

АНАЛІЗ ФАКТОРІВ ВПЛИВУ АВТОМАТИЗАЦІЇ ПІДТРИМКИ ПРОГРАМНИХ КОМПЛЕКСІВ НА БАЗІ КОЛЬОРОВИХ МЕРЕЖ ПЕТРІ

Андрій Пукач¹, Василь Теслюк²

^{1,2}Національний університет “Львівська політехніка”,
кафедра автоматизованих систем управління, Львів, Україна
¹E-mail: andriipukach@gmail.com, ORCID: 0009-0001-8563-3311
²E-mail: vasy1.teslyuk@gmail.com, ORCID: 0000-0002-5974-9310

© Пукач А., Теслюк В. М., 2024

Розроблено модель аналізу факторів впливу автоматизації процесів підтримки програмних комплексів на базі кольорових мереж Петрі, що дає змогу здійснити їхнє моделювання під час розв’язання науково-прикладної задачі відновлення границь факторів впливу суб’єктивного сприйняття об’єктів підтримки з інкапсульованими штучними нейронними мережами типу багат шарового перцептрона. Задача аналізу та відновлення границь факторів впливу входить до переліку задач науково-прикладної проблеми автоматизації підтримки програмних комплексів. Об’єктом дослідження є процес аналізу факторів впливу автоматизації процесів підтримки програмних комплексів. Предметом дослідження є методи та засоби моделювання аналізу факторів впливу автоматизації процесів підтримки програмних комплексів на базі теорії мереж Петрі загалом та кольорових мереж Петрі зокрема. Метою дослідження є розроблення моделі аналізу факторів впливу автоматизації підтримки програмних комплексів на базі кольорових мереж Петрі. Для досягнення поставленої мети розв’язано певні задачі дослідження, подані далі по тексту. Зокрема, подано блок-схему алгоритму аналізу факторів впливу процесів автоматизації підтримки програмних комплексів, а також опис моделі суб’єктивного сприйняття об’єкта підтримки з інкапсульованою штучною нейронною мережею типу багат шарового перцептрона. Також, наведено детальний опис покрокового функціонування розробленої моделі за усіма можливими сценаріями. Додатково побудовано дерево досяжності розробленої моделі, що демонструє досяжність та скінченність кожного із станів представленої моделі на базі кольорових мереж Петрі. Крім того, проведено дослідження динаміки функціонування розробленої моделі на базі кольорових мереж Петрі та подано отримані результати дослідження. А також як приклад аналізу факторів впливу автоматизації процесів підтримки програмних комплексів розв’язано прикладну практичну задачу виявлення домінуючого фактору впливу з набору факторів впливу команди підтримки програмного комплексу.

Ключові слова: автоматизація, програмний комплекс, фактор впливу, підтримка, модель, кольорові мережі Петрі.

Вступ та формулювання проблеми

У наш час ми стаємо свідками значного прогресу технологій в галузі автоматизації найрізноманітніших процесів. Однією з передових галузей автоматизації є сфера інформаційних технологій, де одним із ключових об’єктів досліджень та розробок є, власне, різноманітні програмні комплекси як узагальнене поняття: програмного забезпечення, пов’язаного з ним апаратного забезпечення,

інформаційного забезпечення, ресурсного забезпечення, технологій, архітектур, та абсолютно всіх інших складових компонентів складних програмних комплексів. Особливістю цих комплексів є те, що їх обслуговує величезна кількість людей – професіоналів найрізноманітніших напрямків: від розробників до менеджерів із продажів. Тож виникають цілком логічні та передбачувані питання дослідження можливостей автоматизації цих процесів, яке з огляду на свою нетривіальність вимагає комплексних рішень.

Одним із актуальних напрямків автоматизації під час роботи з програмними колесами є напрям їхньої підтримки. До того ж йдеться не лише про клієнтську підтримку, а й про інші активності довкола підтримуваного програмного компасу, як-от: розроблення, тестування, збірка, випуск, розгортання, підтримка, тощо.

Об'єктом дослідження є процес аналізу факторів впливу автоматизації процесів підтримки програмних комплексів. Предметом дослідження є методи та засоби моделювання аналізу факторів впливу автоматизації процесів підтримки програмних комплексів на базі теорії мереж Петрі загалом та кольорових мереж Петрі (КМП) зокрема. Метою дослідження є розроблення моделі аналізу факторів впливу автоматизації підтримки програмних комплексів на базі кольорових мереж Петрі. Практична цінність роботи полягає в розробленні алгоритму аналізу факторів впливу автоматизації процесів підтримки програмних комплексів для побудови та функціонування моделей аналізу факторів такого впливу на базі кольорових мереж Петрі. Наукова новизна роботи полягає у розробленні моделі аналізу факторів впливу автоматизації підтримки програмних комплексів на базі кольорових мереж Петрі, що дає змогу досліджувати процеси аналізу факторів впливу для автоматизації підтримки програмних комплексів.

Аналіз останніх досліджень та публікацій

Серед основних напрямків автоматизації підтримки програмних комплексів виділимо такі ключові напрямки як: автоматизація реєстрації та опрацювання внутрішніх та зовнішніх звернень, автоматизація тестування, автоматизація DevOps, що фактично представляють базу та лівову частку автоматизації підтримки програмних комплексів в цілому на сьогодні.

Автоматизацію реєстрації та опрацювання звернень можна умовно категоризувати на дві складові, де основним критерієм категоризації є походження звернення. У разі походження звернень із боку клієнтів, замовників або зовнішніх користувачів маємо справу із зовнішнім типом звернень, а в разі створення звернень безпосередньо всередині компаній-розробників програмних комплексів такі звернення реєструються як внутрішні. Прикладом автоматизації зовнішніх звернень є, наприклад, автоматизовані бази знань, автоматизовані чат-боти та автоматизовані системи розпізнавання людської мови, що використовуються в автоматизованих лініях телефонної підтримки. Зокрема, в роботі [11] представлено приклад такої бази знань, що дає змогу забезпечувати автоматизовану підтримку користувачів на початковому етапі виникнення звернень, вивільняючи в такий спосіб значну частину ресурсів клієнтської підтримки. В роботі [12] розглянуто основні тренди розвитку автоматизації із застосуванням чат-ботів, відповідно до яких 80 % власників бізнесів у майбутньому планують використовувати чат-боти. У статті [13] подано комплексний аналіз проблеми та обмеження в технологіях розпізнавання мовлення, критичний огляд алгоритмів, інструментів і систем опрацювання мовних сигналів. Прикладом автоматизації внутрішніх звернень є різноманітні системи трекінгу, сортування та адресації звітів про помилки в роботі підтримуваних програмних комплексів. Зокрема, авторами роботи [14] представлено автоматизовану систему адресації звітів про помилки у функціонуванні підтримуваного програмного комплексу найбільш відповідним розробникам, що дає змогу скоротити сумарний час вирішення таких звернень та мінімізувати ризик адресації звернень до неправильних (невідповідних) адресантів, адже останнє приводить в подальшому до необхідності повторної переадресації та значної втрати як часу, так і ресурсів розробників та іншого персоналу підтримки програмних комплексів. У роботі [15] подано

порівняльний аналіз інструментів відстеження помилок, що дає загальне розуміння основних принципів їхнього призначення та функціонування.

