

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Міністерство освіти і науки України

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Міністерство освіти і науки України

Кваліфікаційна наукова
праця на правах рукопису

ЛЕМЕШКО В'ЯЧЕСЛАВ АНАТОЛІЙОВИЧ

УДК 004.451.44:004.052

ДИСЕРТАЦІЯ
МЕТОДИ ТА ПЛАТФОРМА АВТОМАТИЗАЦІЇ ПРОЄКТУВАННЯ ТА
ВИКОНАННЯ БІЗНЕС-СЕРВІСІВ

126 – Інформаційні системи та технології

12 – Інформаційні технології

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ ЛЕМЕШКО В.А.

Науковий керівник КОРНАГА Ярослав Ігорович, доктор технічних наук,
професор

Київ - 2026

АНОТАЦІЯ

Лемешко В.А. Методи та платформа автоматизації проєктування та виконання бізнес-сервісів. – Кваліфікаційна наукова праця на правах рукопису.

Дисертаційна робота на здобуття ступеня доктора філософії за спеціальністю 126 “Інформаційні системи та технології”. - Національний технічний університет України “Київський політехнічний інститут імені Ігоря Сікорського”, м. Київ, 2026.

Актуальність теми дослідження зумовлена необхідністю підвищення ефективності виконання бізнес-сервісів в умовах зростання складності бізнес-процесів, динамічного навантаження та ресурсних обмежень. Використання формалізованих і автоматизованих підходів сприяє стабільності, передбачуваності, своєчасному виявленню відхилень і дотриманню цільових показників рівня сервісу в сучасних інформаційних системах.

Наявні підходи нерідко демонструють обмеження під час роботи в умовах нерівномірного навантаження, оскільки вони не забезпечують повного циклу контролю використання ресурсів, своєчасного виявлення відхилень та підтримання показників щодо угод про рівень надання послуг. Недоліки сучасних платформ управління бізнес-процесами проявляються під час інтеграції різних моделей, конфігурацій і сценаріїв виконання, коли стандартні засоби моніторингу та планування не забезпечують належного рівня керованості системою. Використання лише базових інструментів є недостатнім, оскільки вони не контролюють повною мірою поведінку всієї платформи в умовах динамічного навантаження.

Питання розроблення, аналізу та оптимізації складних програмних і технічних систем розглядалися у працях багатьох вітчизняних і зарубіжних науковців. До кола дослідників, чий роботи охоплюють окремі аспекти автоматизації виконання процесів, управління ресурсами, оптимізації програмних систем та організації складних інформаційних середовищ, належать О.А. Машков, Ю.П. Зайченко, В.Г. Лазарєв, О.В. Коваль, В.С. Семеніхін, В.Ф. Колчин, В.К. Попков, О.В. Барабаш, А.П. Мусієнко, Г.А. Кучук, Д.М. Обідін, О.А. Кононов, В.Є. Мухін, Я.І. Корнага, В. Цветков, S. Newman, K. Hoffman та інші.

В наведених наукових працях, присвячених проблематиці проєктування та оптимізації програмних систем, розглядалися питання організації програмних рішень, вибору критеріїв їх ефективності та підвищення якості функціонування. У цих роботах обговорювалися підходи до аналізу структури програмних систем, удосконалення обчислювальних процедур, забезпечення надійності виконання операцій і раціонального використання доступних ресурсів. Також увага приділялася питанням синхронізації обробки задач, підтриманню стабільності функціонування та оцінюванню продуктивності програмних систем за різних умов роботи.

Окремий напрям стосувався підвищення стійкості програмних засобів до помилок та впливу змін у параметрах середовища, що впливають на якість виконання операцій. У ряді робіт аналізувалися методи зменшення затримок у обробці даних та покращення показників продуктивності шляхом оптимізації алгоритмів і структури програмних компонентів. При цьому зроблено акцент на забезпеченні передбачуваності роботи та підтриманні стабільних характеристик виконання.

У сучасній теорії та практиці автоматизації проєктування й виконання бізнес-процесів спостерігається протиріччя між потребою у стабільному та передбачуваному функціонуванні бізнес-сервісів і тими можливостями, які надають наявні методи та платформи для забезпечення керованого виконання процесів у змінних умовах навантаження. Одним із ключових викликів є те, що інструменти швидкого моделювання та конфігурування процесів не завжди дозволяють забезпечувати необхідні показники продуктивності, ресурсної ефективності та дотримання встановлених параметрів виконання.

Для усунення зазначеного протиріччя у дисертаційній роботі розглядається наукове завдання *розроблення моделі та методів керованого виконання бізнес-процесів, що поєднують засоби моделювання, моніторингу використання ресурсів та адаптивного планування задач. Запропонований підхід спрямований на підвищення стабільності роботи бізнес-сервісів і забезпечення узгодженого виконання процесів відповідно до встановлених вимог.*

Метою дисертаційної роботи є підвищення ефективності виконання кроків бізнес-процесів у платформах автоматизації проєктування та виконання бізнес-сервісів в умовах змінного навантаження.

Основні завдання дослідження включають:

1. Провести порівняльний аналіз методів і засобів організації сучасних платформ автоматизації проєктування та виконання бізнес-сервісів для визначення особливостей та проблем їх реалізації.
2. Дослідити механізми моніторингу виконання кроків бізнес-процесів у платформі автоматизації проєктування та виконання бізнес-сервісів, визначити вимоги до телеметрії, інструментованого контролю ресурсів та виявлення порушень стабільності.
3. Розробити метод реактивного мульти-токенового планування задач за дедлайном старту, який дозволяє знизити ризик перевантаження середовища та зменшити частку порушень дедлайнів старту в умовах змінного та пікового навантаження.
4. Розробити метод планування задач за правилом "перший прийшов - перший оброблений", який дозволяє зменшити ризик аварійних збоїв середовища виконання, зокрема, спричинених перевищенням обмеження оперативної пам'яті.
5. Розробити модель моніторингу та виконання кроків бізнес-процесів у ресурсно-обмеженому середовищі.
6. Розробити програмне забезпечення системи управління та перевірки виконання кроків бізнес-процесів у платформі автоматизації проєктування та виконання бізнес-сервісів.

Об'єктом дослідження є процес виконання кроків бізнес-процесів у платформах автоматизації проєктування та виконання бізнес-сервісів в умовах змінного навантаження.

Предметом дослідження є моделі, методи та програмні засоби підвищення ефективності виконання кроків бізнес-процесів, що забезпечують моніторинг

використання ресурсів, стабільність роботи бізнес-сервісів у платформах автоматизації проєктування та виконання бізнес-сервісів.

Методи дослідження. Дисертаційне дослідження виконано з використанням методів математичного та імітаційного моделювання, теорії масового обслуговування і теорії автоматичного керування, а також методів порівняльного та експериментального аналізу виконання бізнес-сервісів в умовах змінного навантаження. Це дало змогу дослідити особливості виконання кроків бізнес-процесів, оцінити затримки, пропускну здатність, ресурсоспоживання та стабільність функціонування системи.

Наукова новизна одержаних результатів.

1. Вперше розроблено метод реактивного мульти-токенового планування задач за дедлайном старту, який відрізняється від існуючих поєднанням етапів перевірки ресурсів для запуску задачі за мультиресурсною токенною моделлю з резервуванням "видано/повернуто" та вибором допустимої задачі з найменшим дедлайном старту, зазначений метод дозволяє знизити ризик перевантаження середовища та зменшити частку порушень дедлайнів старту в умовах змінного та пікового навантаження.

Удосконалено метод планування задач за правилом "перший прийшов - перший оброблений", який відрізняється введенням механізму ресурсного допуску задач перед запуском і правилом відкладеного старту за нестачі ресурсів та вибором допустимої задачі зі збереженням черговості обробки із очікуванням вивільнення ресурсів, що дозволяє зменшити ризик аварійних збоїв середовища виконання, зокрема спричинених перевищенням обмежень оперативної пам'яті.

Удосконалено модель моніторингу та виконання кроків бізнес-процесів у ресурсно-обмеженому середовищі, яка відрізняється від існуючих тим, що подає крок процесу як задачу з багатовимірними ресурсними потребами та часом постановки в чергу і дедлайном старту, який відокремлюється правилом пріоритету від механізму багаторесурсної перевірки здійсненності запуску, що задає метод вибору задачі із існуючої черги як фіксований параметр моделі та дозволяє

порівняння альтернативних пріоритетів за показником частки порушень дедлайнів старту.

Набула подальшого розвитку інформаційна технологія автоматизації проектування та виконання бізнес-процесів у частині багаторівневої інтеграції засобів моделювання, виконання, моніторингу, аналітики та керування, що відрізняється реалізацією у замкненому контуру “виконання - телеметрія - аналітика - керування - вплив на виконання”, розробленого методу реактивного мульти-токенового планування задач за дедлайном старту та удосконалено методу планування задач за правилом "перший прийшов - перший оброблений", що дозволяє підвищити обґрунтованість підтримання стабільності виконання процесів за змінного навантаження.

Практичне значення отриманих результатів полягає у створенні інформаційної системи для моніторингу й керованого виконання бізнес-процесів за змінного навантаження та обмежених ресурсів. Результати тестування підтверджують її придатність для оцінювання ресурсів, цільового показника рівня сервісу, порівняння методів планування й удосконалення платформ автоматизації проектування та виконання бізнес-сервісів.

У результаті порівняння базової та запропонованої конфігурації на промисловому середовищі показало зменшення частки порушених дедлайнів старту у сценаріях із 5000 задач. У першому сценарії цей показник знизився з 14,56% до 5,48%, тобто на 9,08%, у другому - з 67,88% до 25,81%, тобто на 42,07%. Найбільше зменшення частки порушень спостерігається за обмеженого обсягу оперативної пам'яті. Зі збільшенням доступної пам'яті рівень порушень дедлайнів зменшується в обох конфігураціях, однак запропонована конфігурація забезпечує нижчі значення частки порушених дедлайнів, що підтверджує доцільність використання запропонованої конфігурації.

Таким чином, мета дисертаційного дослідження щодо підвищення ефективності виконання кроків бізнес-процесів у платформах автоматизації проектування та виконання бізнес-сервісів в умовах змінного навантаження досягнута, а всі поставлені наукові завдання виконані в повному обсязі. Отримані

наукові результати становлять внесок у розвиток інформаційних технологій у частині виконання бізнес-процесів, розроблення методів планування.

Перспективними напрямками подальших досліджень є удосконалення методів контролю ресурсів, розроблення засобів адаптивного планування кроків процесів та розширення моделей оцінювання ефективності виконання бізнес-процесів у змінних технічних та експлуатаційних умовах.

Ключові слова: модель, система управління, функціональна стійкість, інформаційна система, розподіл ресурсів, планування завдань, комп'ютерні системи, моніторинг, інформаційна технологія, динамічне середовище, алгоритм, моделювання, оптимальне керування, метод усереднення, відмовостійкість.

ANOTATION

Lemeshko V.A. Methods and a Platform for Automating the Design and Execution of Business Services. - Qualification scientific work in the form of a manuscript.

Dissertation for the degree of Doctor of Philosophy in specialty 126 "Information Systems and Technologies". - National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, 2026.

The relevance of the research topic is determined by the need to improve the efficiency of business service execution under conditions of increasing business process complexity, dynamic workloads, and resource constraints. The use of formalized and automated approaches contributes to stability, predictability, timely detection of deviations, and compliance with target service level indicators in modern information systems.

Existing approaches often demonstrate limitations under uneven workloads, since they do not provide a full cycle of resource usage control, timely detection of deviations, and maintenance of service level agreement indicators. The shortcomings of modern business process management platforms become evident when integrating different models, configurations, and execution scenarios, where standard monitoring and scheduling tools do not provide a sufficient level of system controllability. The use of only basic tools is insufficient, as they do not fully control the behavior of the entire platform under dynamic load conditions.

Issues related to the development, analysis, and optimization of complex software and technical systems have been considered in the works of many domestic and foreign researchers. The range of scholars whose works cover certain aspects of process execution automation, resource management, software system optimization, and the organization of complex information environments includes O.A. Mashkov, Yu.P. Zaichenko, V.H. Lazarev, O.V. Koval, V.S. Semenikhin, V.F. Kolchin, V.K. Popkov, O.V. Barabash, A.P. Musiienko, H.A. Kuchuk, D.M. Obidin, O.A. Kononov, V.Ye. Mukhin, Ya.I. Kornaha, V. Tsvetkov, S. Newman, K. Hoffman, and others.

In the cited scientific works devoted to the design and optimization of software systems, researchers considered issues of organizing software solutions, selecting criteria for their efficiency, and improving the quality of functioning. These works discussed approaches to analyzing the structure of software systems, improving computational procedures, ensuring the reliability of operation execution, and making rational use of available resources. Attention was also paid to the synchronization of task processing, maintenance of stable functioning, and evaluation of software system performance under different operating conditions.

