (YES) and negation (NO). A confirmed feature triggers a full recomputation of the machine learning model with an updated projection onto a self-organizing map and subsequent re-ranking of hypotheses by a neural network, whereas a negative response allows continuation of the current diagnostic phase without incurring computationally expensive updates.*

*The paper proposes two interchangeable question selection strategies—*Cluster Heuristic* and *Differential Heuristic*—implemented via the Strategy design pattern, ensuring adaptability to diverse diagnostic scenarios and input data characteristics. A set of robustness mechanisms is introduced, including *Critical NO* for handling negation of pathognomonic features, *Exploration Question* for overcoming local minima effects, and *Soft Differential* for mitigating the impact of noisy or imprecise user responses.*

*Experimental validation on a medical dataset comprising 844 diseases and 460 unique symptoms demonstrates a Top-1 accuracy of 83.3–84.1%, alongside a 150-fold speedup compared to the classical *Expected Information Gain* approach. Furthermore, it is shown that asymmetric response processing significantly reduces computational costs without degrading diagnostic quality, which is important for real-time systems. The proposed pattern is domain-independent and can be integrated into various intelligent systems, particularly in decision support tasks requiring efficient interactive narrowing of the hypothesis space. Additionally, the formalization of the pattern enables its future automated optimization and extension within hybrid artificial intelligence architectures.*

Keywords: architectural pattern, interactive diagnostics, machine learning, Self-Organizing Map, asymmetric processing, Expected Information Gain, system robustness, clinical decision support.

Надійшла до редакції: 17.03.2026

Прийнята до друку: 05.05.2026

Опубліковано: 29.06.2026

© 2026 О. С. Бичков, М. О. Мельник.

Цей матеріал ліцензовано за умовами CC BY 4.0. <https://creativecommons.org/licenses/by/4.0/>