Автоматизація тестування програмних комплексів є, мабуть, одним із найперших напрямків автоматизації процесів довкола програмних комплексів. Так, наприклад, ще в далекому 1999-му році вийшла робота [2], в якій доволі детально розглянуто цей напрямок автоматизації. Водночас сьогодні 2024-го року на прикладі роботи [9] ми бачимо, що надзвичайно активно обговорюються питання дедалі ширшого та глибшого провадження технологій штучного інтелекту в цьому напрямку. Використання технологій штучного інтелекту фактично свідчить про наступний етап еволюції цього напрямку – тестування програмних комплексів та його подальший перехід від автоматизації до часткової або повної автоматичності.

Надалі автоматизація тестування зазнала неабиякого прогресу, втіленого в такі різновиди, як: інструменти автоматизованого тестування програмного забезпечення з відкритим вихідним кодом для інтелектуальної розроблення програмного забезпечення [16]; автоматизація наскрізного вебтестування [22–23]; автоматизація генерації тестових даних на основі алгоритму багатоцільової оптимізації мурашиного лева (eng. antlion optimization algorithm – ALO) [17], алгоритму EvoPSO [20], алгоритму NEAT [21]; інтеграція відгуків користувачів у автоматизоване тестування додатків мобільної платформи Android [18]; автоматизація тестів для гібридних мобільних додатків [19] та завершуючи найрізноманітнішими проявами застосування технологій штучного інтелекту у сфері автоматизації тестування, описаних в численних працях, серед яких, як приклад, відзначимо праці [24–28].

Питаннями автоматизації процесів розроблення, збірки, випуску та доставки активно займається напрямок DevOps. Одними із найбільш комплексних робіт в цьому напрямку є, зокрема, роботи [1, 3, 6, 7], які, мабуть, більш ніж повною мірою розкривають тематику DevOps як невід’ємної складової сучасного підходу автоматизації процесів, безпосередньо пов’язаних з процесами розроблення та операцій (ориг. “development and operations”).

Формулювання цілі статті

Як бачимо з проведеного аналізу наявних досліджень за тематикою роботи, однією з актуальних ключових науково-прикладних проблем сьогодення є проблема автоматизації процесів підтримки програмних комплексів, що складається з цілої низки відповідних науково-прикладних задач. Виокремимо, власне, задачу аналізу факторів, що впливають на суб’єктивне сприйняття об’єкта підтримки (підтримуваного програмного комплексу, чи процесів його підтримки) відповідними суб’єктами взаємодії з цим об’єктом. І навіть попри широкий асортимент наявних напрацювань у сфері автоматизації підтримки програмних комплексів, саме ця конкретна задача вимагає додаткової уваги та дослідження. Тож у цьому напрямку виникає необхідність розроблення алгоритмів, моделей, методів та засобів, які б давали змогу дослідження та розв’язання цієї науково-прикладної задачі. Однією з таких моделей є якраз представлена в цій роботі модель аналізу факторів впливу автоматизації процесів підтримки програмних комплексів, розроблена на базі кольорових мереж Петрі [4, 5, 8].

Отже, ціль статті полягає у висвітленні розробленої моделі аналізу факторів впливу автоматизації процесів підтримки програмних комплексів на базі кольорових мереж Петрі, а також усіх супутніх задач дослідження, необхідних для досягнення мети дослідження, серед яких, зокрема: необхідність подання блок-схеми алгоритму аналізу факторів впливу автоматизації процесів підтримки програмних комплексів, а також опис моделі суб’єктивного сприйняття об’єкта підтримки з інкапсульованою штучною нейронною мережею типу багат шарового перцептрона; необхідність наведення детального опису покрокового функціонування розробленої моделі за усіма можливими сценаріями; необхідність побудови дерева досяжності розробленої моделі та проведення досліджень динаміки функціонування розробленої моделі, а також демонстрації прикладу аналізу факторів впливу автоматизації підтримки програмних комплексів під час розв’язання конкретної прикладної практичної задачі.

Виклад основного матеріалу

Фактори впливу. Для дослідження факторів впливу, що вносять відповідні корективи на результативне сприйняття об'єкта підтримки суб'єктами взаємодії з цим об'єктом, був розроблений окремий підхід, повний та детальний опис якого, на жаль, виходить за межі цієї роботи.

Основною платформою для дослідження факторів впливу є модель суб'єктивного сприйняття об'єкта підтримки, що репрезентує процес суб'єктивного перетворення (трансформації, інтерпретації) вхідних характеристик об'єкта підтримки (як самого підтримуваного програмного комплексу, так і процесів його підтримки) у результуючі вихідні характеристики суб'єктивного сприйняття цього об'єкта відповідним(-и) суб'єктом(-ами) взаємодії з ним.

Модель суб'єктивного сприйняття може бути представлена в будь-якій довільній (як канонічній, так і в казуальній) формі. Проте основними та обов'язковими вимогами до таких моделей є:

- наявність чітких, конкретних, визначених вхідних характеристик досліджуваного об'єкта (підтримуваного програмного комплексу, чи процесів його підтримки);
- наявність чітких, конкретних, визначених факторів впливу, що впливають на результати сприйняття цього об'єкта відповідними суб'єктами взаємодії з ним;
- наявність чітких, конкретних, визначених вихідних характеристик, що забезпечують однозначну інтерпретацію результатів сприйняття об'єкта підтримки – відповідним суб'єктом взаємодії з ним (об'єктом);
- модель повинна забезпечувати відображення (інтерпретацію) усіх реальних ключових процесів щодо трансформації вхідних характеристик – у відповідні вихідні результати, через пряму взаємодію факторів впливу.

Наступним етапом є додаткова інкапсуляція в модель суб'єктивного сприйняття – штучної нейронної мережі (ШНМ) в узагальненому випадку – будь-якого існуючого типу, проте в такому частковому досліджуваному та репрезентованому випадку ми розглядатимемо конкретний тип – багатошаровий перцептрон (БП).

Багатошаровий перцептрон, зокрема, описаний в роботі [10]. Він доволі активно використовується в усіх напрямках застосування технології штучного інтелекту. Володіє як деякими перевагами, так і деякими недоліками. Проте в цьому конкретному випадку основною перевагою саме багатошарового перцептрона є його абсолютна структурна відповідність запропонованій моделі суб'єктивного сприйняття, відповідно до якої:

- нейрони вхідного шару БП відповідають вхідним характеристикам об'єкта;
- приховані шари БП відображають фактори впливу;
- нейрони вихідного шару БП відповідають результатам суб'єктивного сприйняття об'єкта.

Та навіть попри ці беззаперечні переваги, все ж є один нюанс при інкапсуляції саме БП у модель суб'єктивного сприйняття. Цей нюанс полягає в розмитті границь факторів впливу. Тобто ми в жодному разі не можемо інтерпретувати кожен окремий прихований шар БП як певний відповідний окремий фактор впливу.

Сама ідея БП полягає в тому, що приховані шари БП (на відміну від вхідного та вихідного шарів) не несуть жодного функціонально-сислового навантаження в отриманій кінцевій мережі типу БП. Адже вони були створені та призначені лише для коректності математично-обчислювальних процесів навчання та функціонування загалом, розробленої мережі БП, проте аж ніяк не для інтерпретації факторів, суб'єктів, функцій, змісту тощо.

Отже, інкапсулюючи штучну нейронну мережу типу БП у модель суб'єктивного сприйняття об'єкта підтримки, ми фактично так втрачаємо і границі факторів впливу, і саму ідентичність факторів впливу.

