

doi: 10.32620/oikit.2026.107.17

УДК 004.94:004.65:629.3.027

А. І. Малюга,
Є. В. Соколова

Метод верифікації адаптивних програмних систем з використанням технології process mining

Національний аерокосмічний університет «Харківський авіаційний інститут»

Предметом статті є методична база верифікації адаптивних програмних систем, що змінюють поведінку під час виконання в залежності від поточного стану користувача. Метою є підвищення ефективності комп'ютерних засобів навчання операторів складної техніки, у тому числі, авіаційної, шляхом розроблення методу верифікації адаптивних програмних систем. Завдання, що потребують вирішення: провести критичний аналіз підходів до верифікації адаптивних програмних систем, розробити метод верифікації адаптивних програмних систем, що забезпечує автоматизоване оцінювання траєкторій адаптації через спеціалізовані метрики якості та провести експериментальне дослідження ефективності розробленого методу. У ході дослідження було розроблено метод адаптивної верифікації динамічних програмних систем на основі інтелектуального аналізу процесів для автоматизованої підтримки коректної адаптивної логіки критичних систем під час експлуатації, з можливістю безперервного самовдосконалення без втручання експертів. Результати експериментальної перевірки засвідчили ефективність розробленого методу.

Ключові слова: адаптивна програмна система, верифікація, адаптивна логіка, когнітивне перенавантаження користувача, навчальний тематичний модуль, темпоральна логіка, еталонна модель, логування, метрика придатності, методи безградієнтної оптимізації.

1. Вступ

Верифікація адаптивних програмних систем, що змінюють поведінку під час виконання (runtime adaptation) в залежності від стану користувача є важливою задачею, вирішення якої є необхідною умовою для побудови безпечних адаптивних систем навчання, авіаційних та медичних симуляторів, оскільки традиційні методи статичної формальної верифікації алгоритмів до впровадження системи та автономне тестування на еталонних наборах даних виявляються неспроможними перевірити коректність адаптивної логіки вибору конфігурації системи в реальних умовах експлуатації при різноманітності станів, у яких можуть перебувати користувачі.

Відсутність автоматизованих методів перевірки відповідності між фактичними траєкторіями адаптації системи та формально специфікованими правилами вибору навчальних модулів призводить до когнітивного перенавантаження або недовантаження користувача.

Таким чином, існує нагальна потреба у розробці методів автоматизованої верифікації адаптивних програмних систем на основі аналізу фактичних траєкторій адаптації, відомих із реальних журналів подій системи під час експлуатації з формуванням кількісних метрик якості адаптації. Такі метрики забезпечують оцінку ступеня відповідності фактичної поведінки системи формально специфікованим правилам та можливість подальшої корекції параметрів правил адаптації без ручного втручання експертів.

1. Формулювання наукової задачі та обґрунтування наукової новизни дослідження

Наукова задача дослідження полягає в розробці методу верифікації адаптивних програмних систем на основі інтеграції інтелектуального аналізу процесів (process mining), що забезпечує автоматизоване виявлення фактичних моделей виконання процесів з журналів подій системи під час експлуатації, а також перевірку ступеня відповідності траєкторій адаптації еталонній моделі процесу, через спеціалізовані метрики якості. Вирішення даної задачі забезпечує автоматизоване виявлення порушень адаптивної логіки із категоризацією за рівнем критичності для безпеки користувачів та самовдосконалення системи.

2. Аналіз публікацій та теоретичне підґрунтя дослідження

Конструктивні шляхи підвищення ефективності функціонування адаптивних програмних систем описані в роботах з інтелектуального аналізу процесів (process mining) [1] та перевірки відповідності бізнес-процесів організацій [2], у яких представлено математичний апарат для автоматичної побудови фактичних моделей процесів безпосередньо з журналів подій інформаційних систем. Проте, методи інтелектуального аналізу процесів традиційно застосовуються для аналізу організаційних процесів без адаптації до специфіки верифікації адаптивних програмних систем з динамічною зміною конфігурації залежно від стану користувача, що обмежує їхнє безпосереднє застосування для реалізації задачі верифікації коректності адаптивної логіки систем з адаптацією під час виконання.

