

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КІЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО”

ЛАБОРАТОРНА РОБОТА

з дисципліни

“Гібридні інтелектуальні системи”

за темою

**“ПОШУК ПРОГРАМНИХ НЕСПРАВНОСТЕЙ В РОБОТІ
ОРГТЕХНІКИ (СКАНЕР, ПРИНТЕР)”**

Виконала:
студентка гр. ТР-41МН
Ольга Титаренко

Київ 2025

ЗМІСТ

<u>ПОСТАНОВКА ЗАДАЧІ ТА МЕТА РОБОТИ</u>	3
<u>АРХІТЕКТУРА СИСТЕМИ</u>	3
<u>ОБГРУНТУВАННЯ ВИБОРУ МЕХАНІЗМУ ЛОГІЧНОГО ВИСНОВКУ</u>	4
<u>МАТЕМАТИЧНА МОДЕЛЬ ЛІНГВІСТИЧНИХ ЗМІННИХ</u>	4
<u>ОРГАНІЗАЦІЯ БАЗИ ЗНАНЬ ТА СТРАТЕГІЯ ПОШУКУ</u>	6
<u>ВИСНОВКИ</u>	8

Варіант завдання: 4

Тип системи: нечітка експертна система

ПОСТАНОВКА ЗАДАЧІ ТА МЕТА РОБОТИ

Об'єктом розробки є інтелектуальна система підтримки прийняття рішень, яка призначена для ідентифікації програмних збоїв офісного обладнання (принтерів, сканерів, факсів).

Актуальність розробки зумовлена необхідністю автоматизації процесу діагностики в умовах, коли первинна інформація надходить від користувачів, які не володіють технічною експертizoю. Вхідні дані часто мають суб'єктивний та неточний характер (наприклад, оцінки "повільна робота" чи "погана якість"), що унеможливлює використання класичних детермінованих алгоритмів.

Для **розв'язання** поставленої задачі обрано архітектуру гібридної інтелектуальної системи, що реалізує інтеграцію методів на макрорівні. Система поєднує математичний апарат нечіткої логіки для обробки невизначеності та експертну систему на базі продукційних правил для формалізації евристичних знань.

АРХІТЕКТУРА СИСТЕМИ

Структурна організація ГІС базується на класичній схемі експертної системи, доповненій модулями фазифікації та дефазифікації для роботи з лінгвістичними змінними.

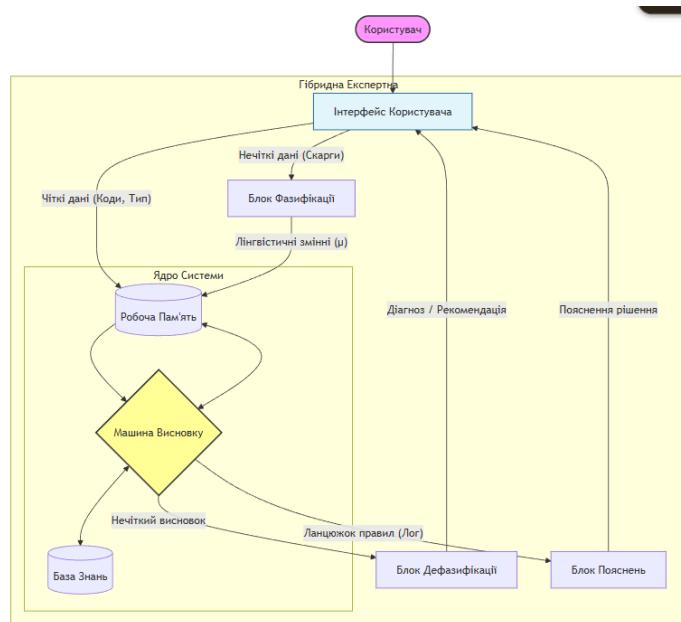


Рис. 1 – Структурна схема взаємодії компонентів ГІС

Інформаційний потік у системі організовано наступним чином: інтерфейс користувача забезпечує введення як чітких даних (коди помилок, тип пристрою), так і нечітких оцінок. Модуль фазифікації трансформує вхідні числові параметри у функції належності μ . Отримані нечіткі факти зберігаються у робочій пам'яті. Центральним ядром системи є машина логічного висновку, яка двосторонньо взаємодіє з базою знань та робочою пам'яттю для генерації рішення. Завершальним етапом є дефазифікація, що перетворює нечіткий результат у конкретну рекомендацію для користувача. Важливою частиною є блок пояснень, який на вимогу користувача демонструє трасування логічного виведення.

ОБГРУНТУВАННЯ ВИBORУ МЕХАНІЗМУ ЛОГІЧНОГО ВИСНОВКУ

Для програмної реалізації діагностичного алгоритму обрано прямий механізм висновку. Даний підхід передбачає просування від наявних фактів (симптомів несправності) до мети (діагнозу). Вибір обґрунтовано специфікою задачі: процес перетворення даних відбувається ітеративно доти, доки можлива зміна значень у робочій пам'яті або досягнення термінального стану. На відміну від зворотного висновку, який ефективний для підтвердження гіпотез, прямий висновок є оптимальним для синтезу рішення на основі поточної картини стану системи.