Саме тому з метою повернення ідентичності факторів впливу, а також можливості відновлення їхніх границь, запропоновано та розроблено відповідний алгоритм аналізу факторів впливу (блок-схема та опис якого будуть подані нижче в цій роботі), а також відповідна модель аналізу

факторів впливу на базі кольорових мереж Петрі, що дає змогу динаміку та надійність процесів на ранніх етапах системного проектування.

Нижче на рис. 1 подано приклад реальної розробленої досліджуваної моделі суб'єктивного сприйняття об'єкта підтримки з інкапсульованої штучною нейронною мережею типу багатoshарового перцептрона.

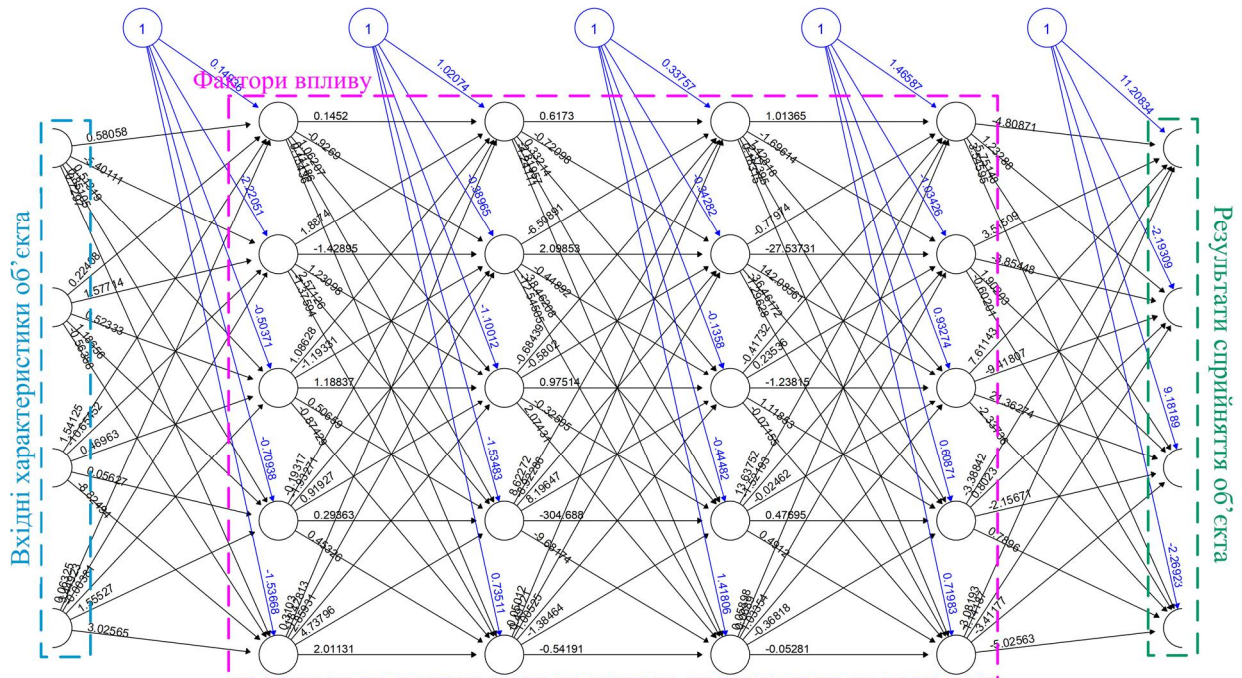


Рис. 1. Приклад моделі суб'єктивного сприйняття об'єкта підтримки з інкапсульованої штучною нейронною мережею типу багатoshарового перцептрона

Алгоритм аналізу факторів впливу. Першим етапом роботи розробленого алгоритму є створення моделі (як вже було вказано вище – в довільній формі) суб'єктивного сприйняття об'єкта підтримки. Далі відбувається розроблення відповідної штучної нейронної мережі (в цьому випадку ми розглядаємо тип мережі – БП), яка буде інкапсульована в саму модель. Після чого здійснюється навчання інкапсульованої мережі БП. За ним слідує етап детермінації всіх наявних факторів впливу. А далі – підготовка (або виокремлення) ізольованих вибірок даних для кожного окремого фактору впливу. На цьому етапі надзвичайно важливо, щоб дані були максимально ізольовані для кожного окремо взятого конкретного фактору впливу, тобто ми готуємо дані для цих ізольованих вибірок при абсолютній домінації окремо взятого поточного фактору впливу над усіма іншими факторами – це надзвичайно важлива обов'язкова умова для подальшої успішної роботи всього процесу аналізу факторів впливу. Після цього відбувається тестування навченої інкапсульованої мережі БП на отриманих на попередньому кроці ізольованих вибірках даних кожного із детермінованих доміантних факторів впливу. Отримані результати тестувань БП на ізольованих вибірках даних проходять додаткове очищення на наступному кроці алгоритму. Принцип очищення надзвичайно простий та єдиний – ми залишаємо до подальшого розгляду та досліджень лише ті результати, де результат тестування навченої БП збігається з очікуваним результатом, натомість результати, де отриманий результат тестування БП, не збігається з очікуваним, ми повністю відкидаємо та не приймаємо до подальшого розгляду взагалі. Очищення також є обов'язковим етапом для коректної роботи всього подальшого процесу аналізу факторів впливу. Слідом за цим етапом відбувається збереження очищених результатів тестувань для ізольованих вибірок даних кожного із факторів впливу.

Наступним етапом роботи розробленого алгоритму є перебір усіх отриманих збережених очищених результатів тестувань з подальшим:

- формуванням зворотних ланцюжків максимальних ваг (від активного нейрона вихідного шару БП – назад через усі приховані шари БП від останнього прихованого шару і аж до першого – і далі до відповідного нейрона вхідного шару БП);

- маркування нейронів (прихованих шарів БП), що входять в отримані на попередньому етапі зворотні ланцюжки максимальних ваг, маркерами відповідних факторів впливу, до яких належать ці ланцюжки.

Фінальним етапом роботи алгоритму є розрахунок імовірностей приналежності нейронів прихованих шарів до факторів впливу, що є фактично відновленням границь факторів впливу, втрачених в результаті інкапсуляції БП.

Нижче на рис. 2 зображено блок-схему алгоритму аналізу факторів впливу автоматизації підтримки програмних комплексів.