A separate line of research concerned improving the fault tolerance of software tools and mitigating the impact of changes in environmental parameters that affect the quality of operation execution. A number of works analyzed methods for reducing data processing delays and improving performance indicators by optimizing algorithms and the structure of software components. At the same time, emphasis was placed on ensuring predictability of operation and maintaining stable execution characteristics.

In modern theory and practice of automating the design and execution of business processes, there is a contradiction between the need for stable and predictable functioning of business services and the capabilities provided by existing methods and platforms for ensuring controlled process execution under varying workload conditions. One of the key challenges is that rapid process modeling and configuration tools do not always make it possible to ensure the required levels of performance, resource efficiency, and compliance with specified execution parameters.

To eliminate this contradiction, the dissertation addresses the scientific task of developing a model and methods for controlled execution of business processes that combine modeling tools, resource usage monitoring, and adaptive task scheduling. The proposed approach is aimed at improving the stability of business services and ensuring coordinated process execution in accordance with the established requirements.

The purpose of the dissertation is to improve the efficiency of executing business process steps in platforms for the automation of business service design and execution under variable workload conditions.

The main research objectives are as follows:

1. To perform a comparative analysis of methods and tools for organizing modern platforms for the automation of business service design and execution in order to determine the features and problems of their implementation.
2. To investigate the monitoring mechanisms for execution of business process steps in a platform for the automation of business service design and execution, and to determine the requirements for telemetry, instrumented resource control, and detection of stability violations.
3. To develop a method of reactive multi-token task scheduling by start deadline, which makes it possible to reduce the risk of environment overload and decrease the share of start-deadline violations under variable and peak workloads.
4. To develop a task scheduling method based on the “first come, first served” rule, which makes it possible to reduce the risk of critical failures of the execution environment, in particular those caused by exceeding the RAM limit.
5. To develop a model for monitoring and execution of business process steps in a resource-constrained environment.
6. To develop software for a system of control and verification of business process step execution in a platform for the automation of business service design and execution.

The object of research is the process of executing business process steps in platforms for the automation of business service design and execution under variable workload conditions.

The subject of research is models, methods, and software tools for improving the efficiency of business process step execution, ensuring resource usage monitoring and stable operation of business services in platforms for the automation of business service design and execution.

Research methods. The dissertation research was carried out using methods of mathematical and simulation modeling, queueing theory, and automatic control theory, as well as methods of comparative and experimental analysis of business service execution under variable workload conditions. This made it possible to investigate the features of business process step execution and to evaluate delays, throughput, resource consumption, and system stability.

Scientific novelty of the obtained results. For the first time, a method of reactive multi-token task scheduling by start deadline has been developed. It differs from existing methods by combining the stages of resource availability verification for task launch based on a multi-resource token model with “issued/returned” reservation and the selection of an admissible task with the earliest start deadline. This method makes it possible to reduce the risk of environment overload and decrease the share of start-deadline violations under variable and peak workloads.

An improved task scheduling method based on the “first come, first served” rule has been developed. It differs by introducing a mechanism for checking task resource admissibility before execution, as well as a deferred-start rule in case of insufficient resources and a procedure for selecting an admissible task while preserving the processing order and waiting for resources to be released. This helps reduce the risk of critical failures in the execution environment, particularly those caused by exceeding RAM limits.

An improved model for monitoring and executing business process steps in a resource-constrained environment has been developed. Unlike existing models, it represents a process step as a task with multidimensional resource requirements, a queue entry time, and a start deadline. The priority rule is separated from the mechanism for multi-resource feasibility checking of task execution, which defines the task selection method from the existing queue as a fixed model parameter and makes it possible to compare alternative priority rules using the proportion of start-deadline violations as a performance metric.

The information technology for automating the design and execution of business processes has been further developed in terms of multi-level integration of modeling, execution, monitoring, analytics, and control tools. It differs by being implemented within a closed-loop framework of “execution – telemetry – analytics – control – impact on execution” by the developed method of reactive multi-token task scheduling based on start deadlines, and by the improved task scheduling method based on the “first come, first served” rule. This makes it possible to enhance the validity of maintaining process execution stability under variable workloads.

It involves the development of an information system for monitoring and controlled execution of business processes under variable workloads and limited resources. The testing results confirm its suitability for resource assessment, evaluation of target service level indicators, comparison of scheduling methods, and improvement of automation platforms for the design and execution of business services.

A comparison of the baseline and proposed configurations in an industrial environment showed a decrease in the share of start-deadline violations in scenarios with 5,000 tasks. In the first scenario, this indicator fell from 14.56% to 5.48%, i.e., by 9.08%; in the second, from 67.88% to 25.81%, i.e., by 42.07%. The greatest reduction was observed under limited RAM conditions. As available memory increased, deadline violations decreased in both configurations; however, the proposed configuration consistently yielded lower violation rates, confirming its feasibility.

The goal of the dissertation research aimed at improving the efficiency of business process step execution in platforms for the automation of business service design and execution under variable workload conditions has been achieved, and all the stated scientific objectives have been fully accomplished. The obtained scientific results contribute to the development of information technologies in the area of business process execution and the development of scheduling methods.

Promising directions for further research include improvement of resource control methods, development of means for adaptive scheduling of process steps, and expansion of models for evaluating the efficiency of business process execution under changing technical and operational conditions.

Keywords: model, control system, functional stability, information system, resource allocation, task scheduling, computer systems, monitoring, information technology, dynamic environment, algorithm, modeling, optimal control, averaging method, fault tolerance.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА

Наукові праці, в яких опубліковані основні наукові результати дисертації:

1. Корнага Я.І., Лемешко В.А. Порівняння алгоритмів Modified First-Come-First-Served та Reactive Multi-Resource Token–Earliest Deadline First за умов обмеженого доступу до оперативної пам'яті // Телекомунікаційні та інформаційні технології. № 3 (2025). С. 26–33. ISSN 2412-4338. <https://doi.org/10.31673/2412-4338.2025.038704>

Корнага Я.І. брав участь у постановці наукового завдання порівняльного дослідження методів в умовах обмеженого доступу до оперативної пам'яті, а також в узагальненні одержаних результатів. Лемешко В.А. розробив алгоритмічний опис удосконаленого методу, реалізував експериментальне середовище та програмну реалізацію зазначених методів, виконав аналіз показників продуктивності, порушень цільового показника сервісу та використання ресурсів, що дозволило вперше обґрунтувати удосконалений метод реактивного мульти-токенового планування задач за правилом "перший прийшов - перший оброблений" та обґрунтувати метод реактивного мульти-токенового планування задач за дедлайном старту в ресурсно-обмеженому середовищі.

2. Лемешко В.А. Формалізація процесу планування за методом RMRT-EDF з урахуванням обмежених ресурсів // Адаптивні системи автоматичного управління. Т. 2, № 47 (2025). С. 101–107. ISSN 1560-8956. <https://doi.org/10.20535/1560-8956.47.2025.340183>

Лемешко В.А. виконав аналіз наукових публікацій за темою статті, запропонував математичну модель процесу планування, що дозволило удосконалити модель моніторингу та виконання кроків бізнес-процесів у ресурсно-обмеженому середовищі.

3. Пархомей І., Бойко Ю., Лемешко В. Математична модель та адаптивне управління алгоритмом обслуговування даних журналу в умовах змінного серверного навантаження // Вісник Хмельницького національного університету.

Серія «Технічні науки». Т. 347, № 1 (2025). С.168–174. ISSN 2307-5732.
<https://doi.org/10.31891/2307-5732-2025-347-23>

Пархомей І. запропонував критерій ефективності, взяв участь в узагальненні результатів роботи. Бойко Ю. здійснив постановку наукового завдання дослідження, обрахував результати, подані на графіках "Показник sql IOPS", "Показник sql processes blocked", "Показник sql disk queue". Лемешко В. розробив опис параметрів налаштування й алгоритм обслуговування даних журналу бізнес-процесів, що забезпечило розвиток інформаційної технології автоматизації проектування та виконання бізнес-процесів через реалізацію механізму адаптивного обслуговування й оцінювання ефективності за показниками продуктивності серверного середовища.

4. Пархомей І., Бойко Ю., Лемешко В., Єрмоєнко О. Оптимізація логування в інформаційних системах на основі математичної моделі управління буфером у шаблоні Producer–Consumer // Вимірювальна та обчислювальна техніка в технологічних процесах. № 2(2025). С. 218–225. ISSN 2219-9365.
<https://doi.org/10.31891/2219-9365-2025-82-30>

Пархомей І. визначив науково-практичне завдання дослідження. Бойко Ю. формалізував умову ефективного функціонування буфера та виконав розрахунки параметрів моделі. Єрмоєнко О. виконав обрахунок і аналіз експериментальних результатів. Лемешко В. розробив загальні схеми базового і запропонованого підходів для підсистеми логування в системі виконання бізнес-процесів, здійснив параметричне удосконалення підсистеми логування, обґрунтував застосування буфера з агрегацією повідомлень і використання даних Query Store для оцінювання ефективності запропонованого підходу, що забезпечило подальший розвиток інформаційної технології автоматизації проектування та виконання бізнес-процесів на основі даних моніторингу та оцінювання поточного стану середовища.

5. Пархомей І., Бойко Ю., Лемешко В. Алгоритм налаштування кількості потоків для виконання фонових задач // Вимірювальна та обчислювальна техніка в технологічних процесах. № 4 (2024). С. 162–173. ISSN 2219-9365.
<https://doi.org/10.31891/2219-9365-2024-80-20>

Пархомей І. поставив наукову задачу та брав участь в узагальненні отриманих результатів. Бойко Ю. брав участь у визначенні ресурсних параметрів, що враховуються в алгоритмі, та в обґрунтуванні відповідних розрахунків. Лемешко В. розробив загальну схему підсистеми обробки задач, метод визначення кількості фонових обробників та запропонував конфігурацію кількості одночасно виконуваних фонових операцій для всіх підсистем. Загальна схема підсистеми обробки задач відображає виконання кроків бізнес-процесів у спільному ресурсно-обмеженому середовищі та враховує взаємодію підсистем, які використовують спільні обчислювальні ресурси, що дозволило удосконалити модель моніторингу та виконання кроків бізнес-процесів в умовах змінного навантаження. Алгоритм визначення кількості фонових обробників ґрунтується на аналізі стану системи, графіків навантаження, журналів подій і зворотного зв'язку щодо реакції середовища на зміну конфігурації, що забезпечило подальший розвиток інформаційна технологія автоматизації проєктування та виконання бізнес-процесів на основі даних моніторингу та оцінювання поточного стану середовища.

Наукові праці, які засвідчують апробацію матеріалів дисертації:

1. Корнага Я. І., Лемешко В.А. RMRT-EDF: реактивний мультиресурсний токен-планувальник для гарантованого запуску задач у реальному часі // XIII Міжнародна науково-практична конференція з інформаційних систем та технологій. 21-22 травня 2025 року. Матеріали конференції, ст. 35-38; Форма участі – дистанційно.

Корнага Я.І. запропонував методологію дослідження реактивного мультиресурсного токенового планування задач за дедлайном старту та брав участь в узагальненні отриманих результатів. Лемешко В.А. розробив алгоритмічне та архітектурне представлення методу, виконав його імітаційну перевірку та аналіз показників ефективності, що дозволило вперше обґрунтувати метод реактивного мульти-токенового планування задач за дедлайном старту в ресурсно-обмеженому середовищі.