Робота одного з авторів даного дослідження [3] присвячена методам адаптивної диспетчеризації навчальних модулів у системах віртуальної реальності на основі багатокритеріальної оцінки поточного стану користувача через інтеграцію когнітивних та біометричних параметрів для вибору оптимальної конфігурації системи. Такий підхід забезпечує високу точність відповідності вибраних модулів рівню підготовки користувача для підтримки ефективного процесу навчання.

Концепція журналу подій (event log) [4] передбачає наявність структурованого сховища інформації про всі зміни стану системи під час її експлуатації. Таке сховище фіксує повну історію взаємодій системи з користувачами у вигляді послідовності дискретних подій. Виявлення процесів (process discovery) [5] полягає в автоматичній побудові моделі процесу безпосередньо з журналу подій без апріорних припущень щодо структури процесу.

Для автоматичної побудови фактичної моделі процесу адаптації застосовується алгоритм індуктивного майнінгу (Inductive Miner algorithm), що дозволяє автоматично виявити всі унікальні траєкторії адаптації, що реально відбувалися під час експлуатації системи з різноманітними користувачами. Такий підхід забезпечує емпіричну основу для порівняння фактичної поведінки системи з формально специфікованою еталонною моделлю процесу адаптації, через процедури перевірки відповідності. Еталонна модель процесу адаптації (reference model) [6] представляє собою формальний опис коректної поведінки підсистеми диспетчеризації, що визначає правила вибору навчальних модулів, залежно від параметрів поточного стану користувача на основі експертних знань про ефективні стратегії навчання та вимоги безпеки у критичних ситуаціях.

Використання темпоральної логіки дозволяє визначити умови, за яких система диспетчеризації має підключати або відключати конкретні навчальні модулі для забезпечення коректності адаптивної логіки. Повна еталонна модель процесу адаптації формується як система темпоральних правил $R_{ref} = \{r_1, r_2, \dots, r_M\}$ для всіх навчальних модулів системи та різних типів рішень щодо адаптації. В сукупності ці правила задають формальну специфікацію коректної поведінки системи адаптації, тобто задають еталон для виявлення відхилень фактичних траєкторій адаптації з журналів подій через процедури перевірки відповідності фактичної моделі процесу еталонній специфікації.

Перевірка відповідності (conformance checking) є ключовим етапом розробленого методу верифікації, та полягає в кількісній оцінці ступеня відповідності побудованої фактичної моделі процесу адаптації з журналів подій системи формально специфікованої еталонній моделі процесу через обчислення спеціалізованих метрик якості, дозволяють кількісно оцінити ступінь відповідності фактичної поведінки системи еталонній специфікації, через обчислення відносних частот коректно відтворених траєкторій та нереалістичних переходів для формування об'єктивної оцінки якості адаптивної логіки системи на основі її фактичної поведінки під час експлуатації з реальними користувачами. Результатом виявлення порушень є структурована інформація про системні проблеми адаптивної логіки для подальшої автоматичної корекції параметрів правил адаптації через механізм самовдосконалення системи. Формально, порушення для події e_i визначається через логічний предикат $Violation(e_i) = \neg \exists r_j \in R_{ref}: Satisfies(e_i, r_j) = true$, що встановлює факт порушення у випадку, коли подія e_i не задовольняє жодному з правил r_j еталонної моделі процесу адаптації R_{ref} . Частота порушень обчислюється як відношення кількості подій у журналі, що порушують правила еталонної моделі, до загальної кількості подій адаптації для кількісної оцінки масштабу проблем з коректністю адаптивної логіки системи, що є об'єктивним показником якості функціонування системи адаптації на основі аналізу її фактичної поведінки під час експлуатації з реальними користувачами, та дозволяє реалізувати механізм самовдосконалення навчальної системи формально, реалізація задачі самовдосконалення здійснюється на основі визначення функції втрат $Q(\theta, B^*, w) = w_1 \cdot VR(\theta, B^*, w) + w_2 \cdot RT(\theta, B^*, w) + w_3 \cdot RU(\theta, B^*, w)$. Мінімізація функції втрат здійснюється через побудову симплексу в просторі параметрів та його ітеративну деформацію для пошуку локального мінімуму цільової функції без обчислення градієнтів, що дозволяє знаходити оптимальні значення параметрів адаптації $(\theta^*, B^{**}, w^*) = \arg \min_{\theta, B^*, w} M(\theta, B^*, w)$ через ітеративне відтворення поведінки системи з різними комбінаціями значень параметрів на журналі подій для оцінки якості функціонування системи з кожною комбінацією параметрів та вибору комбінації з найкращими показниками якості для оновлення параметрів правил адаптації.