МАТЕМАТИЧНА МОДЕЛЬ ЛІНГВІСТИЧНИХ ЗМІННИХ

Для опису вхідних параметрів системи використовуються лінгвістичні змінні, що задаються кортежем $\langle N, T, U, G, M \rangle$, де

- N - назва змінної,
- T – базова терм-множина,
- U – універсум,
- G – синтаксичне правило, граматика, що породжує нові терми з використанням лінгвістичних модифікаторів (хеджів типу “дуже”, “не”, “більш менш”). Наприклад, для базового терму A правило G дозволяє сформувати терми

$T_{new} = \{"\text{дуже } A", "не A", "більш - менш } A"\}$, що модифікують функцію належності зведенням у ступінь або добуванням кореня.

- M – семантичне правило, яке ставить у відповідність кожному терму нечітку множину на універсумі U через функції належності $\mu(x)$.

Наведемо приклад формалізації змінної “Час відповіді пристрою”, яка характеризує часовий інтервал між надсиланням завдання та початком його виконання. Універсумом U буде часовий проміжок $[0; 120]$ секунд. Задамо терм-множину T (“Миттєво”, “Затримки”, “Тайм-аут”).

Опишемо аналітично функції належності M :

1. Для терма “Миттєво” (Z-подібна функція):

$$M_{Instant}(x) = \begin{cases} 1, & x \leq 5 \\ \frac{15-x}{15-5}, & 5 < x < 15 \\ 0, & x \geq 15 \end{cases}$$

2. Для терма “Затримка” (трикутна функція з параметрами $a = 10, b = 30, c = 60$)

$$M_{Delay}(x) = \begin{cases} 0, & x \leq 10 \\ \frac{x-10}{20}, & 10 < x < 30 \\ \frac{60-x}{30}, & 30 < x < 60 \\ 0, & x \geq 60 \end{cases}$$

3. Для терма “Таймаут” (S-подібна функція)

$$M_{Timeout}(x) = \begin{cases} 0, & x \leq 45 \\ \frac{x-45}{75}, & 45 < x < 120 \\ 1, & x \geq 120 \end{cases}$$

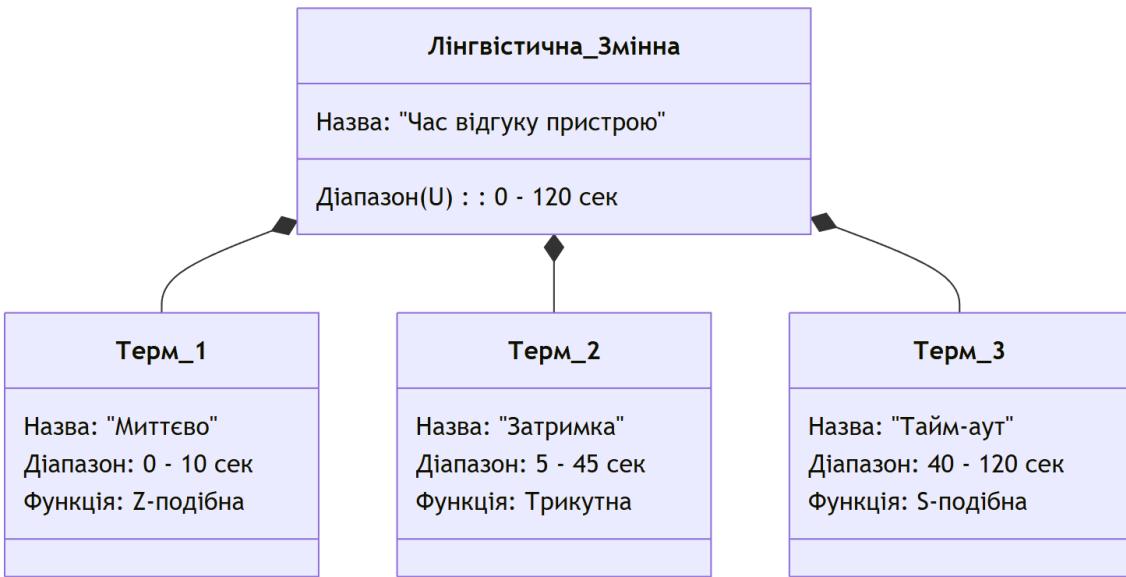


Рис. 2 – Структурна схема лінгвістичної змінної

Аналогічним чином формалізовано змінні “Якість друку” (суб'єктивна шкала з термами “Норма”, “Артефакти”, “Дефект”) та “Завантаженість черги” (кількісна шкала з термами “Вільна”, “Робоча”, “Зависла”).

ОРГАНІЗАЦІЯ БАЗИ ЗНАНЬ ТА СТРАТЕГІЯ ПОШУКУ

База знань системи побудована на основі продукційної моделі представлення знань, яка використовує правила виду: ЯКЩО (передумова) – ТО (дія). Правила враховують гібридну природу вхідних даних, розділяючи чіткі умови (тип обладнання) та нечіткі (суб'єктивні характеристики). Такий підхід забезпечує природність опису причинно-наслідкових зв'язків у діагностиці.