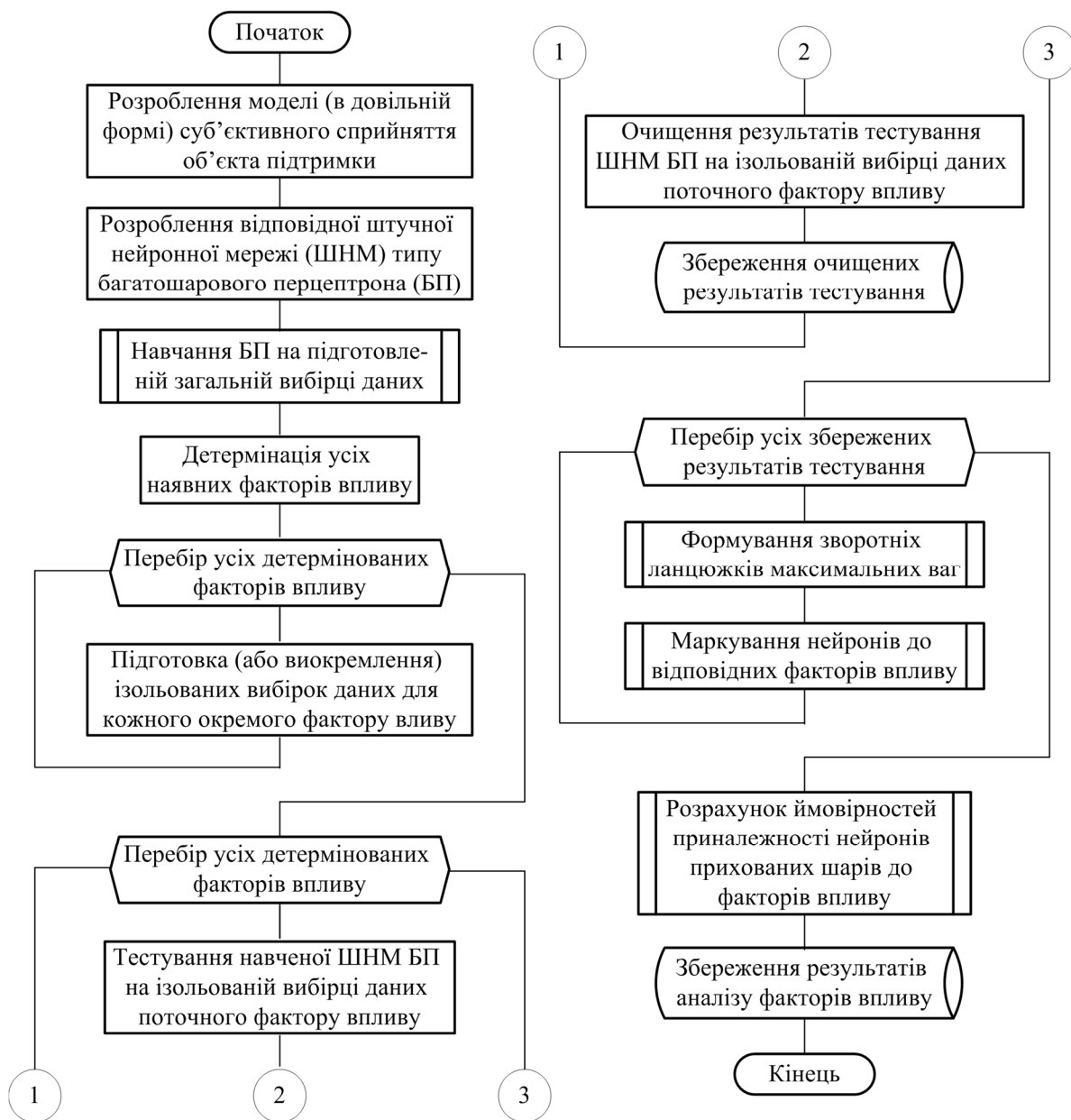


Рис. 2. Блок-схема алгоритму аналізу факторів впливу автоматизації підтримки програмних комплексів

Модель на базі кольорових мереж Петрі. Нижче на рис. 3 представлена розроблена модель аналізу факторів впливу автоматизації підтримки програмних комплексів на базі кольорових мережі Петрі.

Модель функціонує строго відповідно до розробленого алгоритму та розпочинає свою роботу з активації переходу T1, що репрезентує процес розроблення моделі суб'єктивного сприйняття об'єкта підтримки, після чого основний маркер одразу ж потрапляє в позицію P2, що відображає стан підготовки до інкапсуляції ШНМ БП, та відбувається подальша активація переходу T3, яка представляє процес детермінації всіх наявних факторів впливу. В результаті спрацювання переходу T3 відбувається наповнення позицій P5 та P8 різнокольоровими маркерами, кожен з яких репрезентує відповідний окремий детермінований фактор впливу.

Паралельно з цим основний маркер, помножуючись у n разів (де n – це кількість цих детермінованих факторів впливу), потрапляє в позицію P4, що відповідає за стан підготовки ізольованих вибірок даних.

Після цього відбувається поступове (порційне) вивільнення по одному маркеру з позиції P4, так і по одному відповідному маркеру фактору впливу з позиції P5, що щоразу призводить до активації переходу T4, який відображає процес формування ізольованої вибірки даних кожного відповідного окремого фактору впливу, та збереження кожної отриманої таким чином ізольованої вибірки даних у відповідній позиції P6.

Процес формування ізольованих вибірок даних завершується саме тоді, коли абсолютно всі тригеруючі маркери в позиції P4 остаточно вичерпані, і в цій позиції не залишилося жодного маркера. Тож відбувається активація переходу T5, що репрезентує процес завершення формування ізольованих вибірок даних.

Одразу після цього, в результаті спрацювання переходу T5, відбувається помноження основного маркера в n разів із потраплянням цієї множини результуючого добутку основних маркерів у відповідну позицію P7, що відповідає за стан підготовки навченої ШНМ БП до тестувань на отриманих ізольованих вибірках даних.

Далі відбувається порційне виокремлення по одному маркеру з позицій: P7 – що виконує функцію основного тригера; P8 – що зберігає набір декомпонованих факторів впливу; та P6 – що містить відповідні ізольовані вибірки даних, сформовані на попередньому етапі. Водночас щоразу відбувається спрацювання переходу T6, що демонструє процес тестування навченої ШНМ БП на кожній ізольованій вибірці даних кожного відповідного декомпонованого фактору впливу. А отримані результати тестувань акумулюються в позиції P9, що репрезентує наступний функціональний стан – очищення результатів тестувань. Звідки активацією переходу T7 очищені результати зберігаються та акумулюються у відповідній позиції P10.

Цикл тестувань БП на ізольованих вибірках даних завершується одночасно із закінченням основних тригеруючих маркерів в позиції P7. Тож у цей самий момент стає можливим спрацювання наступного переходу – T8, що відображає процедуру формування зворотних ланцюжків максимальних ваг на основі результатів тестувань, попередньо вже в повному обсязі акумульованих множиною різнокольорових маркерів у позиції P10, звідки ці маркери по одному надходять на вхід вказаного переходу T8, активуючи його щоразу.

Сформовані зворотні ланцюжки максимальних ваг акумулюються в позиції P11, звідки по одному активують перехід T9, ініціюючи щоразу процедуру маркування нейронів маркерами відповідних факторів впливу.

Результати роботи описаних вище процедур, а саме: процедури формування зворотних ланцюжків максимальних ваг, та процедури маркування нейронів маркерами факторів впливу, акумулюються у відповідній позиції P12.

Цикл опрацювання результатів тестувань БП у контексті описаних процедур завершується із завершенням маркерів у відповідній позиції P10, котра якраз і містила ці попередньо акумульовані туди результати тестувань.



Рис. 3. Модель аналізу факторів впливу автоматизації підтримки програмних комплексів на бази кольорових мережі Петрі (продовження)

Отож як тільки в позиції P10 не залишилося жодного маркера, то стає можливим спрацювання наступного переходу T10, що запускає процедуру розрахунку ймовірностей приналежності нейронів прихованих шарів БП до факторів впливу, вхідними даними для якої є результати, попередньо акумульовані в описаній вище позиції P12. Отримані в такий спосіб результати аналізу факторів впливу акумулюються в позиції P13.

Водночас як тільки всі результати з позиції P12 будуть опрацьовані, стає можливим активація останнього переходу T11, що сигналізує нам про фактичне звершення роботи всієї моделі, та перехід основного маркера у фінальну позицію P14.