Праці, які додатково відображають результати дисертації:

Відсутні.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	20
ВСТУП.....	21
РОЗДІЛ 1 ПОРІВНЯЛЬНИЙ АНАЛІЗ СУЧАСНИХ ПЛАТФОРМ	
АВТОМАТИЗАЦІЇ ПРОЄКТУВАННЯ ТА ВИКОНАННЯ	34
БІЗНЕС-СЕРВІСІВ	34
1.1. Аналіз організації, проєктування та виконання бізнес-процесів у сучасних платформах з низьким обсягом програмування	34
1.2. Дослідження особливостей побудови сучасних систем управління ресурсами та угоди рівня надання послуг у платформах проєктування та виконання бізнес-сервісів.....	42
1.3. Аналіз архітектурних рішень платформ автоматизації проєктування та виконання бізнес-сервісів.....	51
1.4. Дослідження виконання бізнес-процесів у платформах автоматизації проєктування та виконання бізнес-сервісів.....	60
1.5. Постановка завдання дослідження та часткових завдань дисертації	68
Висновки до розділу 1	72
РОЗДІЛ 2 ДОСЛІДЖЕННЯ МЕХАНІЗМІВ МОНІТОРИНГУ ВИКОНАННЯМ	
БІЗНЕС-ПРОЦЕСІВ У ПЛАТФОРМАХ АВТОМАТИЗАЦІЇ ПРОЄКТУВАННЯ	
ТА ВИКОНАННЯ БІЗНЕС-СЕРВІСІВ	74
2.1. Аналіз засобів підтримки безпечного та стабільного виконання бізнес-процесів у платформах автоматизації проєктування та виконання бізнес-сервісів	74
2.2. Дослідження системи моніторингу виконання бізнес-процесів в платформах автоматизації проєктування та виконання бізнес-сервісів	78
2.3. Розробка моделі моніторингу виконання кроків бізнес-процесів в платформах автоматизації проєктування та виконання бізнес-сервісів на основі модуля контролю виконання бізнес-процесів.....	84
2.3.1. Організація засобів моніторингу та зворотного зв'язку у платформі автоматизації проєктування та виконання бізнес-сервісів	91
2.3.2. Організація модуля контролю запропонованої системи моніторингу виконання бізнес-процесів	95
Висновки до розділу 2	100

РОЗДІЛ 3 РОЗРОБКА МЕТОДУ РОЗПОДІЛУ ВИКОНАННЯ КРОКІВ БІЗНЕС-ПРОЦЕСІВ У ПЛАТФОРМІ АВТОМАТИЗАЦІЇ ПРОЄКТУВАННЯ ТА ВИКОНАННЯ БІЗНЕС-СЕРВІСІВ.....	102
3.1. Вибір і обґрунтування моніторингової моделі у платформі автоматизації проєктування та виконання бізнес-сервісів.....	102
3.2. Розробка алгоритму ресурсно-орієнтованого планування задач із використанням гібридного підходу.....	107
3.3. Моделювання сценаріїв виконання кроків бізнес-процесів у середовищі з обмеженими ресурсами	114
3.4. Розробка модифікованого методу “перший прийшов – перший оброблений” з урахуванням багатовимірних ресурсів.....	119
3.5. Порівняльні аналітичні оцінки методу планування виконання кроків бізнес-процесів в платформі автоматизації проєктування та виконання бізнес-сервісів	124
Висновки до розділу 3	128
РОЗДІЛ 4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ УПРАВЛІННЯ ТА ПЕРЕВІРКИ СЕРЕДОВИЩА ПРОЄКТУВАННЯ ТА ВИКОНАННЯ БІЗНЕС-СЕРВІСІВ.....	130
4.1. Розробка спеціалізованого фреймворку для дослідження параметрів та способів контролю за виконанням бізнес-процесів в платформах проєктування та виконання бізнес-сервісів	130
4.2. Графічний інтерфейс розробленого спеціалізованого середовища системи управління та перевірки	138
4.3. Система управління та перевірки середовища проєктування та виконання бізнес-сервісів.....	145
4.3.1. Розробка системи управління в середовищі проєктування та виконання бізнес-сервісів.....	145
4.3.2. Підсистема управління та перевірки ресурсів і УРНП у платформі автоматизації проєктування та виконання бізнес-сервісів	149
4.3.3. Підсистема обробки та підготовки даних у середовищі проєктування та виконання бізнес-сервісів.....	153
4.3.4. Підсистема інтеграції у середовищі проєктування та виконання бізнес-сервісів.....	157
4.3.5. Підсистема зберігання та отримання даних в середовищі проєктування та виконання бізнес-сервісів.....	158

4.4. Експериментальні дослідження параметрів засобів моніторингу виконання кроків бізнес-процесів	162
4.5. Розробка рекомендацій по впровадженню запропонованих моделей та методів.....	167
Висновки до розділу 4	171
ВИСНОВКИ.....	173
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	176
ДОДАТОК А	195
ДОДАТОК Б.....	198

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ПНОП	платформи з низьким обсягом програмування (low-code, no-code)
ЦП	Центральний процесор (CPU)
ОП	Оперативна пам'ять (RAM)
В/В	операції введення-виведення, I/O (Input/Output)
БП	бізнес-процес
МБП	модель бізнес-процесу
УРНП	угода про рівень надання послуг, SLA (Service Level Agreement)
ЦПРС	Цільовий показник рівня сервісу SLO (Service Level Objective)
ПРС	Показник рівня сервісу SLI (Service Level Indicator)

ВСТУП

Актуальність теми.

Актуальність теми дослідження зумовлена необхідністю підвищення ефективності виконання бізнес-сервісів в умовах зростання складності бізнес-процесів, динамічного навантаження та ресурсних обмежень. Використання формалізованих і автоматизованих підходів сприяє стабільності, передбачуваності, своєчасному виявленню відхилень і дотриманню цільових показників рівня сервісу в сучасних інформаційних системах.

Наявні підходи нерідко демонструють обмеження під час роботи в умовах нерівномірного навантаження, оскільки вони не забезпечують повного циклу контролю використання ресурсів, своєчасного виявлення відхилень та підтримання показників щодо угод про рівень надання послуг. Недоліки сучасних платформ управління бізнес-процесами проявляються під час інтеграції різних моделей, конфігурацій і сценаріїв виконання, коли стандартні засоби моніторингу та планування не забезпечують належного рівня керованості системою. Використання лише базових інструментів є недостатнім, оскільки вони не контролюють повною мірою поведінку всієї платформи в умовах динамічного навантаження.

Питання розроблення, аналізу та оптимізації складних програмних і технічних систем розглядалися у працях багатьох вітчизняних і зарубіжних науковців. До кола дослідників, чії роботи охоплюють окремі аспекти автоматизації виконання процесів, управління ресурсами, оптимізації програмних систем та організації складних інформаційних середовищ, належать О.А. Машков, Ю.П. Зайченко, В.Г. Лазарєв, О.В. Коваль, В.С. Семеніхін, В.Ф. Колчин, В.К. Попков, О.В. Барабаш, А.П. Мусієнко, Г.А. Кучук, Д.М. Обідін, О.А. Кононов, В.Є. Мухін, Я.І. Корнага, В. Цветков, S. Newman, K. Hoffman та інші.

В наведених наукових працях, присвячених проблематиці проєктування та оптимізації програмних систем, розглядалися питання організації програмних рішень, вибору критеріїв їх ефективності та підвищення якості функціонування. У цих роботах обговорювалися підходи до аналізу структури програмних систем,

удосконалення обчислювальних процедур, забезпечення надійності виконання операцій і раціонального використання доступних ресурсів. Також увага приділялася питанням синхронізації обробки задач, підтриманню стабільності функціонування та оцінюванню продуктивності програмних систем за різних умов роботи.

Окремий напрям стосувався підвищення стійкості програмних засобів до помилок та впливу змін у параметрах середовища, що впливають на якість виконання операцій. У ряді робіт аналізувалися методи зменшення затримок у обробці даних та покращення показників продуктивності шляхом оптимізації алгоритмів і структури програмних компонентів. При цьому зроблено акцент на забезпеченні передбачуваності роботи та підтриманні стабільних характеристик виконання.

У сучасній теорії та практиці автоматизації проєктування й виконання бізнес-процесів спостерігається протиріччя між потребою у стабільному та передбачуваному функціонуванні бізнес-сервісів і тими можливостями, які надають наявні методи та платформи для забезпечення керованого виконання процесів у змінних умовах навантаження. Одним із ключових викликів є те, що інструменти швидкого моделювання та конфігурування процесів не завжди дозволяють забезпечувати необхідні показники продуктивності, ресурсної ефективності та дотримання встановлених параметрів виконання.

Для усунення зазначеного протиріччя у дисертаційній роботі розглядається наукове завдання *розроблення моделі та методів керованого виконання бізнес-процесів, що поєднують засоби моделювання, моніторингу використання ресурсів та адаптивного планування задач. Запропонований підхід спрямований на підвищення стабільності роботи бізнес-сервісів і забезпечення узгодженого виконання процесів відповідно до встановлених вимог.*

Метою дисертаційної роботи є підвищення ефективності виконання кроків бізнес-процесів у платформах автоматизації проєктування та виконання бізнес-сервісів в умовах змінного навантаження.

Основні завдання дослідження включають:

1. Провести порівняльний аналіз методів і засобів організації сучасних платформ автоматизації проєктування та виконання бізнес-сервісів для визначення особливостей та проблем їх реалізації.
2. Дослідити механізми моніторингу виконання кроків бізнес-процесів у платформі автоматизації проєктування та виконання бізнес-сервісів, визначити вимоги до телеметрії, інструментованого контролю ресурсів та виявлення порушень стабільності.
3. Розробити метод реактивного мульти-токенового планування задач за дедлайном старту, який дозволяє знизити ризик перевантаження середовища та зменшити частку порушень дедлайнів старту в умовах змінного та пікового навантаження.
4. Розробити метод планування задач за правилом "перший прийшов - перший оброблений", який дозволяє зменшити ризик аварійних збоїв середовища виконання, зокрема, спричинених перевищенням обмеження оперативної пам'яті.
5. Розробити модель моніторингу та виконання кроків бізнес-процесів у ресурсно-обмеженому середовищі.
6. Розробити програмне забезпечення системи управління та перевірки виконання кроків бізнес-процесів у платформі автоматизації проєктування та виконання бізнес-сервісів.

Об'єктом дослідження є процес виконання кроків бізнес-процесів у платформах автоматизації проєктування та виконання бізнес-сервісів в умовах змінного навантаження.

Предметом дослідження є моделі, методи та програмні засоби підвищення ефективності виконання кроків бізнес-процесів, що забезпечують моніторинг використання ресурсів, стабільність роботи бізнес-сервісів у платформах автоматизації проєктування та виконання бізнес-сервісів.

Методи дослідження. Дисертаційне дослідження виконано з використанням методів математичного та імітаційного моделювання, теорії масового обслуговування і теорії автоматичного керування, а також методів порівняльного

та експериментального аналізу виконання бізнес-сервісів в умовах змінного навантаження. Це дало змогу дослідити особливості виконання кроків бізнес-процесів, оцінити затримки, пропускну здатність, ресурсоспоживання та стабільність функціонування системи.

Наукова новизна одержаних результатів.

Вперше розроблено метод реактивного мульти-токенового планування задач за дедлайном старту, який відрізняється від існуючих поєднанням етапів перевірки ресурсів для запуску задачі за мультиресурсною токенною моделлю з резервуванням "видано/повернуто" та вибором допустимої задачі з найменшим дедлайном старту, зазначений метод дозволяє знизити ризик перевантаження середовища та зменшити частку порушень дедлайнів старту в умовах змінного та пікового навантаження.

Удосконалено метод планування задач за правилом "перший прийшов - перший оброблений", який відрізняється введенням механізму ресурсного допуску задач перед запуском і правилом відкладеного старту за нестачі ресурсів та вибором допустимої задачі зі збереженням черговості обробки із очікуванням вивільнення ресурсів, що дозволяє зменшити ризик аварійних збоїв середовища виконання, зокрема спричинених перевищенням обмежень оперативної пам'яті.

Удосконалено модель моніторингу та виконання кроків бізнес-процесів у ресурсно-обмеженому середовищі, яка відрізняється від існуючих тим, що подає крок процесу як задачу з багатовимірними ресурсними потребами та часом постановки в чергу і дедлайном старту, який відокремлюється правилом пріоритету від механізму багаторесурсної перевірки здійсненності запуску, що задає метод вибору задачі із існуючої черги як фіксований параметр моделі та дозволяє порівняння альтернативних пріоритетів за показником частки порушень дедлайнів старту.

Набула подальшого розвитку інформаційна технологія автоматизації проєктування та виконання бізнес-процесів у частині багаторівневої інтеграції засобів моделювання, виконання, моніторингу, аналітики та керування, що відрізняється реалізацією у замкненому контурі "виконання - телеметрія -

аналітика - керування - вплив на виконання”, розробленого методу реактивного мульти-токенового планування задач за дедлайном старту та удосконалено методу планування задач за правилом "перший прийшов - перший оброблений", що дозволяє підвищити обґрунтованість підтримання стабільності виконання процесів за змінного навантаження.

Наукове значення дослідження полягає у розвитку методів, моделей та інформаційних технологій автоматизації проектування та виконання бізнес-процесів у ресурсно-обмежених динамічних середовищах, зокрема в частині дедлайн-орієнтованого планування задач, багаторесурсного контролю запуску, моніторингу виконання та підтримання стабільності бізнес-сервісів за змінного навантаження.

Практичне значення отриманих результатів полягає у створенні інформаційної системи для моніторингу й керованого виконання бізнес-процесів за змінного навантаження та обмежених ресурсів. Результати тестування підтверджують її придатність для оцінювання ресурсів, цільового показника рівня сервісу, порівняння методів планування й удосконалення платформ автоматизації проектування та виконання бізнес-сервісів.

У результаті порівняння базової та запропонованої конфігурації на промисловому середовищі показало зменшення частки порушених дедлайнів старту у сценаріях із 5000 задач. У першому сценарії цей показник знизився з 14,56% до 5,48%, тобто на 9,08%, у другому - з 67,88% до 25,81%, тобто на 42,07%. Найбільше зменшення частки порушень спостерігається за обмеженого обсягу оперативної пам'яті. Зі збільшенням доступної пам'яті рівень порушень дедлайнів зменшується в обох конфігураціях, однак запропонована конфігурація забезпечує нижчі значення частки порушених дедлайнів, що підтверджує доцільність використання запропонованої конфігурації.