На верхньому рівні відбувається ідентифікація типу пристрою, на середньому - локалізація підсистеми (драйвер, мережа, механіка), на нижньому - визначення конкретної програмної помилки.

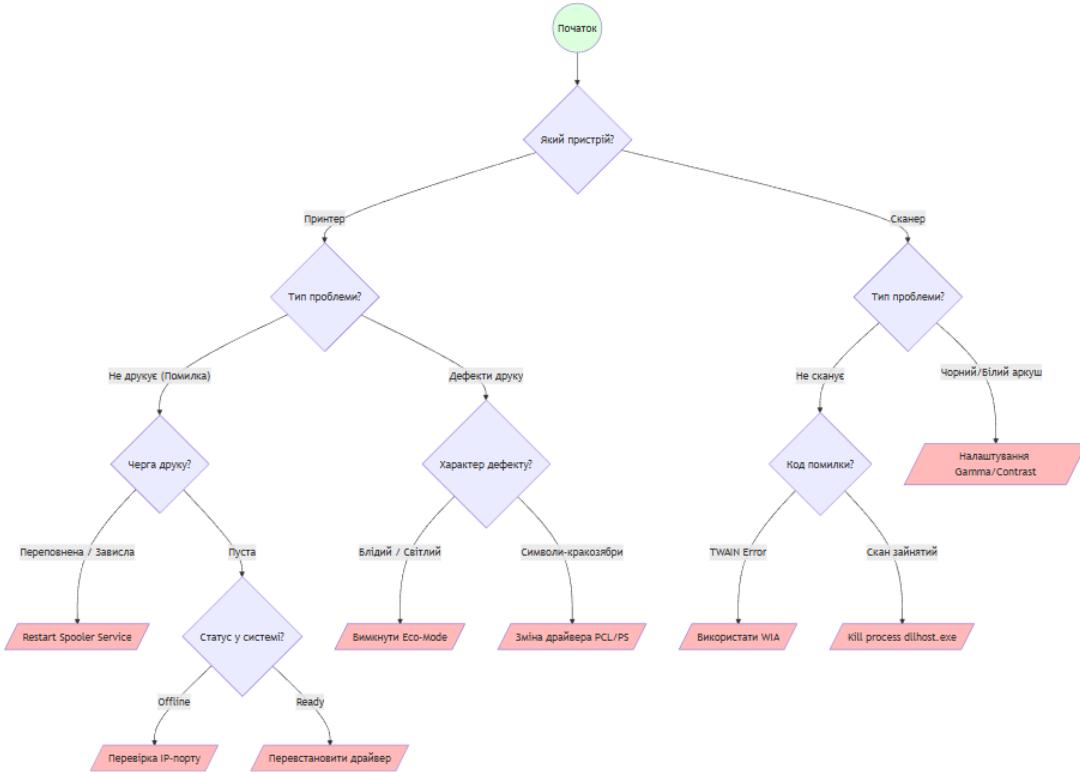


Рис. 3 – Фрагмент дерева рішень для діагностики програмних збоїв

Наведемо приклади продукційних правил бази знань:

- **Правило №1 (Діагностика служби Spooler):**

IF (Пристрій == “Принтер”) AND (Черга друку IS “Переповнена”) AND (Час відгуку IS “Тайм-аут”)

THEN (Діагноз = “критичний збій служби Print Spooler”) з ймовірністю 1.0.

Коментар: Правило спрацьовує при одночасній наявності ознак зависання черги та відсутності реакції принтера.

- **Правило №2 (конфлікт драйверів сканування):**

IF (Пристрій == “Сканер”) AND (З'єднання IS “Стабільне”) AND (Код помилки == “TWAIN_Error”)

THEN (Діагноз = “конфлікт TWAIN-інтерфейсу з ОС”) з ймовірністю 0.9

ELSE (Рекомендація = “Перевстановити драйвер у режимі сумісності”).

- **Правило №3 (проблема передачі даних):**

IF (Пристрій == "Принтер") AND (Якість_друку IS “Норма”) AND (Час відгуку IS “Затримка”)

THEN (Діагноз = “Помилка інтерфейсу передачі даних або мережеві затримки”).

Коментар: правило спрацьовує, коли механічна частина справна (якість у нормі), але спостерігається повільна обробка потоку даних.

ВИСНОВКИ

Розроблений ескізний проект обґруntовує доцільність використання гібридного підходу для побудови дiагностичної системи для виявлення несправностей оргтехніки. Використання лінгвістичних змiнних, визначених кортежем $\langle N, T, U, G, M \rangle$, дозволяє системi оперувати набiженими поняттями, властивими людському мисленню, а прямий механiзм висновку забезпечує ефективний пошук рiшення на основi симптомiв. Подальша реалiзацiя передбачає програмне втiлення запропонованої моделi з використанням бiблiотек нечiткої логiки.