3. Метод верифікації адаптивних програмних систем

Розроблений метод верифікації адаптивних програмних систем послідовно виконує збереження структурованих подій, формалізацію еталонної моделі процесу адаптації, автоматичну побудову фактичної моделі процесу з журналів подій, перевірку відповідності фактичної моделі еталонній специфікації, виявлення порушень адаптивної логіки із категоризацією за критичністю, а також

вдосконалення системи шляхом корекції параметрів правил адаптації. Метод складається з таких етапів.

Етап 1. Логування подій адаптації програмної системи. Мета першого етапу полягає в систематичному зборі та збереженні структурованих подій про всі зміни стану адаптивної системи під час експлуатації для подальшого автоматичного аналізу фактичних траєкторій адаптації.

Етап 2. Формалізація еталонної моделі процесу адаптації. Мета другого етапу технологічного процесу верифікації полягає в створенні формальної специфікації коректної поведінки системи адаптації через визначення правил вибору навчальних модулів залежно від параметрів поточного стану користувача.

Етап 3. Виявлення процесів адаптації через побудову фактичної моделі з журналів подій. Мета третього етапу полягає в автоматичному виявленні фактичної моделі процесу адаптації безпосередньо з журналів подій системи під час експлуатації через застосування алгоритму індуктивного майнінгу для побудови графа переходів між станами системи з визначенням частоти кожного переходу.

Етап 4. Перевірка відповідності через порівняння фактичної та еталонної моделей. Мета четвертого етапу полягає в кількісній оцінці ступеня відповідності фактичної моделі процесу адаптації з журналів подій формально специфікованій еталонній моделі процесу шляхом обчислення метрик придатності та точності.

Етап 5. Виявлення порушень адаптивної логіки із категоризацією за критичністю. Мета п'ятого етапу полягає в автоматичному детектуванні конкретних подій у журналі подій системи, які порушують формально специфіковані правила еталонної моделі процесу адаптації.

Етап 6. Вдосконалення через ітеративну корекцію параметрів адаптації. Мета шостого етапу верифікації полягає в автоматичній корекції параметрів правил вибору навчальних модулів на основі аналізу виявлених порушень адаптивної логіки через застосування методів безградієнтної оптимізації для мінімізації частоти відхилень від еталонної поведінки системи.

4. Експериментальна перевірка результатів дослідження

Метою експерименту було кількісне оцінювання впливу адаптивної верифікації на якість адаптивної логіки VR-системи навчання водінню за двома типовими сценаріями: міське керування з інтенсивним трафіком та керування в складних дорожніх умовах (дощ, погана видимість). Для кожного сценарію аналізувались журнали подій, отримані від користувачів під час проходження навчальних сесій до та після застосування механізму самовдосконалення адаптивної логіки.

Для оцінки коректності поведінки системи було використано метрики, узгоджене використання яких надало змогу побудувати об'єктивну кількісну оцінку якості адаптивної логіки системи. У табл. 1 наведено результати обчислення частоти порушень, придатності та точності для двох сценаріїв до та після адаптивної верифікації.

Для обох сценаріїв спостерігається помітне зниження частоти порушень еталонної моделі після застосування адаптивної верифікації. Одночасно підвищуються значення метрик придатності та точності. Це свідчить про те, що після самовдосконалення еталонна модель краще узгоджується з реальною поведінкою системи.