Таким чином, мета дисертаційного дослідження щодо підвищення ефективності виконання кроків бізнес-процесів у платформах автоматизації проектування та виконання бізнес-сервісів в умовах змінного навантаження досягнута, а всі поставлені наукові завдання виконані в повному обсязі. Отримані

наукові результати становлять внесок у розвиток інформаційних технологій у частині виконання бізнес-процесів, розроблення методів планування.

Перспективними напрямками подальших досліджень є удосконалення методів контролю ресурсів, розроблення засобів адаптивного планування кроків процесів та розширення моделей оцінювання ефективності виконання бізнес-процесів у змінних технічних та експлуатаційних умовах.

Отримані у дисертації результати є обґрунтованими, оскільки при їх виведенні коректно використано відповідний математичний апарат. Крім того, теоретичні висновки перевірено експериментально та вони узгоджуються з результатами проведених досліджень.

У першому розділі виконано узагальнений аналіз методів, засобів і технологічних підходів, що застосовуються в платформах автоматизації проєктування та виконання бізнес-сервісів, з акцентом на low-code/no-code підхід, використання візуального моделювання та модельно-орієнтованих принципів побудови процесних рішень.

На основі аналізу показано, що сучасні платформи цього класу є багаторівневими інформаційними системами, для яких критичною є узгодженість між моделями бізнес-процесів, механізмами їх виконання та інструментами контролю параметрів функціонування; при цьому зростання складності цифрових сервісів і поява ресурсомістких задач підвищують вимоги до стабільності, керованості та передбачуваності роботи таких платформ.

Встановлено, що значна частина наявних рішень зосереджується на оптимізації окремих компонентів (моделювання, управління потоками задач або моніторинг ресурсів) без повної інтеграції в єдиний контур керування, а також недостатньо враховує динамічне навантаження та багатовимірні ресурсні обмеження, що може призводити до деградації продуктивності, зростання затримок, накопичення черг і порушення часових вимог.

Окремо підкреслено, що за відсутності формалізованих моделей ресурсних обмежень і угод рівня надання послуг (УРНП) можливі непередбачувана деградація сервісів; водночас домінування реактивних механізмів керування ресурсами та

нестача превентивного планування обмежують здатність підтримувати задані показники якості у довгостроковій перспективі.

На цій основі сформульовано наукову проблему розроблення моделей і методів, що забезпечують узгоджене ресурсно-орієнтоване планування та моніторинг виконання кроків бізнес-процесів у динамічному середовищі з урахуванням часових вимог і багатовимірних обмежень ресурсів

У другому розділі розглянуто механізми моніторингу та забезпечення стабільного і безпечного виконання бізнес-процесів у платформах автоматизації проектування та виконання бізнес-сервісів в умовах змінного навантаження та обмежених обчислювальних ресурсів. Уточнено, що вимоги до якості сервісу доцільно визначати не лише через латентність і пропускну здатність, а також через передбачуваність запуску кроків, зокрема дотримання дедлайнів старту, і контроль зростання черги та використання ресурсів у межах заданих обмежень.

Для ресурсно чутливих середовищ запропоновано розрізняти жорсткі обмеження (гранично допустимі значення використання ресурсів і місткості черги) та м'які цільові рівні, що використовуються як критерії для раннього виявлення ризиків і ініціювання превентивних дій.

Засоби підтримки стабільності систематизовано за п'ятьма групами: керування навантаженням, керування ресурсами, забезпечення узгодженості та стійкості, моніторинг/логування/трасування, а також відновлення і самозцілення.

Набір показників стабільності включає перцентильні характеристики латентності (p50/p95/p99), пропускну здатність, частку відмов і повторів, метрики використання ресурсів, а також параметри черги (довжина та вік). Для задач зі стартовими дедлайнами додатково враховуються затримка запуску та частка порушень дедлайну старту, зокрема випадки відхилення задач через переповнення черги.

У розділі визначено вимоги до системи моніторингу, що охоплюють спостереження виконання BPMN-кроків у реальному часі, збір технічних метрик (центрального процесора, оперативної пам'яті, операцій вводу-виводу і параметри черг), журналювання та трасування з кореляцією "екземпляр процесу - технічні

події”, а також перевірки цільових показників рівня сервісу, сповіщення і підтримку аналітичних сценаріїв (аналіз першопричин, прогнозування деградацій, виявлення аномалій).

Розглянуто агентський та інструментований підходи до збирання телеметрії. Зазначено, що інструментована телеметрія є необхідною у випадках, коли потрібно фіксувати надходження задач у чергу, момент запуску, причини відхилення (зокрема переповнення черги) та ресурсний профіль виконання.

Визначено склад моделі телеметричних даних (події, метрики та контекстні атрибути для кореляції), а також обґрунтовано принцип агрегування “від загального до деталізації”, за якого первинний аналіз виконується за агрегованими показниками, а деталізація застосовується після спрацювання тригерів відхилень.

Запропоновано концептуальну модель підсистеми моніторингу та конвеєра телеметрії (збір - агрегація/нормалізація - аналітична обробка), що передбачає збереження та візуалізацію результатів, а також окремий блок оцінювання виконання цільових показників рівня сервісу і реагування. Для задач зі стартовими дедлайнами в цьому блоці враховуються порушення дедлайну старту та відхилення задач через переповнення черги.

Додатково уточнено організацію контуру зворотного зв'язку платформи та роль модуля контролю, який функціонує в циклі “спостереження - оцінювання стану - вибір керуючої дії - виконання - аудит” і перетворює телеметрійні сигнали на керуючі впливи (зокрема дроселювання прийому/виконання, зміну рівня паралельності, обмеження операцій вводу-виведення, пріоритезацію класів задач) для забезпечення керованої деградації з фіксацією прийнятих рішень.

У третьому розділі розроблено формальну модель ресурсно-орієнтованого виконання бізнес-процесів у платформах автоматизації проєктування та виконання бізнес-сервісів та обґрунтовано необхідність побудови алгоритмів планування, здатних забезпечити стабільність системи в умовах обмежених ресурсів і динамічного потоку задач. На першому етапі дослідження сформовано математичне підґрунтя опису поведінки бізнес-процесів через множину атомарних задач, кожна з яких має визначені характеристики споживання ресурсів, дедлайн

початку та тривалість виконання. Введено формальні умови ресурсної допустимості та стохастичної стійкості, які визначають межі, в яких система здатна забезпечувати гарантований запуск задач без порушення угоди рівня надання послуг. Показано, що порушення цих умов призводить до неконтрольованого зростання черги, перевантаження ресурсів та виникнення затримок виконання, характерних для платформ автоматизації проєктування та виконання бізнес-сервісів під навантаженням.

На основі сформульованих теоретичних положень розроблено гібридний алгоритм планування, який поєднує статичну оцінку завантаженості системи з динамічним реактивним перерозподілом задач. Введено механізм ресурсних токенів, що дозволяє враховувати не лише часову компоненту задачі, а й її мульти-ресурсний профіль, включно з використанням ЦП, оперативної пам'яті та операцій введення-виведення. Алгоритм визначає коефіцієнт готовності задач до виконання, який узагальнює пріоритет, наближеність дедлайну та ступінь відповідності ресурсним обмеженням. Такий метод забезпечує адаптивний вибір задач, мінімізує кількість порушень дедлайнів і дозволяє запобігати локальним перевантаженням.

У розділі наведено докладну схему роботи гібридного алгоритму планування та описано механізм його переходу у реактивний режим у випадках пікового навантаження. Показано, що адаптивне регулювання вагових коефіцієнтів дозволяє алгоритму реагувати на зміну навантаження та забезпечувати баланс між пропускнуою здатністю та стабільністю виконання. Розглянуто приклади застосування запропонованої моделі й проаналізовано її поведінку у типових сценаріях, характерних для бізнес-процесів у платформах автоматизації проєктування та виконання бізнес-сервісів.

Наведено обґрунтування того, що розроблена модель і алгоритм планування утворюють основу для створення інтегрованої системи керування виконанням кроків бізнес-процесів, яка враховує як детерміновані, так і стохастичні властивості потоків задач. Показано, що результати третього розділу є теоретичною передумовою для подальшої реалізації програмного фреймворку, описаного в четвертому розділі, та слугують математичною базою для побудови ресурсно-

орієнтованих методів контролю й адаптивного керування виконанням бізнес-сервісів.

У четвертому розділі описано програмно-технічну реалізацію спеціалізованого фреймворку та інформаційної технології контрольованого виконання кроків бізнес-процесів у платформі проєктування та виконання бізнес-сервісів. Фреймворк орієнтовано на відтворюване проведення експериментів для дослідження правил диспетчеризації та механізмів контролю ресурсів у середовищах із обмеженнями обчислювальної потужності та оперативної пам'яті. Вхідними даними є моделі бізнес-процесів, конфігурація середовища, параметри формування навантаження та вибрані правила диспетчеризації; вихідними даними є показники якості планування, частка порушень часових обмежень і значення використання ресурсів. Робочий процес організовано як замкнений цикл: ініціалізація середовища - диспетчеризація та виконання - вимірювання стану ресурсів - аналіз результатів.

Реалізацію методів планування інтегровано в систему за інтерфейсно-орієнтованим підходом: окремо виділено керувальний модуль планування, який координує запуск і зупинку виконання та регулювання кількості паралельних виконавців, а правила вибору задач інкапсульовано у взаємозамінних реалізаціях політик. Контроль багатовимірних ресурсних обмежень забезпечено механізмом виділення ресурсних токенів, а зворотний зв'язок реалізовано через модулі моніторингу, що формують знімки стану ресурсів для прийняття рішень у контурі керування. Результати експериментів зберігаються у табличному текстовому форматі та підлягають подальшій обробці й порівнянню за фіксованими параметрами прогону (кількість задач, тип навантаження, тривалість, ресурсні ліміти), що підтримує відтворюваність та коректність зіставлення.

Для взаємодії з підсистемами фреймворку передбачено графічний інтерфейс із модульною структурою, який взаємодіє з ядром системи через прикладний інтерфейс обміну даними. У розділі наведено опис взаємодії ключових компонентів, зокрема шини задач, механізму ресурсних токенів, сенсорів

моніторингу та засобів збору телеметрії, а також зазначено можливість розгортання системи як у локальному виконанні, так і в контейнеризованому середовищі.

Перевірку застосовності запропонованих моделей та алгоритмів виконано шляхом серії експериментальних досліджень із використанням розробленого фреймворку. Оцінювання проводилось кількісно, зокрема за часткою порушень часових обмежень, що визначається як відношення кількості кроків із зафіксованим порушенням до загальної кількості кроків у відповідному прогоні за однакових умов навантаження та ресурсних обмежень. На підставі отриманих результатів сформульовано узагальнені рекомендації щодо застосування запропонованого підходу в практичних середовищах, включно з поєднанням планування та моніторингу, поетапним впровадженням і технічними заходами для забезпечення стабільної роботи в умовах змінного навантаження.

Особистий внесок здобувача.

Усі наукові результати, наведені в дисертаційному дослідженні, отримані здобувачем особисто.

У роботі [1] автором розроблено алгоритмічний опис удосконаленого методу, реалізовано експериментальне середовище та програмну реалізацію зазначених методів, виконано аналіз показників продуктивності, порушень цільового показника сервісу та використання ресурсів, що дозволило вперше обґрунтувати удосконалений метод реактивного мульти-токенового планування задач за правилом "перший прийшов - перший оброблений" та обґрунтувати метод реактивного мульти-токенового планування задач за дедлайном старту в ресурсно-обмеженому середовищі.

У роботі [2] (одноосібна публікація) автором виконано аналіз наукових публікацій за темою статті, запропоновано математичну модель процесу планування, що дозволило удосконалити модель моніторингу та виконання кроків бізнес-процесів у ресурсно-обмеженому середовищі.

У роботі [3] автор розробив опис параметрів налаштування й алгоритм обслуговування даних журналу бізнес-процесів, що забезпечило розвиток інформаційної технології автоматизації проєктування та виконання бізнес-процесів

через реалізацію механізму адаптивного обслуговування й оцінювання ефективності за показниками продуктивності серверного середовища.