Отже, на виході моделі ми отримуємо результати аналізу факторів впливу, репрезентовані та акумульовані в позиції P13.

Варто відзначити, що розроблена модель є узагальненою інтерпретацією процесів аналізу факторів впливу без багатьох додаткових деталізацій, як-от: деталізований процес навчання БП; деталізований процес підготовки або виокремлення ізольованих вибірок даних; деталізований процес опрацювання форматів вхідних даних (CSV, XML, DAT, та інших); деталізований процес формування зворотних ланцюжків максимальних ваг; деталізований процес маркування нейронів прихованих шарів маркерами відповідних факторів впливу; деталізований процес розрахунку ймовірностей приналежності нейронів прихованих шарів БП до відповідних факторів впливу тощо.

Деталізація кожного з цих процесів вимагає розроблення окремих додаткових моделей на базі мереж Петрі.

Проте в згаданому конкретному випадку нас цікавить саме дослідження глобального процесу аналізу факторів впливу автоматизації підтримки програмних комплексів саме на системному рівні, а не на по-компонентному, по-елементному чи інших можливих рівнях деталізації. Тож саме з цієї точки зору була вибрана такий ступінь деталізації процесів, як подано у відповідному розробленому алгоритмі аналізу факторів впливу та подальшій розробленій та представленій моделі аналізу факторів впливу на базі кольорових мереж Петрі.

Дерево досяжності. Отже, розробивши та представивши відповідну модель аналізу факторів впливу на базі кольорових мереж Петрі, залишається остаточний етап фіналізації отриманої моделі, що полягає в розробленні відповідного дерева досяжності.

Нижче на рис. 4 представлено дерево досяжності для розробленої моделі аналізу факторів впливу автоматизації процесів підтримки програмних комплексів на базі кольорових мереж Петрі.

Дерево досяжності, або граф досяжності станів мереж Петрі – це класичне поняття теорії мереж Петрі. Дерево є графом, вершинами якого є бінарна матриця усіх наявних в побудованій мережі Петрі станів, а гілками – відповідно, усі наявні в побудованій мережі Петрі переходи. Основною задачею дерева досяжності (незалежно від типу мережі Петрі) є потактовий аналіз функціонування розробленої мережі Петрі з індексацією динаміки зміни станів мережі, а також запобігання тупикових ситуацій, коли мережа зациклоється в певному стані або ж завчасно припиняє своє функціонування в певному стані, не доходячи до кінцевого стану логічного завершення функціонування мережі. Також дерево досяжності демонструє скінченність кожного із станів розробленої мережі Петрі. У разі розробленої мережі Петрі, представленій в цій роботі, ми маємо такі параметри мережі:

- $P = \{P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P11, P12, P13, P14\}$ – масив позицій;
- $T = \{T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11\}$ – масив переходів.

Отож представлене дерево досяжності демонструє перехід як основного робочого маркера мережі, так і його передбачених можливих мультиплікацій, між абсолютно всіма задекларованими позиціями з масиву позицій P мережі через відповідні задекларовані переходи з масиву переходів T – від розміщення маркера в початкову позицію запуску мережі P1 і аж до досягнення ним кінцевої позиції завершення функціонування мережі P14.

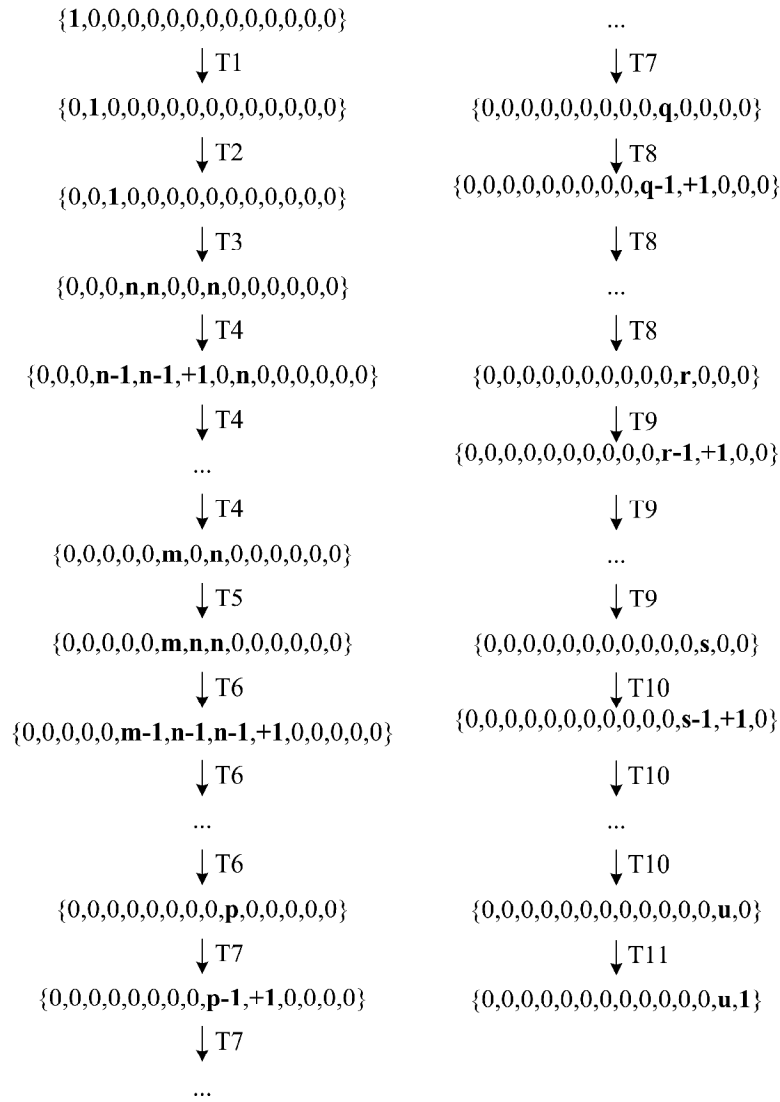


Рис. 4. Дерево досяжності станів розробленої моделі аналізу факторів впливу автоматизації процесів підтримки програмних комплексів на базі кольорових мереж Петрі

Аналіз результатів досліджень динаміки розробленої моделі. Особливістю моделей на базі кольорових мереж Петрі (як і, в принципі, будь-яких мереж Петрі) є те, що вони дають змогу досліджувати надійність та динаміку проєктованих процесів ще на ранніх (початкових) етапах системного проєктування. Дослідження динаміки процесів надзвичайно важливе для синхронізації ресурсів та зусиль у разі паралельного опрацювання багатопотокових даних у режимі реального часу. Також такі дослідження дають важливу інформацію щодо проблемних ділянок моделювання, які потребують додаткової оптимізації ресурсів. У разі застосування комп'ютерного моделювання також обов'язково слід враховувати технічні характеристики конкретного комп'ютерного пристрою, на якому проводиться моделювання, адже від обчислювальної потужності пристрою на пряму залежатимуть і отримані результати динаміки моделювання.

Нижче на рис. 5 подано показники середнього часу спрацювання переходів розробленої моделі, а на рис. 6 наведено отримані результати динаміки моделювання розробленої моделі аналізу факторів впливу на базі кольорових мереж Петрі.