Таблиця 1

Результати застосування адаптивної верифікації в сценаріях
VR-навчання водінню

Сценарій	Стан системи	Кількість подій у журналі	Частота порушень, %	Придатність (Fitness)	Точність (Precision)
S1 – Міське водіння з інтенсивним трафіком	До самовдосконалення	12 500	7,8	0,86	0,81
S1 – Міське водіння з інтенсивним трафіком	Після самовдосконалення	13 200	3,1	0,93	0,88
S2 – Складні дорожні умови	До самовдосконалення	10 400	9,5	0,82	0,78
S2 – Складні дорожні умови	Після самовдосконалення	11 050	4,0	0,90	0,86

5. Обговорення результатів дослідження

Безумовними перевагами розробленого методу є здатність до ідентифікації виявленого порушення штатної поведінки навчальної системи, та визначення потенційних наслідків такого відхилення від еталонної моделі автоматично, без залучення до цього процесу експертів. Це суттєво впливає на швидкість та точність оцінювання і, по кінцевому рахунку, визначає рівень безпеки та ефективності навчання користувачів. До позитивних сторін методу слід віднести також спрямованість на використання у навчальних системах критичного призначення з підвищеними вимогами до надійності функціонування та забезпечення безпеки користувачів, зокрема у системах навчання операторів складного технологічного обладнання, тренажерів для підготовки пілотів, тощо.

Слабка сторона методу полягає у тому, що для оцінки коректності поведінки системи було використано лише три кількісні метрики - частоту порушень, метрику придатності та метрику точності. Розширення у подальшому номенклатури задіяних метрик надасть змогу суттєво підвищити ефективність розробленого методу.

Висновки

У роботі представлено вирішення актуальної науково-прикладної задачі адаптивної верифікації динамічних програмних систем на основі інтелектуального аналізу процесів для автоматизованої підтримки коректної адаптивної логіки критичних систем під час експлуатації, з можливістю безперервного самовдосконалення без втручання експертів.

Наукова новизна отриманих результатів полягає в розробленні методу адаптивної верифікації програмних систем зі змінною поведінкою, що поєднують

інтелектуальний аналіз процесів для побудови фактичної моделі поведінки системи з даних з журналів подій, перевірку відповідності фактичної моделі еталонній специфікації на основі використання темпоральної логіки та безградієнтну оптимізацію для корекції параметрів логіки адаптації програмної системи.

Метод дає можливість зменшити кількість порушень формально специфікованих правил коректної поведінки адаптивної програмної системи та забезпечити подальше вдосконалення адаптивної логіки навчальної системи з урахуванням рівня підготовки та стану користувача.

Список літератури

1. Calinescu R., Gerasimou S., Banks A. "Self-Adaptive Software Needs Quantitative Verification at Runtime." *Communications of the ACM*, 2012, 55(9): 69-73. DOI: 10.1145/2330667.2330686; Link: ACM Digital Library.

2. Su Z., Wang H., Yu H., Sun Q. "Process Mining over Sensor Data: Goal Recognition for Users of Powered Transhumeral Prosthetics." *Information and Software Technology*, 2025. DOI: 10.1016/j.infsof.2025.107680; Link: ScienceDirect.

3. Малуґа А. І. Методи побудови адаптивного програмного забезпечення навчальних систем // *Авіаційно-космічна техніка і технологія*. – 2025. - №6 (208). - С. 113 – 127.

4. Carmona J., van Dongen B., Solti A., Weidlich M. *Conformance Checking – Relating Processes and Models*. Springer, Cham, 2018. DOI: 10.1007/978-3-319-99414-7; Link: Springer.

5. Verbeek H. M. W., van der Aalst W. M. P. "Process Discovery and Conformance Checking Using Passages." In: *Business Process Management Workshops. BPM 2012. LNBP*, vol. 132. Springer, Berlin, Heidelberg, 2013. DOI: 10.1007/978-3-642-36285-9_24; Link: Author PDF.

6. Weerd J. De, De Backer M., Vanthienen J., Baesens B. "A Multi-Dimensional Quality Assessment of State-of-the-Art Process Discovery Algorithms Using Real-Life Event Logs." *Information Systems*, 2012, 37(7): 654–676. DOI: 10.1016/j.is.2012.02.004; Link: ScienceDirect.