У роботі [4] автор розробив загальні схеми базового і запропонованого підходів для підсистеми логування в системі виконання бізнес-процесів, здійснив параметричне удосконалення підсистеми логування, обґрунтував застосування буфера з агрегацією повідомлень і використання даних Query Store для оцінювання ефективності запропонованого підходу, що забезпечило подальший розвиток інформаційної технології автоматизації проєктування та виконання бізнес-процесів на основі даних моніторингу та оцінювання поточного стану середовища

У роботі [5] автор розробив загальну схему підсистеми обробки задач, метод визначення кількості фонових обробників та запропонував конфігурацію кількості одночасно виконуваних фонових операцій зазначених підсистем. Загальна схема підсистеми обробки задач відображає виконання кроків бізнес-процесів у спільному ресурсно-обмеженому середовищі та враховує взаємодію підсистем, які використовують спільні обчислювальні ресурси, що дозволило удосконалити модель моніторингу та виконання кроків бізнес-процесів в умовах змінного навантаження. Алгоритм визначення кількості фонових обробників ґрунтується на аналізі стану системи, графіків навантаження, журналів подій і зворотного зв'язку щодо реакції середовища на зміну конфігурації, що забезпечило подальший розвиток інформаційна технологія автоматизації проєктування та виконання бізнес-процесів на основі даних моніторингу та оцінювання поточного стану середовища.

Апробація результатів дисертації.

Результати, отримані в межах цього дослідження та відображені в дисертації, були представлені у доповідях і пройшли обговорення під час участі в таких наукових конференціях:

1. Корнага Я. І., Лемешко В.А. RMRT-EDF: реактивний мультиресурсний токен-планувальник для гарантованого запуску задач у реальному часі // XIII Міжнародна науково-практична конференція з інформаційних систем та

технологій. 21-22 травня 2025 року. Матеріали конференції, ст. 35-38; Форма участі – дистанційно.

Автор розробив алгоритмічне та архітектурне представлення методу, виконав його імітаційну перевірку та аналіз показників ефективності, що дозволило вперше обґрунтувати метод реактивного мульти-токенового планування задач за дедлайном старту в ресурсно-обмеженому середовищі.

Публікації. Основні наукові результати дисертаційної роботи опубліковані у 5 наукових працях, серед яких 5 наукових статей у фахових журналах [1 – 5], затверджених МОН України, а також 1 публікації в матеріалах наукових конференцій.

Структура та обсяг дисертації.

Загальний обсяг 201 сторінок машинописного тексту, що складається зі вступу, чотирьох розділів, висновків, переліку використаних джерел із 160 найменувань та додатків. Дисертація містить 17 рисунків, 10 таблиць і 2 додатки.

РОЗДІЛ 1 ПОРІВНЯЛЬНИЙ АНАЛІЗ СУЧАСНИХ ПЛАТФОРМ АВТОМАТИЗАЦІЇ ПРОЄКТУВАННЯ ТА ВИКОНАННЯ БІЗНЕС-СЕРВІСІВ

1.1. Аналіз організації, проєктування та виконання бізнес-процесів у сучасних платформах з низьким обсягом програмування

Сучасний етап розвитку цифрової економіки характеризується істотним ускладненням бізнес-процесів, зростанням їхньої динамічності та підвищенням вимог до швидкості адаптації організацій до змін зовнішнього та внутрішнього середовища. У цих умовах бізнес-процеси дедалі частіше реалізуються у формі комплексних бізнес-сервісів, які інтегрують інформаційні ресурси, програмні компоненти та організаційні правила у межах розподілених цифрових платформ. Дослідження, присвячені цифровій трансформації управління, свідчать, що традиційні підходи до автоматизації бізнес-процесів, засновані на розробленні індивідуальних програмних рішень, не завжди відповідають сучасним вимогам щодо гнучкості, швидкості впровадження та вартості супроводу [6; 7; 9].

Одним із поширених підходів, що розглядається у літературі, є розвиток платформ з низьким обсягом програмування, low-code (ПНОП) та платформ без програмування (no-code), які орієнтовані на спрощення процесу створення та модифікації бізнес-застосунків. У наукових публікаціях такі платформи розглядаються як інструменти, що дозволяють скоротити розрив між бізнес-вимогами та їх технічною реалізацією шляхом використання візуального моделювання, декларативних описів логіки та повторно використовуваних компонентів [6; 8; 10]. У цьому контексті ПНОП розглядаються не лише як технологічне рішення, а як організаційно-методичний підхід до управління бізнес-процесами.

Еволюція платформ з низьким обсягом програмування тісно пов'язана з розвитком систем управління бізнес-процесами (BPMS) та концепцією розробки на основі моделей (model-driven development). У класичних BPMS модель бізнес-

процесу, зазвичай представлена у нотації BPMN, виконувала роль формалізованого опису, на основі якого здійснювалася подальша реалізація програмної логіки або її часткове виконання процесним рушієм. Проте, як зазначають дослідники, у таких системах модель процесу рідко була повністю самодостатнім виконуваним артефактом і часто вимагала значного обсягу ручного програмування, що ускладнювало швидке внесення змін і підвищувало залежність від ІТ-фахівців [6; 11].

ПНОП розвинули цю парадигму, трансформували модель бізнес-процесу у центральний елемент життєвого циклу автоматизації. У таких платформах модель процесу поєднується з описами бізнес-правил, форм користувацької взаємодії та механізмів інтеграції, утворюючи цілісну модель бізнес-сервісу, яка може безпосередньо виконуватися або транслюватися у виконувану логіку [10; 12; 15]. Це дозволяє розглядати бізнес-процес не лише як аналітичну конструкцію, а як активний компонент інформаційної системи, що визначає її поведінку у реальному часі.

Організація проєктування бізнес-процесів у сучасних платформах з низьким обсягом програмування ґрунтується на принципі підвищеного рівня абстракції. Користувачеві надаються інструменти для роботи з високорівневими елементами процесної логіки - подіями, завданнями, умовами, ролями та сервісними викликами - без необхідності безпосереднього програмування алгоритмів обробки даних. За результатами оглядових досліджень, такий підхід сприяє суттєвому скороченню часу розроблення та спрощує комунікацію між бізнес-аналітиками та технічними спеціалістами [6; 7; 13].

Важливою особливістю підходу з низьким обсягом програмування є розширення кола учасників процесу автоматизації за рахунок залучення так званих бізнес-користувачів (citizen developers), які не мають глибокої підготовки у сфері програмування, але володіють знаннями предметної області. У низці робіт підкреслюється, що залучення бізнес-користувачів дозволяє організаціям швидше реагувати на зміни бізнес-середовища та зменшувати навантаження на ІТ-підрозділи [10; 13; 14]. Водночас дослідники звертають увагу на необхідність

формування чітких механізмів управління та контролю, оскільки неконтрольоване використання інструментів з низьким обсягом програмування може призводити до фрагментації процесів і зростання технічного боргу [8].

Проектування бізнес-процесів у ПНОП зазвичай здійснюється в межах інтегрованого середовища, яке поєднує редактори процесів, конструктори інтерфейсів користувача та засоби опису бізнес-правил. Така інтеграція забезпечує узгодженість між різними аспектами бізнес-логіки, проте водночас посилює залежність процесної моделі від конкретної платформи реалізації. У наукових публікаціях зазначається, що платформа-специфічні розширення моделей ускладнюють їх перенесення та повторне використання в інших середовищах, що є одним з обмежень підходу з низьким обсягом програмування [12; 19].

Особливе місце в аналізі сучасних ПНОП займає питання виконання бізнес-процесів. На відміну від традиційних підходів, де виконання процесу часто було жорстко пов'язане з програмною логікою, у ПНОП реалізуються різні стратегії виконання моделей. Найпоширенішою є інтерпретація процесної моделі у реальному часі, коли рушій процесів безпосередньо зчитує модель та керує переходами між її елементами. Такий підхід забезпечує високу гнучкість змін, але може впливати на продуктивність виконання [6; 11].

Поряд з інтерпретаційним підходом у сучасних ПНОП застосовуються механізми трансляції або часткової компіляції моделей у проміжний код чи набір сервісних викликів. Дослідники відзначають, що гібридні підходи, які поєднують інтерпретацію та компіляцію, дозволяють досягти компромісу між гнучкістю та ефективністю виконання бізнес-процесів [15; 26; 27]. Вибір конкретної стратегії виконання має безпосередній вплив на масштабованість бізнес-сервісів і визначає можливості їх використання у середовищах з високим навантаженням.

Організація виконання бізнес-процесів у ПНОП тісно пов'язана з використанням подієвих механізмів та асинхронної взаємодії компонентів. Такі механізми дозволяють реалізовувати реактивну поведінку бізнес-процесів і забезпечувати їх адаптацію до змін стану зовнішніх систем [18; 27; 29]. Разом із тим, зростання складності виконання підвищує вимоги до прозорості та

керованості процесів, що не завжди повною мірою забезпечується сучасними рішеннями із низьким обсягом програмування.

Можна відзначити, що платформи з низьким обсягом програмування відіграють важливу роль у трансформації підходів до організації, проектування та виконання бізнес-процесів. Вони забезпечують суттєве скорочення циклу автоматизації та розширюють участь бізнес-користувачів у розробленні процесних рішень. Водночас наукові публікації вказують на наявність низки обмежень, пов'язаних з керованістю виконання, залежністю від платформ та необхідністю інтеграції процесного підходу з більш формалізованими механізмами управління. Ці аспекти створюють передумови для подальшого аналізу питань управління ресурсами та стабільності виконання бізнес-сервісів [6–8; 10; 33–35]

Поняття малокодових та безкодових платформ не є однозначними й охоплюють кілька різних класів платформ, які відрізняються за рівнем абстракції, глибиною підтримки процесного підходу та можливостями виконання бізнес-процесів. У роботах [6; 7; 20] підкреслюється, що безкодові платформи орієнтовані переважно на створення простих прикладних рішень і автоматизацію окремих завдань, тоді як ПНОП надають ширші можливості для формалізації складних бізнес-процесів і їх інтеграції в корпоративні інформаційні системи. Саме малокодові BPM-платформи найчастіше розглядаються як розвиток класичних BPMS, адаптований до умов швидкої цифрової трансформації.

Організація бізнес-процесів у малокодових середовищах характеризується поєднанням централізованих і децентралізованих підходів до проектування. З одного боку, платформи пропонують уніфіковані моделі процесів, стандартизовані нотації та спільні репозиторії артефактів, що сприяє узгодженості процесних рішень на рівні організації [12]. З іншого боку, завдяки доступності інструментів моделювання значна частина ініціатив з автоматизації може виникати безпосередньо у бізнес-підрозділах, що змінює традиційну роль ІТ-служб і потребує нових моделей взаємодії між бізнесом та ІТ [8; 13].

У наукових дослідженнях зазначається, що така трансформація організаційних ролей має як позитивні, так і проблемні аспекти. Позитивним є скорочення часу

між виявленням потреби в автоматизації та її реалізацією, а також підвищення відповідності процесних рішень реальним бізнес-вимогам [10; 14]. Водночас, звертається увага на ризики втрати цілісності процесної архітектури, дублювання функціональності та ускладнення подальшого супроводу, якщо відсутні чітко встановлені правила управління і архітектурні обмеження [8; 15].

Проектування бізнес-процесів у сучасних ПНОП дедалі частіше ґрунтується на принципах повторного використання та композиції. У роботах [14; 15] підкреслюється роль шаблонів процесів, попередньо налаштованих компонентів і бібліотек бізнес-правил, які дозволяють прискорити розроблення та забезпечити певний рівень стандартизації. Разом із тим, науковці зазначають, що надмірна залежність від готових компонентів може обмежувати можливості адаптації процесів до специфічних вимог організації та призводити до компромісних рішень у складних сценаріях.

Суттєвим напрямом розвитку ПНОП є інтеграція модельного підходу до розробки із сучасними засобами штучного інтелекту. У низці публікацій розглядаються можливості використання AI для автоматизованого проектування процесів, зокрема на основі аналізу текстових описів, журналів виконання або історичних даних [16; 18; 19]. Такі підходи потенційно знижують когнітивне навантаження на користувачів і розширюють можливості залучення нефахових розробників до автоматизації бізнес-процесів. Водночас автори наголошують, що поєднання AI-засобів із low-code середовищами ускладнює питання інтерпретації та контролю прийнятих рішень. Автоматично згенеровані або рекомендовані моделі процесів можуть бути складними для формального аналізу та перевірки на відповідність внутрішнім регламентам і політикам організації [18; 25]. Це підсилює актуальність досліджень, спрямованих на забезпечення прозорості та керованості процесного виконання у ПНОП.

Виконання бізнес-процесів у таких платформах зазвичай здійснюється на основі процесних рушіїв, які підтримують паралельність, асинхронну взаємодію та обробку подій. У роботах [27; 28; 31; 32] зазначається, що подієво-орієнтовані механізми дозволяють підвищити гнучкість виконання процесів і забезпечити їх

адаптацію до динамічних умов. Разом із тим, зростання кількості подій та взаємодій між компонентами підвищує складність управління станами процесів і вимагає більш розвинених засобів моніторингу та аналізу.