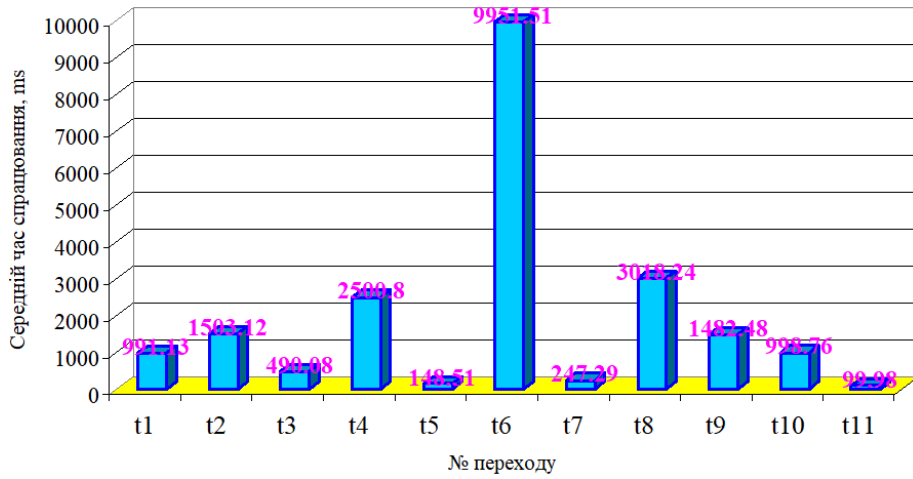


Рис. 5. Порівняльна діаграма середнього часу спрацювання переходів розробленої моделі аналізу на базі кольорових мереж Петрі

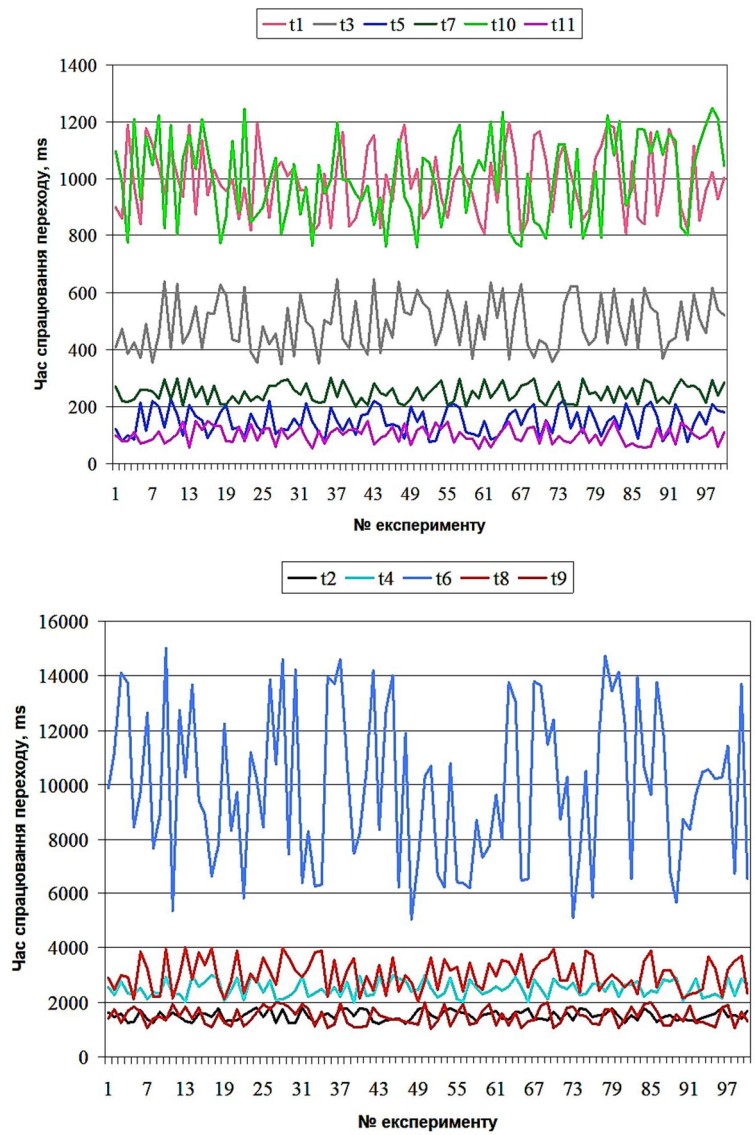


Рис. 6. Графіки динаміки спрацювання переходів розробленої моделі аналізу факторів впливу на базі КМП

Аналіз результатів практичного застосування. У ролі практичного застосування аналізу факторів впливу автоматизації підтримки програмних комплексів розв'язано прикладну задачу виявлення домінуючого фактору впливу з набору факторів впливу команди підтримки програмного комплексу.

Відповідно до алгоритму аналізу факторів впливу, блок-схема якого була подана вище на рис. 2, на виході моделі отримуємо результати розрахунків приналежності нейронів прихованих шарів інкапсульованої ШНМ БП до кожного із факторів впливу. Далі, маючи ці дані, тестуємо модель для кожного із суб'єктів, якими виступають члени команди підтримки, і на основі результатів тестування отримуємо усереднені значення кожного із факторів впливу на результат сприйняття об'єкта підтримки (підтримуваного програмного комплексу) відповідним суб'єктом (членом команди підтримки) взаємодії з ним.

Саме такі усереднені значення впливів кожного з факторів для кожного з членів команди подані нижче в табл. 1.

Таблиця 1

Розподіл факторів впливу для кожного з членів команди підтримки

№ спеціаліста команди підтримки	Фактор 1	Фактор 2	Фактор 3	Фактор 4
1	0.2694	0.1986	0.1681	0.3639
2	0.2323	0.2548	0.3531	0.1598
3	0.3147	0.2341	0.3308	0.1204
4	0.1401	0.3914	0.2466	0.2219
5	0.1712	0.4039	0.2604	0.1645
6	0.1581	0.2755	0.3246	0.2418
7	0.3004	0.3263	0.1642	0.2091
8	0.1109	0.2702	0.2095	0.4094
9	0.2611	0.1914	0.1988	0.3487
10	0.1921	0.3411	0.1691	0.2977

На основі даних з табл. 1 отримаємо відповідну гістограму з “факторіальними портретами” членів команди підтримки, зображену нижче на рис. 7.

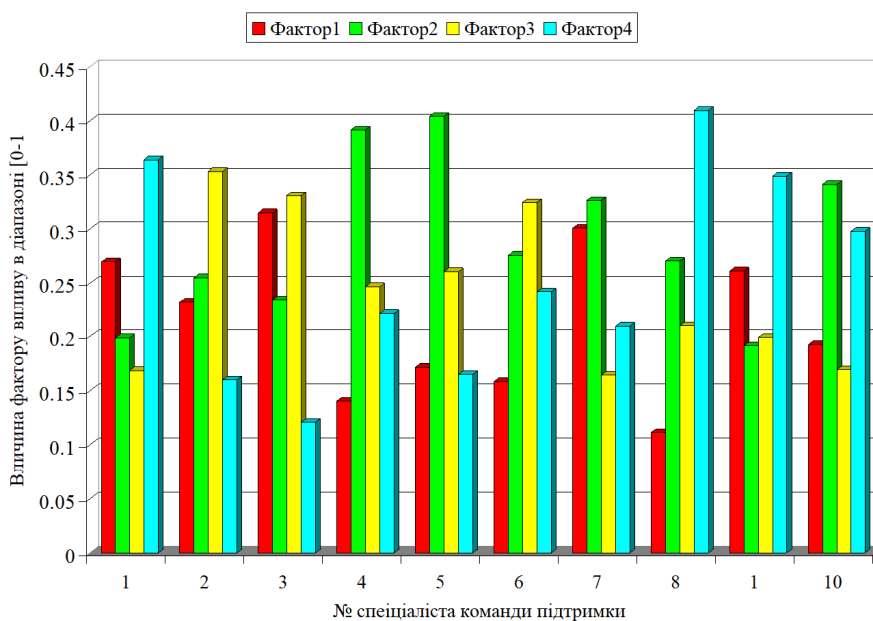


Рис. 7. Факторіальні портрети членів команди підтримки

Розрахувавши середнє значення сум кожного з факторів впливу для всіх членів команди підтримки, отримаємо дані, подані нижче на рис. 8 у вигляді гістограми, відповідно до яких домінуючим фактором впливу для вказаного дослідного кейсу є “Фактор 2”, сумарна величина якого є максимальною.

