**УДК 004.423**

***Стисло Оксана Василівна****, старший викладач кафедри*

*інформаційних технологій, Заклад вищої освіти*

*«Університет Короля Данила», м. Івано-Франківськ*

***Панчук Павло Степанович,*** *магістр, Заклад вищої освіти*

*«Університет Короля Данила», м. Івано-Франківськ*

## *Кутинський Андрій Ярославович, аспірант, ІФНТУНГ, м. Івано-Франківськ*

## *Дмитрів Ігор Ярославович*, *аспірант, ІФНТУНГ, м. Івано-*

## *Франківськ*

**ОСОБЛИВОСТІ ФРЕЙМВОРКІВ САМОНАЛАШТУВАННЯ СИСТЕМ**

Сучасний світ вимагає від програмних систем здатності до швидкої адаптації, відгуку на змінні обставини та гнучкості у відповіді на нові виклики. Самоналаштовувані системи, які можуть самостійно моніторити, аналізувати та вносити зміни до своєї поведінки без зовнішнього втручання, стають ключовим рішенням для цього завдання. Однією з основних складових таких систем є фреймворки, що надають інструменти та механізми для розробки самоналаштовуваних рішень**[1,2]**. В даному дослідженні розглянуто особливості цих фреймворків, їх переваги та виклики, а також розкрито, як вони можуть сприяти створенню ефективних, надійних та адаптивних програмних систем.

**Дослідження особливостей побудови фреймворку самоналаштовуваної системи**

Постійне ускладнення програмних систем разом з необхідністю забезпечення їх якості обслуговування (QoS) в умовах невизначеності та неблагосприятливого середовища призвело до створення самоадаптивних програмних систем. Самоадаптивні системи можуть самостійно модифікуватися під час роботи для досягнення визначених функціональних цілей чи показників QoS. Однак, розробка таких систем виявилася значно важчою в порівнянні зі стандартними програмними системами. Основна складність полягає у розробці логіки адаптації. Через те що розробники програмного забезпечення часто не можуть передбачити всі потенційні зміни у вимогах та робочому середовищі системи на етапі проектування, вони звертаються до аналітичних моделей. Цей підхід має три основні недоліки:

Складність використання. Багато сучасних самоадаптивних систем вимагають складних аналітичних моделей. Щоб ці моделі були ефективними, їх треба налаштовувати відповідно до особливостей конкретної предметної області. Нерідко потрібно створювати нову модель для кожного нового завдання, що є часомістким процесом.

Помилкові припущення. Аналітичні моделі базуються на спрощеннях або припущеннях щодо системи, які можуть виявитися невірними. Якщо реальні умови відхиляються від цих припущень, аналіз може бути неточним, що призводить до помилок в адаптації.

Проблеми з ефективністю. Швидкість аналізу і планування відіграє велике значення для самоадаптивних систем, які повинні оперативно реагувати на зміни. Однак пошук оптимальної конфігурації може бути обчислювально складним і потребувати значних ресурсів. Очікувана самоадаптивна структура рішення є інструментом адаптації програмних систем. Ця структура дозволяє програмній системі змінюватися відповідно до потреб і умов роботи, а саме - дозволяє вибирати необхідні функції системи на льоту, в залежності від поточних вимог до якості обслуговування (QoS). Головна ідея полягає в тому, щоб адаптувати програмну систему, коли виникають компроміси між різними параметрами QoS, аби намагатися досягти максимального задоволення від її роботи. Наприклад, якщо система TRS (можливо, це конкретний модуль або підсистема) має надмірний час відповіді, фреймворк може адаптувати систему, вибравши нову функцію, яка прискорює відгук, одночасно не порушуючи інших вимог до QoS. Процес адаптації складається з безперервного циклу, який ми можемо назвати "циклом адаптації". Цей цикл включає в себе такі етапи:

Detect (Виявлення): На цьому етапі система аналізує зібрані показники від поточної роботи системи, щоб визначити, чи задовольняє вона поточні цілі QoS. Якщо показники відхиляються від цілей, система переходить до наступного етапу.

Plan (Планування): Тут система шукає нову, оптимальну конфігурацію, яка дозволить максимізувати корисність (задоволеність користувача) і відповідати цілям QoS. Це може включати вибір нових функцій, зміну параметрів системи або відключення непотрібних функцій.

Effect (Впровадження): На цьому кроці система реалізує зміни, вибрані на попередньому етапі. Це може бути активація або деактивація певних функцій, зміна параметрів тощо.

Таким чином, інтелектуальний механізм адаптації, постійно моніторить роботу програмної системи, виявляє відхилення від цілей QoS і вносить необхідні корективи для забезпечення оптимальної роботи системи.

## Висновки

В даному дослідженні оглянуто інноваційні методи розробки самоадаптивних систем, які спрямовані на вирішення ключових проблем, пов'язаних з використанням аналітичних моделей. Відмінно від традиційних методів, які базуються на складних моделях та часто помилкових припущеннях, новітні методи даного класу впроваджують підхід, заснований на онлайн-навчанні, для динамічної адаптації до змінних умов. Центральна ідея полягає у використанні динамічного представлення системи, орієнтованого на функції, для навчання та оптимізації адаптивної поведінки системи.

Література:

1. Weyns, Danny, and Usman M. Iftikhar. "ActivFORMS: A formally founded model-based approach to engineer self-adaptive systems." ACM Transactions on Software Engineering and Methodology 32.1 (2023): 1-48.

2. Carmichael, Ted, and Mirsad Hadžikadić. The fundamentals of complex adaptive systems. Springer International Publishing, 2019.