Порівняльний аналіз стратегій виконання бізнес-процесів показує, що інтерпретаційні підходи забезпечують максимальну гнучкість, але можуть поступатися за продуктивністю у сценаріях з високим навантаженням. [6; 11; 15; 33–35]

Підходи, засновані на компіляції або трансляції моделей у виконуваних компоненти, дозволяють підвищити ефективність, але обмежують можливості оперативного внесення змін [15; 30; 33]. Більшість сучасних ПНОП реалізують гібридні рішення, поєднуючи обидва підходи залежно від типу процесів і вимог до їх виконання.

Особлива увага приділяється питанням масштабованості та стабільності виконання бізнес-процесів у малокодових середовищах. Підвищений рівень абстракції та універсальність платформ можуть негативно впливати на передбачуваність поведінки систем у пікових режимах роботи [33; 34; 35]. Це проявляється у зростанні затримок, складності прогнозування використання ресурсів та обмежених можливостях тонкого налаштування виконання процесів.

Впровадження малокодових та безкодових платформ у практику управління вимагає адаптації методів моделювання бізнес-процесів до специфіки організаційного та нормативного середовища. Зокрема, удослідженнях [21; 22; 23] аналізуються підходи до формалізації управлінських процесів в умовах цифрової трансформації та підкреслюється необхідність поєднання процесного підходу з вимогами прозорості та підзвітності.

Узагальнюючи аналіз малокодових і безкодових платформ дозволяє виокремити кілька концептуальних ліній, які визначають сучасне розуміння організації та проектування бізнес-процесів у таких середовищах. По-перше, у більшості досліджень ПНОП розглядаються як інструменти, що поєднують процесний підхід із прикладною орієнтацією на швидку реалізацію бізнес-рішень. У цьому контексті вони позиціонуються як компроміс між гнучкістю

індивідуальної розробки та стандартизованістю класичних BPMS, що підтверджується оглядовими роботами та емпіричними дослідженнями [6; 7; 12].

По-друге, важливим аспектом є трансформація ролі процесної моделі у життєвому циклі автоматизації. Якщо в традиційних підходах модель бізнес-процесу виконувала переважно аналітичну та документувальну функції, то в ПНОП вона стає центральним елементом виконання, інтегруючи логіку управління, правила прийняття рішень і механізми взаємодії з користувачами. Наукові публікації підкреслюють, що така еволюція підвищує значущість якості процесного моделювання та потребує більшої формалізації моделей навіть у середовищах, орієнтованих на нефархових користувачів [10; 11; 15].

По-третє, аналіз організації проєктування бізнес-процесів у малокодових середовищах виявляє зростання значення повторного використання та модульності. Використання шаблонів, бібліотек процесів і стандартних компонентів сприяє уніфікації підходів до автоматизації, проте одночасно може знижувати здатність системи до врахування специфічних організаційних контекстів. У наукових роботах зазначається, що баланс між стандартизацією та гнучкістю є одним з ключових викликів при впровадженні ПНОП у великих організаціях [14; 17].

Окрема увага приділяється аспектам організаційного управління процесною автоматизацією в умовах широкої участі бізнес-користувачів. З одного боку, демократизація інструментів проєктування бізнес-процесів дозволяє бізнес-підрозділам самостійно ініціювати та реалізовувати зміни, що підвищує адаптивність організації [13; 14]. З іншого боку, відсутність належних механізмів координації та контролю може призводити до фрагментації процесної архітектури, дублювання функцій і порушення принципів узгодженості інформаційних систем [8; 11].

У контексті виконання бізнес-процесів сучасні ПНОП реалізують різноманітні технічні механізми, які часто приховані від кінцевих користувачів за високорівневими абстракціями. Така прихованість спрощує розроблення, але ускладнює аналіз поведінки процесів у критичних ситуаціях, зокрема за умов

високого навантаження або нестандартних сценаріїв виконання. Дослідники відзначають, що обмежена прозорість внутрішніх механізмів виконання знижує можливості точного прогнозування характеристик бізнес-сервісів [33; 34].

Важливим напрямом досліджень є також інтеграція ПНОП із сучасними архітектурними підходами, такими як мікросервіси, подієво-орієнтовані архітектури та хмарні обчислення. Інтеграція дозволяє підвищити масштабованість і гнучкість виконання бізнес-процесів, але водночас вимагає більш складних механізмів управління взаємодією між компонентами [27; 30]. Для ПНОП це створює додаткові виклики, оскільки необхідно поєднати простоту використання з можливістю налаштування складних архітектурних сценаріїв.

Впровадження ПНОП не усуває потреби в системному процесному мисленні. Навпаки, спрощення інструментів проєктування підвищує ризик поверхневого підходу до моделювання, коли процеси формалізуються без достатнього аналізу їх структури, залежностей і ресурсних обмежень. Без належної методологічної підтримки та навчання користувачів малокодові інструменти можуть призводити до формування неефективних або важкокерованих процесних рішень [6; 22].

У джерелах [21; 22; 23] наголошується на необхідності поєднання процесного підходу з вимогами прозорості, підзвітності та формалізації управлінських процедур, що є важливим для великих організацій і публічного сектору. Узагальнення результатів проведеного аналізу дає змогу сформулювати кілька висновків щодо поточного стану досліджень у сфері малокодової автоматизації бізнес-процесів. По-перше, відзначається значний потенціал ПНОП для прискорення проєктування та впровадження бізнес-процесів. По-друге, обмежена увага до питань керованості виконання процесів, прозорості використання ресурсів і забезпечення стабільності бізнес-сервісів у довгостроковій перспективі. По-третє, більшість наявних підходів зосереджується на функціональних аспектах автоматизації, тоді як нефункціональні вимоги часто не включаються до формалізованого аналізу [24; 33; 35].

1.2. Дослідження особливостей побудови сучасних систем управління ресурсами та угоди рівня надання послуг у платформах проєктування та виконання бізнес-сервісів

Сучасні платформи автоматизації проєктування та виконання бізнес-сервісів функціонують у середовищах, які характеризуються високою динамічністю, гетерогенністю ресурсів і постійною зміною вимог до якості обслуговування. У таких умовах управління ресурсами перестає бути суто технічною задачею і набуває системного характеру, тісно пов'язаного з організацією бізнес-процесів і механізмами їх виконання. Ефективність бізнес-сервісів визначається не лише правильністю процесної логіки, а й здатністю платформи забезпечувати узгоджене використання ресурсів відповідно до визначених сервісних рівнів [36–38].

У рамках процесно-орієнтованих платформ ресурси можуть мати різну природу, включаючи обчислювальні потужності, мережеві та сховищні компоненти, а також людські та організаційні ресурси. Особливістю сучасних бізнес-сервісів є те, що всі ці типи ресурсів залучаються у виконання процесів одночасно і взаємопов'язано. У дослідженнях наголошується, що ігнорування взаємозалежності між різними видами ресурсів призводить до неузгодженого виконання процесів і зниження якості сервісів [39; 40]. Це зумовлює необхідність комплексного підходу до управління ресурсами, який виходить за межі традиційних моделей інфраструктурного менеджменту.

Угода про рівень надання послуг (УРНП) у такому контексті розглядаються як інструмент формалізації нефункціональних вимог до бізнес-сервісів, зокрема вимог щодо часу відгуку, пропускну здатності, доступності та надійності. УРНП, сформовані для хмарних або сервіс-орієнтованих середовищ, здебільшого орієнтовані на окремі сервіси або інфраструктурні компоненти і недостатньо враховують процесну природу бізнес-сервісів [36; 37]. У результаті виникає розрив між формально визначеними сервісними рівнями та фактичною поведінкою складних бізнес-процесів.

Важливою проблемою є те, що більшість сучасних платформ автоматизації реалізують управління ресурсами як фонову функцію, приховану від рівня процесного моделювання. Ресурсні обмеження та механізми їх розподілу часто задаються на рівні платформи або середовища виконання і не відображаються безпосередньо у процесних моделях. Такий спосіб ускладнює аналіз і прогнозування виконання бізнес-процесів, особливо в умовах змінного навантаження або конкуренції за ресурси між паралельними процесами [38–40].

У процесно-орієнтованих платформах управління ресурсами тісно пов'язане з питаннями планування та диспетчеризації виконання. Кожен екземпляр бізнес-процесу конкурує за доступ до ресурсів, що може призводити до затримок, черг і порушення заданих сервісних рівнів. Дослідження показують, що відсутність явних механізмів ресурсного планування на рівні процесної моделі обмежує можливості адаптивного управління виконанням і ускладнює забезпечення стабільності бізнес-сервісів [41; 42].

Зростання складності бізнес-сервісів призводить до необхідності врахування часових аспектів виконання процесів у поєднанні з ресурсними обмеженнями. У наукових публікаціях наголошується, що часові характеристики процесів, такі як дедлайни, часові вікна та допустимі затримки, мають безпосередній вплив на вибір стратегій розподілу ресурсів [43; 44]. Проте у більшості сучасних платформ ці аспекти або реалізуються фрагментарно, або залишаються поза межами формалізованого процесного управління.

Окрему увагу дослідники приділяють подієво-орієнтованим моделям виконання, які широко використовуються у сучасних платформах бізнес-сервісів. Подієва природа виконання дозволяє системам реагувати на зміни стану ресурсів і зовнішніх умов у режимі реального часу, проте, ускладнює прогнозування навантаження і дотримання УРНП [45–47]. У таких умовах статичні УРНП-показники виявляються недостатніми, що зумовлює потребу у динамічних і контекстно-залежних моделях управління ресурсами.

Інтеграція управління ресурсами з процесною логікою виконання залишається однією з найменш розроблених проблем у сучасних платформах автоматизації.

Попри наявність окремих підходів до ресурсно-орієнтованого BPM, більшість рішень не забезпечують цілісної моделі, яка б поєднувала процесні, ресурсні та сервісні аспекти виконання бізнес-процесів [48–50]. Це створює передумови для виникнення конфліктів між вимогами до якості сервісів і фактичними можливостями платформи.

На рівні експлуатації бізнес-сервісів проблема управління ресурсами проявляється у вигляді складнощів із забезпеченням масштабованості та стабільності. У дослідженнях зазначається, що високий рівень абстракції, характерний для сучасних платформ автоматизації, ускладнює точне планування ресурсів і призводить до непередбачуваної поведінки систем у пікових режимах роботи [51–53]. Це, у свою чергу, негативно впливає на здатність дотримуватися визначених УРНП у довгостроковій перспективі.

Управління ресурсами у платформах проєктування та виконання бізнес-сервісів дедалі більше розглядається крізь призму процесно- та сервіс-орієнтованих підходів. У таких платформах бізнес-процеси виступають не лише як логічні послідовності дій, а як активні споживачі ресурсів, поведінка яких безпосередньо впливає на досягнення показників, зафіксованих в УРНП. Ігнорування процесного контексту при управлінні ресурсами призводить до фрагментарних рішень, які не забезпечують узгодженості між функціональною та нефункціональною поведінкою бізнес-сервісів [41–43].

У процесно-орієнтованих системах управління ключовим викликом є поєднання абстрактних моделей бізнес-процесів із конкретними механізмами розподілу ресурсів під час виконання. Процесні моделі зазвичай описують логіку виконання завдань, але не містять явних специфікацій щодо потреб у ресурсах, допустимих затримок або пріоритетів доступу. Відсутність зв'язку між процесами та ресурсами ускладнює реалізацію УРНП-орієнтованого управління, оскільки платформа не має достатньої інформації для прийняття оптимальних рішень у динамічних умовах [44; 45].

Сервіс-орієнтовані підходи до побудови бізнес-сервісів передбачають композицію незалежних компонентів, кожен з яких може мати власні обмеження

щодо продуктивності та доступності. У цьому випадку загальний рівень сервісу бізнес-процесу визначається не лише властивостями окремих сервісів, а й їх взаємодією в межах процесу. Відсутність механізмів агрегування та узгодження УРНП для композицій сервісів призводить до ситуацій, коли формально виконуються локальні угоди, але порушуються глобальні вимоги бізнес-сервісу [36; 41; 46].

Особливої складності набуває управління ресурсами в умовах подієво-орієнтованого виконання бізнес-процесів. Асинхронна обробка подій дозволяє підвищити масштабованість і гнучкість систем, проте ускладнює контроль за використанням ресурсів і дотриманням часових обмежень. У наукових дослідженнях наголошується, що подієві архітектури створюють додаткові труднощі для прогнозування навантаження, оскільки кількість та частота подій можуть змінюватися непередбачувано [45–47]. У таких умовах статичні моделі УРНП виявляються недостатніми, а динамічні механізми управління потребують складних алгоритмів адаптації.

Важливим аспектом є взаємозв'язок між управлінням ресурсами та механізмами планування виконання бізнес-процесів. У процесно-орієнтованих платформах планування зазвичай здійснюється на основі черг, пріоритетів і політик диспетчеризації, які не завжди враховують вимоги УРНП. Відсутність інтеграції УРНП-показників у процес планування унеможливорює гарантування заданих рівнів сервісу, особливо за конкуренції між процесами за обмежені ресурси [48; 49; 76].