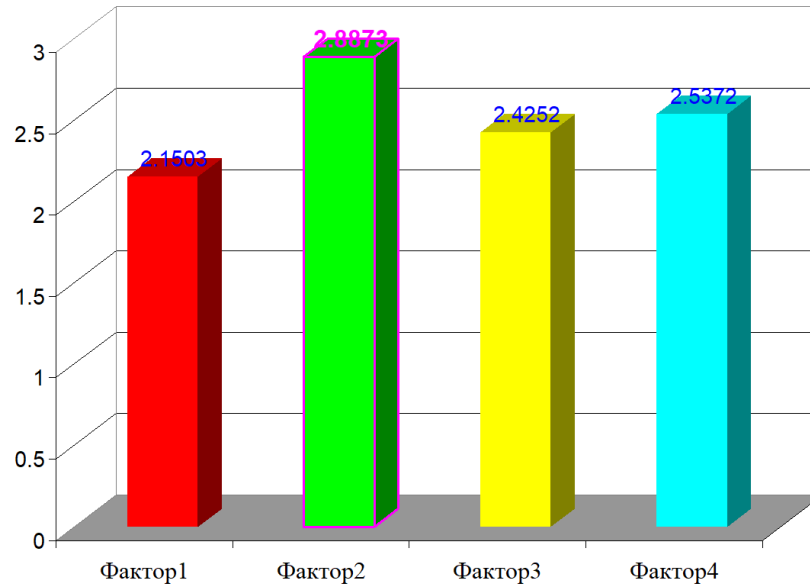


Рис. 8. Порівняльна гістограма середніх значень набору факторів впливу команди підтримки

Отже, отримано та представлено результат розв’язання прикладної практичної задачі виявлення домінуючого фактора впливу з допомогою моделювання аналізу факторів впливу автоматизації підтримки програмних комплексів. Додаткові експерименти показали також позитивний ефект використання вказаного типу моделювання для вирішення низки суміжних прикладних практичних задач, як-от: задача пошуку взаємозамінних членів команди підтримки; задача аналізу частоти та інтенсивності прояву досліджуваного фактору впливу; задача формування узагальненого портрету команди підтримки; задача пошуку членів команди з максимальним відхиленням від узагальненого портрету всієї команди; задача пошуку членів-антагоністів та ін.

Висновки

Розроблено модель аналізу факторів впливу автоматизації підтримки програмних комплексів на базі кольорових мереж Петрі. Модель призначена для дослідження глобального процесу аналізу факторів впливу автоматизації підтримки програмних комплексів саме на системному рівні без додаткових по-компонентних чи поелементних деталізацій. Основною науково-прикладною задачею, яку частково допомагає вирішувати розроблена модель, є задача аналізу та відновлення границь факторів впливу в моделях суб’єктивного сприйняття об’єктів підтримки з інкапсульованими штучними нейронними мережами типу багат шарового перцептрона, що своєю чергою входить до переліку задач науково-прикладної проблеми автоматизації підтримки програмних комплексів. Подано алгоритм аналізу факторів впливу автоматизації підтримки програмних комплексів, на основі якого функціонує також і розроблена модель, а також наведено графічне зображення блок-схеми алгоритму. Представлено детальний опис покрокового функціонування розробленої моделі за усіма можливими сценаріями. Побудовано дерево досяжності, що демонструє досяжність та скінченність кожного зі станів розробленої моделі на базі мережі Петрі та потактовий аналіз функціонування розробленої моделі з індексацією динаміки зміни станів мережі. Проведено дослі-

дження динаміки функціонування розробленої моделі на базі кольорових мереж Петрі та подано відповідні результати такого дослідження, візуалізовані у вигляді відповідних діаграм. Здійснено прикладне застосування аналізу факторів впливу автоматизації підтримки програмних комплексів для вирішення конкретної практичної задачі виявлення домінуючого фактору впливу з набору факторів впливу команди підтримки програмного комплексу. В перспективі розроблену модель буде використано для проведення подальших досліджень як в напрямку аналізу факторів впливу, так і в суміжних напрямках, зокрема взаємодії по типу “людина-людина” та “людина-машина”, шляхом додаткових вдосконалень та доопрацювань розробленої моделі аналізу факторів впливу автоматизації підтримки програмних комплексів.

REFERENCES

1. Cowell, C., Lotz, N., Timberlake, Ch. (2023). Automating DevOps with GitLab CI/CD Pipelines, *Packt Publishing*. ISBN: 9781803233000, 348 p.
2. Fewster, M., Graham, D. (1999). Software Test Automation Effective use of test execution tools, *Published by Addison-Wesley, Harlow, Essex, U.K.* ISBN: 0-201-33140-3, 574 p.
3. Humble, J., Farley, D. (2010). Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation, *Addison-Wesley Professional*. ISBN: 9780321670250, 512 p.
4. Jensen, K. (1996–1997). *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use*, Berlin: Spingler, 1–3.
5. Jensen, K., Kristensen, L. M. (2009). *Coloured Petri Nets: Modelling and Validation of Concurrent Systems*, Springer-Verlag Berlin Heidelberg, ISBN 978-3-642-00283-0, 384 p.
6. Kim, G., Behr, K., Spafford, G. (2020). *The Phoenix Project: A Novel about IT, DevOps, and Helping Your Business Win*, London: IT Revolution Press. 345 p.
7. Kim, G., Debois, P., Willis, J., Humble, J. (2016). *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*, IT Revolution Press, ISBN:978-1-942788-00-3, 480 p.
8. Peterson, J.-L. (1981). *Petri Net Theory and the Modeling of Systems*, New York, Prentice-Hall, ISBN: 978-0136619833, 290 p.
9. Rohit, K. (2024, February). Ai in test automation: overcoming challenges, embracing imperatives, *International Journal on Soft Computing, Artificial Intelligence and Applications (IJSCAI)*, 13(1), DOI: 10.5121/ijscai.2024.13101
10. Vang-Mata, R. (2020). *Multilayer Perceptrons: Theory and Applications*, New York, Nova Science Publishers, 143 p.
11. Al-oqaily, R., Alharbi, R., Alnomsi, S., Alharbi, A., Selmi, A. (2020). Incident Management with Knowledge base: College of computer in Qassim University as a case study. *International Journal of Engineering Research and Technology*, 13(3), 393–396. <https://dx.doi.org/10.37624/IJERT/13.3.2020.393-396>
12. Skrebeca, J. et al. (2021). Modern Development Trends of Chatbots Using Artificial Intelligence (AI). *62nd International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS)*, Riga, Latvia, 1–6. <https://doi.org/10.1109/itms52826.2021.9615258>
13. Basak, S., Agrawal, H., Jena, S., Gite, S., Bachute, M. et al. (2023). Challenges and limitations in speech recognition technology: A critical review of speech signal processing algorithms, tools and systems. *Computer Modeling in Engineering & Sciences*, 135(2), 1053–1089. <https://doi.org/10.32604/cmescs.2022.021755>
14. Ahsan, S. N., Ferzund, J., Wotawa, F. (2009). Automatic Software Bug Triage System (BTS) Based on Latent Semantic Indexing and Support Vector Machine. *Fourth International Conference on Software Engineering Advances*, Porto, Portugal, 216–221, <https://doi.org/10.1109/ICSEA.2009.92>
15. Sujatha, R., Bhattacharya, S., Jat, D. S. (2016). Comparative analysis of bug tracking tools. *The International Journal of Petroleum Technology*, 8(4), 4989–4998. https://www.researchgate.net/profile/Suja-Radha/publication/316888056_Comparative_analysis_of_bug_tracking_tools/links/5a1648720f7e9bc6481c8afa/Comparative-analysis-of-bug-tracking-tools.pdf
16. Sivaji, A. et al. (2020). Software Testing Automation: A Comparative Study on Productivity Rate of Open Source Automated Software Testing Tools For Smart Manufacturing. *IEEE Conference on Open Systems (ICOS)*, Kota Kinabalu, Malaysia, 7–12. <https://doi.org/10.1109/ICOS50156.2020.9293650>