References

1. Calinescu R., Gerasimou S., Banks A. "Self-Adaptive Software Needs Quantitative Verification at Runtime." *Communications of the ACM*, 2012, 55(9): 69-73. DOI: 10.1145/2330667.2330686; Link: ACM Digital Library.

2. Su Z., Wang H., Yu H., Sun Q. "Process Mining over Sensor Data: Goal Recognition for Users of Powered Transhumeral Prosthetics." *Information and Software Technology*, 2025. DOI: 10.1016/j.infsof.2025.107680; Link: ScienceDirect.

3. Maluḡa A. I. Metody pobudovy adaptivnoho prohramnoho zabezpechennia navchalnykh system / A. I. Maluḡa // *Aviatsiino-kosmichna tekhnika i tekhnolohiia*. – 2025. - №6 (208). - S. 113 – 127.

4. Carmona J., van Dongen B., Solti A., Weidlich M. *Conformance Checking – Relating Processes and Models*. Springer, Cham, 2018. DOI: 10.1007/978-3-319-99414-7; Link: Springer.

5. Verbeek H. M. W., van der Aalst W. M. P. "Process Discovery and Conformance Checking Using Passages." In: *Business Process Management Workshops. BPM 2012. LNBP*, vol. 132. Springer, Berlin, Heidelberg, 2013. DOI:

10.1007/978-3-642-36285-9_24; Link: Author PDF.

6. Weerdt J. De, De Backer M., Vanthienen J., Baesens B. "A Multi-Dimensional Quality Assessment of State-of-the-Art Process Discovery Algorithms Using Real-Life Event Logs." *Information Systems*, 2012, 37(7): 654–676. DOI: 10.1016/j.is.2012.02.004; Link: ScienceDirect.

Надійшла до редакції 14.02.2026, розглянута на редколегії 16.02.2026

Method for verifying adaptive software systems using process mining technology

The subject of the article is the methodological basis for the verification of adaptive software systems that change their behavior during execution depending on the current state of the user. The goal is to increase the efficiency of computer-aided training for operators of complex equipment, including aviation equipment, by developing a method for verifying adaptive software systems. Tasks that need to be solved: to conduct a critical analysis of approaches to the verification of adaptive software systems, to develop a method for verifying adaptive software systems that provides automated assessment of adaptation trajectories through specialized quality metrics, and to conduct an experimental study of the effectiveness of the developed method. During the study, a method for adaptive verification of dynamic software systems was developed based on intelligent process analysis for automated support of correct adaptive logic of critical systems during operation, with the possibility of continuous self-improvement without expert intervention. The results of the experimental verification confirmed the effectiveness of the developed method.

Keywords: adaptive software system, verification, adaptive logic, user cognitive overload, educational thematic module, temporal logic, reference model, logging, fitness metric, gradient-free optimization methods.

Відомості про авторів:

Малюга Артур Іванович – аспірант кафедри інженерії програмного забезпечення Національного аерокосмічного університету «Харківський авіаційний інститут», м. Харків, Україна, e-mail: artur.maliuha@gmail.com, ORCID: 0009-0004-1855-5757

Соколова Євгенія Віталіївна – доцент, кандидат технічних наук, доцент кафедри інженерії програмного забезпечення Національного аерокосмічного університету «Харківський авіаційний інститут», м. Харків, Україна, e-mail: y.sokolova@khai.edu, ORCID: 0000-0002-1497-4987

About the authors:

Artur MALIUHA – PhD Student, Department of Software Engineering, National Aerospace University "Kharkiv Aviation Institute", Kharkiv, Ukraine, e-mail: artur.maliuha@gmail.com, ORCID: 0009-0004-1855-5757

Yevheniia SOKOLOVA – Associate Professor, Candidate of Technical Sciences, Associate Professor of the Department of Software Engineering, National Aerospace University "Kharkiv Aviation Institute", Kharkiv, Ukraine, e-mail: y.sokolova@khai.edu, ORCID: 0000-0002-1497-4987