У науковій літературі широко розглядається проблема узгодження локальних і глобальних оптимізацій. Управління ресурсами на рівні окремих сервісів або компонентів може забезпечувати локальне підвищення продуктивності, однак водночас негативно впливати на загальну ефективність бізнес-процесу. У цьому контексті наголошується на доцільності розгляду бізнес-процесів як цілісних об'єктів управління, для яких оптимізація має здійснюватися з урахуванням глобальних цілей і обмежень [49–51].

У дослідженнях, присвячених адаптивним системам управління, розглядаються підходи до динамічного перерозподілу ресурсів залежно від поточного стану процесів і виконання УРНП. Показано, що адаптивні механізми можуть підвищувати стійкість бізнес-сервісів до змін навантаження, проте їх реалізація ускладнюється високим рівнем абстракції сучасних платформ автоматизації [48; 52]. У багатьох випадках платформи не надають достатньої прозорості щодо внутрішніх механізмів виконання, що обмежує можливості впровадження таких рішень.

Проблематика управління ресурсами також тісно пов'язана з питаннями моніторингу та аналізу виконання бізнес-процесів. Ефективне УРНП-орієнтоване управління потребує не лише фіксації показників після факту виконання, а й безперервного моніторингу стану процесів і ресурсів у реальному часі [36; 37; 53]. Однак більшість сучасних платформ реалізують моніторинг як допоміжну функцію, що не інтегрована безпосередньо з механізмами управління виконанням.

Окрему групу досліджень становлять роботи, присвячені масштабованості та стабільності виконання бізнес-сервісів у розподілених середовищах. У них показано, що зі зростанням кількості паралельних процесів і складності сервісних композицій різко зростають вимоги до ефективного управління ресурсами. За відсутності формалізованих моделей ресурсних обмежень і УРНП це призводить до деградації сервісів і порушення очікуваних рівнів якості [51–54].

Таким чином, процесно- та сервіс-орієнтовані підходи створюють необхідні передумови для інтеграції управління ресурсами та УРНП у платформах виконання бізнес-сервісів, але на практиці ця інтеграція реалізується фрагментарно. Більшість платформ зосереджуються на окремих аспектах управління, не пропонуючи цілісної моделі, яка б поєднувала процесну логіку, ресурсні обмеження та сервісні показники в єдиному механізмі управління [36–38; 55].

Виконання бізнес-процесів в умовах обмежених ресурсів є однією з ключових проблем сучасних платформ проектування та виконання бізнес-сервісів. На відміну від статичних інформаційних систем, де навантаження може бути відносно передбачуваним, процесно-орієнтовані платформи мають справу з динамічними

потоками екземплярів процесів, кількість і структура яких змінюються у часі. У наукових дослідженнях наголошується, що саме ця динамічність у поєднанні з обмеженими ресурсами створює основні труднощі для забезпечення стабільності виконання та дотримання УРНП [36; 38; 51].

Однією із основних характеристик виконання бізнес-процесів є конкуренція за ресурси між паралельними екземплярами процесів. У процесно-орієнтованих платформах кілька процесів можуть одночасно вимагати доступу до одних і тих самих ресурсів, таких як обчислювальні потужності, канали зв'язку або людські виконавці. Дослідження показують, що відсутність механізмів пріоритизації та явного управління ресурсними конфліктами призводить до накопичення черг, зростання часу очікування та порушення заданих часових обмежень [41; 42; 49]. У таких умовах навіть формально коректно спроектовані бізнес-процеси можуть не відповідати очікуваному сервісним рівням.

Проблема ускладнюється тим, що сучасні платформи автоматизації часто приховують внутрішні механізми розподілу ресурсів за високорівневими абстракціями. З одного боку, це спрощує розроблення і експлуатацію систем, з іншого - знижує прозорість виконання та ускладнює аналіз причин порушення УРНП. Низький рівень прозорості перешкоджає точному моделюванню поведінки системи під навантаженням і обмежує можливості завчасного управління ресурсами [38; 52].

Важливим аспектом виконання бізнес-процесів є вплив часових характеристик на використання ресурсів. Бізнес-процеси часто містять часові обмеження, такі як дедлайни, допустимі затримки або часові вікна виконання окремих завдань. Ігнорування цих характеристик при управлінні ресурсами призводить до ситуацій, коли ресурси використовуються неефективно, а процеси не вкладаються у визначені часові рамки [43; 44]. У контексті УРНП це означає зниження якості сервісів навіть за наявності достатніх ресурсів у середньому.

Особливу складність становить виконання бізнес-процесів у розподілених і хмарних середовищах, де ресурси можуть бути географічно розподілені та динамічно змінювати свою доступність. В таких середовищах традиційні підходи

до управління ресурсами, засновані на статичних конфігураціях, є неефективними [45–47]. Замість цього необхідні адаптивні механізми, здатні реагувати на зміни стану ресурсів у режимі реального часу, що ускладнює реалізацію стабільних УРНП.

Масштабованість виконання бізнес-сервісів безпосередньо пов'язана з можливістю платформи ефективно управляти ресурсами при зростанні кількості процесів і користувачів. Зі збільшенням масштабу системи зростає складність координації між компонентами, а також підвищується ризик виникнення вузьких місць у використанні ресурсів [51; 53]. За відсутності формалізованих моделей ресурсних обмежень і УРНП це призводить до непередбачуваної деградації сервісів.

Стабільність виконання бізнес-процесів розглядається як здатність системи підтримувати задані характеристики якості за різних умов експлуатації. Стабільність не є властивістю окремих компонентів, а формується на рівні всієї платформи, з урахуванням взаємодії процесів, сервісів і ресурсів [51–54]. У цьому контексті УРНП відіграють роль формального інструменту, який дозволяє визначити допустимі межі поведінки системи, але самі по собі не гарантують їх дотримання.

Сучасні платформи автоматизації здебільшого використовують реактивні механізми управління ресурсами, спрямовані на усунення наслідків перевантаження після їх виникнення. Така модель дає змогу відновити роботу системи, проте не запобігає повторному виникненню порушень УРНП у подальшому [36; 37; 55]. Відсутність превентивного планування та прогнозування потреб у ресурсах обмежує можливості довгострокового забезпечення якості бізнес-сервісів.

Аналіз рішень для моніторингу виконання бізнес-процесів свідчить, що більшість платформ зосереджується на збиранні показників продуктивності та доступності на інфраструктурному рівні. Водночас, зв'язок між цими показниками та конкретними екземплярами процесів або завданнями часто залишається

непрямим або недостатньо формалізованим [36; 53]. Це ускладнює виявлення причин порушення УРНП і прийняття обґрунтованих управлінських рішень.

Окремі дослідження розглядають можливості поєднання процесного моніторингу з управлінням ресурсами шляхом використання аналітичних і прогностичних методів. Показано, що аналіз історичних даних виконання процесів може слугувати основою для більш точного прогнозування ресурсних потреб і адаптації стратегій управління [48; 52]. Проте впровадження таких підходів у сучасних платформах ускладнюється обмеженням доступом до внутрішніх механізмів виконання та високим рівнем абстракції.

Аналіз виконання бізнес-процесів за умов обмежених ресурсів засвідчує, що наявні платформи автоматизації не завжди забезпечують рівень передбачуваності та керованості, достатній для дотримання УРНП. Існуючі рішення здебільшого зосереджені на функціональній коректності процесів, тоді як системні аспекти управління ресурсами та стабільності виконання часто залишаються поза межами їх реалізації [36–38; 57].

Проблематика управління ресурсами та угод про рівень надання послуг у платформах автоматизації проектування та виконання бізнес-сервісів має системний характер і не зводиться до окремих технічних або організаційних аспектів. Управління ресурсами в таких платформах формується на перетині процесної логіки, сервісної архітектури та інфраструктурних механізмів виконання, що ускладнює побудову цілісних і формалізованих підходів до забезпечення стабільності бізнес-сервісів [36–38].

УРНП використовуються як засіб формалізації нефункціональних вимог, однак у сучасних платформах їх реалізація переважно обмежується рівнем окремих сервісів або інфраструктурних компонентів. При цьому процесна природа бізнес-сервісів, у межах якої показники якості залежать від узгодженого виконання послідовності завдань і взаємодії кількох компонентів, враховується не повною мірою. Унаслідок цього може спостерігатися розбіжність між формально визначеними сервісними рівнями та фактичною поведінкою бізнес-процесів в умовах експлуатації [36; 37; 41].

У сучасних платформах автоматизації основна увага приділяється гнучкості та швидкості проектування бізнес-процесів, тоді як питання управління ресурсами, як правило, розглядаються окремо від процесного управління. Ресурсні обмеження зазвичай не включаються безпосередньо до процесних моделей і враховуються опосередковано, зокрема через механізми планування та диспетчеризації виконання. Така організація ускладнює прогнозування поведінки бізнес-сервісів і може впливати на ефективність дотримання УРНП в умовах змінного навантаження [38–40, 56].

Одним із чинників, що ускладнюють управління ресурсами, є використання подієво-орієнтованих та асинхронних моделей виконання. Такі моделі сприяють підвищенню адаптивності й масштабованості платформ, водночас можуть ускладнювати контроль за використанням ресурсів і часовими характеристиками виконання процесів. У науковій літературі зазначається, що за таких умов традиційні статичні механізми реалізації УРНП не завжди забезпечують достатню передбачуваність і потребують доповнення динамічними та контекстно залежними засобами управління [45–47].

Проблема масштабованості виконання бізнес-сервісів тісно пов'язана з відсутністю формалізованих моделей ресурсних потреб на рівні процесів. Зі зростанням кількості паралельних екземплярів процесів і складності сервісних композицій платформи стикаються з труднощами у забезпеченні стабільної продуктивності та дотриманні сервісних показників. У дослідженнях зазначається, що реактивні підходи до управління ресурсами, орієнтовані на усунення наслідків перевантажень, не дозволяють гарантувати стабільність бізнес-сервісів у довгостроковій перспективі [51–55].

Проведений аналіз дає змогу окреслити низку обмежень наявних методів моніторингу та контролю виконання бізнес-процесів. Більшість платформ забезпечує збирання показників продуктивності й доступності, однак зв'язок між цими показниками та конкретними екземплярами процесів часто має непрямий характер. Це ускладнює ідентифікацію причин порушення УРНП та зменшує

можливості цілеспрямованого управління виконанням процесів з урахуванням ресурсних обмежень [36; 53; 57].

Узагальнення результатів аналізу свідчить про наявність розбіжності між зростаючими вимогами до стабільності, передбачуваності й керованості бізнес-сервісів та поточними можливостями сучасних платформ щодо формалізованого управління ресурсами і УРНП. З одного боку, такі платформи забезпечують гнучкість і оперативність автоматизації бізнес-процесів, з іншого - їх інструментарій є обмеженим для гарантованого виконання процесів у межах заданих ресурсних і сервісних обмежень [36–38; 57].

1.3. Аналіз архітектурних рішень платформ автоматизації проєктування та виконання бізнес-сервісів

Архітектурні рішення сучасних платформ автоматизації проєктування та виконання бізнес-сервісів визначають не лише функціональні можливості таких систем, а й їх здатність забезпечувати масштабованість, стабільність, керованість і передбачуваність виконання бізнес-процесів. У контексті цифрової трансформації архітектура платформ розглядається як ключовий чинник, що поєднує процесну логіку, сервісну композицію та інфраструктурні механізми виконання в єдину систему. Розвиток архітектурних рішень безпосередньо пов'язаний зі зростанням складності бізнес-сервісів і вимог до їх адаптивності [58–60].

Сучасні платформи автоматизації дедалі частіше базуються на сервіс-орієнтованих і мікросервісних архітектурах, які забезпечують модульність, повторне використання та незалежність компонентів. У таких архітектурах бізнес-сервіси реалізуються як композиції автономних сервісів, взаємодія яких координується за допомогою оркестрації або хореографії. У наукових публікаціях зазначається, що вибір між оркестраційними та хореографічними підходами має істотний вплив на керованість виконання бізнес-процесів і можливості централізованого контролю [61; 62].

Оркестраційні архітектури передбачають наявність центрального компонента, який керує послідовністю викликів сервісів відповідно до визначеної процесної логіки. Це покращує реалізацію складних бізнес-процесів і спрощує моніторинг виконання, проте може створювати вузькі місця та знижувати масштабованість системи. Централізація управління ускладнює адаптацію до динамічних змін навантаження і підвищує залежність стабільності виконання від одного архітектурного елемента [61; 63].

Хореографічні архітектури, навпаки, ґрунтуються на децентралізованій взаємодії сервісів, де кожен компонент самостійно реагує на події та визначає подальші дії. Це сприяє підвищенню масштабованості та стійкості до відмов, але водночас ускладнює забезпечення глобальної узгодженості виконання бізнес-процесів. Відсутність централізованого контролю ускладнює реалізацію комплексних бізнес-правил і контроль дотримання нефункціональних вимог [62; 69].