17. Singh, M., Srivastava, V. M., Gaurav, K., Gupta, P. K. (2017). Automatic test data generation based on multi-objective ant lion optimization algorithm. *Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech)*, Bloemfontein, 168–174. <https://doi.org/10.1109/RoboMech.2017.8261142>
18. Grano, G., Ciurumelea, A., Panichella, S., Palomba, F., Gall, H. C. (2018). Exploring the integration of user feedback in automated testing of Android applications. *IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, Campobasso, 72–83. <https://doi.org/10.1109/SANER.2018.8330198>
19. Menegassi, A. A., Endo, A. T. (2016). An evaluation of automated tests for hybrid mobile applications. *XLII Latin American Computing Conference (CLEI)*, Valparaiso, 1–11. <https://doi.org/10.1109/CLEI.2016.7833337>
20. Shahabi, M. M. D., Badiei, S. P., Beheshtian, S. E., Akbari, R., Moosavi, S. M. R. (2017). On the performance of EvoPSO: A PSO based algorithm for test data generation in EvoSuite. *2nd Conference on Swarm Intelligence and Evolutionary Computation (CSIEC)*, Kerman, 2017, 129–134. <https://doi.org/10.1109/CSIEC.2017.7940170>
21. Raj, H. L. P., Chandrasekaran, K. (2018). NEAT Algorithm for Testsuite generation in Automated Software Testing. *IEEE Symposium Series on Computational Intelligence (SSCI)*, Bangalore, India, 2361–2368, <https://doi.org/10.1109/SSCI.2018.8628668>
22. Leotta, M., Clerissi, D., Ricca, F., Tonella, P. (2016). Approaches and Tools for Automated End-to-End Web Testing. *In Advances in Computers*, 101(1). Elsevier Inc. <https://doi.org/10.1016/bs.adcom.2015.11.007>
23. Ricca, F., Stocco, A. (2021). Web Test Automation: Insights from the Grey Literature. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12607 LNCS, 472–485. https://doi.org/10.1007/978-3-030-67731-2_35
24. Trudova, A., Dolezel, M., Buchalceva, A. (2020). Artificial intelligence in software test automation: A systematic literature review. *ENASE 2020. Proceedings of the 15th International Conference on Evaluation of Novel Approaches to Software Engineering*, 6(12), 181–192. <https://doi.org/10.5220/0009417801810192>
25. Serna, M. E., Acevedo, M. E., Serna, A. A. (2019). Integration of properties of virtual reality, artificial neural networks, and artificial intelligence in the automation of software tests: A review. *Journal of Software: Evolution and Process*, 31(7), 1–12. <https://doi.org/10.1002/smr.2159>
26. Li, J. J., Ulrich, A., Bai, X., Bertolino, A. (2020). Advances in test automation for software with special focus on artificial intelligence and machine learning. *Software Quality Journal*, 28(1), 245–248. <https://doi.org/10.1007/s11219-019-09472-3>
27. Sugali, K., Sprunger, C., Inukollu, V. (2021). Software Testing: Issues and Challenges of Artificial Intelligence & Machine Learning. *International Journal of Artificial Intelligence & Applications*, 12(1), 101–112. <https://doi.org/10.5121/ijaia.2021.12107>
28. Ricca, F., Marchetto, A., Stocco, A. (2021). AI-based test automation: A grey literature analysis. *Proceedings – 2021 IEEE 14th International Conference on Software Testing, Verification and Validation Workshops, ICSTW*, 263–270. <https://doi.org/10.1109/ICSTW52544.2021.00051>

ANALYSIS OF SOFTWARE COMPLEXES SUPPORT AUTOMATION IMPACT FACTORS WITH USAGE OF COLORED PETRI NETS

Andrii Pukach¹, Vasyl Teslyuk²

^{1,2}Lviv Polytechnic National University,

Department of Automated Control Systems, Lviv, Ukraine

E-mail: andriipukach@gmail.com, ORCID: 0009-0001-8563-3311

E-mail: vasy1.teslyuk@gmail.com, ORCID: 0000-0002-5974-9310

© Pukach A., Teslyuk V., 2024

Model based on colored Petri nets, and dedicated for analysis an impact factors of the software complexes support automation processes, has been developed. Model provides possibilities for simulation of the processes of impact factors analysis in the field of software complexes support automation when solving the scientific and applied task of analyzing and restoring the boundaries of

impact factors in supported objects subjective perception models with encapsulated artificial neural networks of multilayer perceptron type. The task of analyzing and restoring the boundaries of impact factors is included in the list of tasks of the scientific and applied problem of software complexes support automation. The object of the study is the process of analyzing an impact factors software complexes support automation. The subject of the study are methods and means of modeling the processes analysis an impact factors of software complexes support automation, based on the theory of Petri nets in general and colored Petri nets in particular. The purpose of the study is to develop a colored Petri nets based model for analysis an impact factors of the software complexes support automation. To achieve the set goal, the following research tasks were solved. A block diagram of the algorithm of the software complexes support automation impact factors analysis has been presented, as well as a description of the supported objects subjective perception model encapsulated by an artificial neural network of the multilayer perceptron type. A detailed description of the step-by-step functioning of the developed model within all possible scenarios is given as well. A reachability tree of the developed model is constructed, demonstrating the reachability and finiteness of each of the states of the presented colored Petri nets based models. A study of dynamics of the developed colored Petri nets based models functioning processes has been conducted as well as the depicted results of this study. As an example of software complex support automation impact factors analysis, – the applied practical problem of identifying the dominant impact factor among the set of impact factors of the software complex support team has been solved.

Keywords: automation, software complex, impact factor, support, model, colored Petri nets.