У платформах автоматизації проєктування та виконання бізнес-сервісів процесні рушії та механізми виконання часто інтегруються з мікросервісною інфраструктурою. Це дозволяє поєднати формальні процесні моделі з гнучкими архітектурними рішеннями, проте створює додаткові виклики щодо узгодження рівнів абстракції. Процесні моделі, розроблені на високому рівні абстракції, не завжди коректно відображають поведінку мікросервісних систем у реальних умовах виконання [64; 67].

Важливим напрямом розвитку архітектур платформ автоматизації є використання подієво-орієнтованих підходів. Подієві архітектури дозволяють системам реагувати на зміни стану процесів і ресурсів у режимі реального часу, що є критично важливим для адаптивного виконання бізнес-сервісів. Поєднання процесного управління з подієво-орієнтованими механізмами сприяє підвищенню гнучкості, але водночас ускладнює аналіз і верифікацію виконання бізнес-процесів [62; 69].

Архітектурні рішення платформ автоматизації тісно пов'язані з принципами розробки та побудови систем на основі моделей. У межах такої організації

процесні, сервісні та моделі даних виступають основними артефактами, на базі яких здійснюється генерація або налаштування виконуваної системи. Модельно-орієнтовані архітектури (model-driven architecture) спрощують узгодження бізнес-вимог із технічною реалізацією, водночас ускладнюючи підтримку моделей в актуальному стані під час еволюції системи [58; 65].

Зростання складності архітектур сучасних платформ автоматизації зумовлює потребу вирішення питань сумісності та узгодженої взаємодії між різними інструментами і середовищами виконання. У дослідженнях зазначається, що відсутність єдиних стандартів і залежність від конкретних платформ обмежують можливості перенесення процесних рішень і повторного використання архітектурних компонентів [58; 59]. Це створює ризики технологічної прив'язаності та ускладнює довгострокове планування розвитку бізнес-сервісів.

Архітектурні рішення також визначають можливості інтеграції платформ автоматизації з зовнішніми системами та технологіями, зокрема з хмарними, edge- та IoT-середовищами. У наукових роботах показано, що інтеграція з такими середовищами розширює функціональні можливості бізнес-сервісів, але одночасно підвищує вимоги до архітектурної узгодженості та управління виконанням [60; 64]. Це особливо актуально для сценаріїв, де бізнес-процеси охоплюють як централізовані, так і розподілені компоненти.

Архітектурні рішення сучасних платформ автоматизації проєктування та виконання бізнес-сервісів формуються під впливом потреб у гнучкості, масштабованості та швидкості адаптації. Водночас у наукових дослідженнях фіксується наявність протиріч між прагненням до децентралізації та необхідністю централізованого контролю виконання бізнес-процесів, що безпосередньо впливає на керованість і стабільність бізнес-сервісів [61–63; 69].

Подальший розвиток архітектурних рішень платформ автоматизації тісно пов'язаний із трансформацією процесних рушіїв (workflow-рушіїв), які виконують роль центрального компонента управління виконанням бізнес-процесів. У сучасних платформах ці компоненти дедалі частіше реалізуються як набори сервісів, інтегровані в мікросервісні або хмарні архітектури. Це дозволяє

підвищити масштабованість і гнучкість виконання, але водночас ускладнює забезпечення узгодженості між процесною логікою та архітектурною реалізацією [60; 67].

Процесні рушії у сервіс-орієнтованих архітектурах зазвичай відповідають за інтерпретацію або виконання процесних моделей, описаних за допомогою нотації та моделі бізнес-процесів (BPMN), нотація та модель прийняття рішень (DMN) або подібних нотацій. Інтеграція BPMN- і DMN-орієнтованих рішень дозволяє поєднати управління потоками робіт і логікою прийняття рішень у межах єдиного архітектурного рішення [63; 64]. Така інтеграція підвищує складність архітектури і вимагає узгодження різних рівнів абстракції, що не завжди повною мірою підтримується сучасними платформами автоматизації.

Використання мікросервісних архітектур для реалізації процесних рушіїв змінює підходи до управління станами виконання бізнес-процесів. У традиційних монолітних системах стан процесу зазвичай зберігається централізовано, що спрощує контроль і відновлення виконання. У мікросервісних середовищах стан може бути розподіленим між кількома компонентами, що ускладнює забезпечення цілісності та узгодженості процесів [62; 68]. Розподілене управління станами створює додаткові виклики для моніторингу, відлагодження та аналізу виконання бізнес-сервісів.

Подієво-орієнтовані архітектури відіграють ключову роль у сучасних платформах автоматизації, особливо в контексті асинхронного виконання бізнес-процесів. Подієві механізми дозволяють сервісам реагувати на зміни стану процесів і зовнішніх умов без жорсткої прив'язки до централізованого керування. Поєднання процесних рушіїв з подієво-орієнтованими архітектурами підвищує гнучкість і масштабованість систем, але одночасно знижує прозорість глобального виконання бізнес-процесів [62; 69].

Подієво-орієнтовані архітектури ускладнюють реалізацію класичних механізмів контролю виконання, таких як відстеження прогресу процесу або забезпечення транзакційної цілісності. Відсутність централізованого контролю потребує застосування додаткових архітектурних шаблонів, зокрема використання

журналів подій, кореляційних ідентифікаторів та механізмів компенсації [62; 68]. Це підвищує складність архітектури і вимагає більшої уваги до проєктування системи на ранніх етапах.

Модельно-орієнтовані архітектури відіграють важливу роль у платформах автоматизації, орієнтованих на спрощення проєктування бізнес-сервісів. У таких архітектурах процесні, сервісні та дані моделі використовуються як основа для автоматичної генерації або конфігурації компонентів виконання. Модельно-орієнтована організація розробки сприяє узгодженню бізнес-логіки з технічною реалізацією, водночас створюючи ризики надмірної залежності від інструментів моделювання та їх обмежень [58; 65].

Застосування штучного інтелекту та машинного навчання в архітектурах платформ автоматизації розглядається як перспективний напрям, спрямований на підвищення адаптивності та автоматизації управління виконанням бізнес-процесів. ІІІ-орієнтовані компоненти можуть використовуватися для оптимізації маршрутів виконання, прогнозування навантаження та підтримки прийняття рішень у режимі реального часу [65; 66; 75]. Проте інтеграція таких компонентів у архітектуру платформ ускладнює забезпечення передбачуваності та формальної верифікації поведінки системи.

Інтероперабельність між різними платформами автоматизації та зовнішніми системами є ще одним важливим аспектом архітектурних рішень. Відсутність єдиних стандартів і різноманіття рішень до реалізації процесних і сервісних моделей обмежують можливості перенесення рішень і повторного використання архітектурних компонентів [58; 59]. Це особливо актуально для організацій, які використовують кілька платформ або інтегрують бізнес-сервіси з різних джерел.

Архітектури, орієнтовані на хмарні та edge-середовища, розширюють можливості виконання бізнес-сервісів у розподілених контекстах, але одночасно підвищують вимоги до управління конфігурацією та узгодженості компонентів. Поєднання хмарних і edge-архітектур у межах єдиної платформи автоматизації ускладнює забезпечення стабільності виконання та контроль за нефункціональними характеристиками бізнес-сервісів [60; 64].

Архітектурні рішення сучасних платформ автоматизації характеризуються високим рівнем різноманіття та складності. З одного боку, використання мікросервісів, подієво-орієнтованих підходів і модельно-орієнтованих архітектур створює передумови для гнучкого та масштабованого виконання бізнес-процесів. З іншого боку, така складність ускладнює забезпечення керованості, прозорості та передбачуваності виконання бізнес-сервісів, що залишається однією з ключових наукових і практичних проблем [61–63; 69; 75].

Важливим напрямом аналізу архітектурних рішень платформ автоматизації є питання управління (governance), тобто сукупності механізмів управління, контролю та координації виконання бізнес-сервісів у межах складних розподілених систем. У наукових дослідженнях підкреслюється, що зі зростанням ролі мікросервісних і подієво-орієнтованих архітектур традиційні централізовані підходи до управління втрачають ефективність і потребують переосмислення [61; 62; 72]. Управління у цьому контексті охоплює не лише технічні аспекти, а й організаційні правила, політики безпеки та механізми контролю відповідності архітектурних рішень визначеним вимогам.

Архітектурні рішення у платформах автоматизації тісно пов'язані з питаннями керування життєвим циклом бізнес-сервісів. Сервіси та процеси проходять етапи проєктування, впровадження, експлуатації та подальшого розвитку, при цьому архітектура платформи має забезпечувати узгодженість між цими етапами. Відсутність цілісної архітектурної моделі управління призводить до фрагментації керування та зниження керованості виконання бізнес-процесів у довгостроковій перспективі [70; 72; 73].

У мікросервісних і подієво-орієнтованих системах виконання процесів розподіляється між багатьма компонентами, що ускладнює відстеження глобального стану і контроль дотримання нефункціональних вимог. У наукових публікаціях наголошується, що відсутність централізованої точки контролю вимагає використання спеціалізованих архітектурних шаблонів для збору та кореляції інформації про виконання [62; 68].

Механізми спостережуваності (observability) набувають особливого значення в архітектурах платформ автоматизації. Спостережуваність охоплює збирання метрик, логів і трасувань, які дозволяють аналізувати поведінку системи в реальному часі. У наукових дослідженнях підкреслюється, що без достатнього рівня спостережуваності неможливо забезпечити ефективне управління виконанням бізнес-сервісів і своєчасно виявляти відхилення від очікуваних рівнів якості [62; 69; 72; 74]. Водночас інтеграція механізмів спостережуваності у процесно-орієнтовані платформи ускладнюється різноманітністю компонентів і високим рівнем абстракції.

Проблема керованості виконання бізнес-процесів тісно пов'язана з архітектурними рішеннями щодо управління станами. У розподілених архітектурах стан процесу може бути фрагментований між кількома сервісами або зберігатися у вигляді подій, що ускладнює забезпечення узгодженості та відновлення після збоїв. У наукових роботах зазначається, що відсутність чітких архітектурних принципів управління станами знижує надійність виконання бізнес-сервісів і ускладнює реалізацію компенсаторних механізмів [62; 68; 71].

Архітектурні рішення також визначають можливості реалізації політик безпеки та контролю доступу у платформах автоматизації. Бізнес-сервіси часто обробляють критично важливі дані і виконують операції, які потребують суворого дотримання політик безпеки. У мікросервісних і подієво-орієнтованих архітектурах реалізація єдиних політик безпеки є складним завданням, оскільки контроль розподіляється між багатьма компонентами [70; 72]. Це підвищує вимоги до архітектурної узгодженості та механізмів централізованого управління політиками.

Одним із ключових викликів сучасних архітектур є проблема технічної прив'язаності до конкретних платформ і інструментів. Використання специфічних механізмів оркестрації, подієвих брокерів або модельно-орієнтованих інструментів може обмежувати можливості міграції та еволюції бізнес-сервісів [58; 59]. Це особливо актуально для платформ автоматизації, які прагнуть поєднати простоту використання з високим рівнем функціональності.

Проблема керованості та стабільності виконання бізнес-сервісів у сучасних архітектурах також пов'язана з використанням адаптивних і самокерованих механізмів. У літературі описуються рішення, спрямовані на формування самоналаштовуваних архітектур, здатних динамічно реагувати на зміни навантаження та стану ресурсів [71; 72]. Водночас зазначається, що впровадження таких механізмів у процесно-орієнтовані платформи ускладнює формальну верифікацію поведінки системи та може впливати на передбачуваність виконання бізнес-процесів.

Більшість сучасних платформ автоматизації не забезпечують повною мірою цілісної підтримки управління на всіх рівнях виконання бізнес-сервісів. Їх архітектури зазвичай зосереджені на окремих аспектах, зокрема масштабованості або гнучкості, але не завжди враховують системні вимоги до керованості та стабільності. Унаслідок цього можуть виникати розбіжності між архітектурними рішеннями та бізнес-вимогами [61–63; 72].

Узагальнення результатів аналізу архітектурних рішень сучасних платформ автоматизації проєктування та виконання бізнес-сервісів показує, що архітектура є визначальним чинником, який формує можливості та обмеження таких платформ щодо керованості, масштабованості та стабільності виконання бізнес-процесів. Еволюція архітектур від монолітних до сервіс-орієнтованих, мікросервісних і подієво-орієнтованих моделей створила передумови для підвищення гнучкості та адаптивності бізнес-сервісів, проте водночас суттєво ускладнила забезпечення цілісного контролю виконання процесів [58–60].

Сучасні платформи автоматизації дедалі частіше використовують децентралізовані архітектурні підходи, які зменшують залежність від централізованих компонентів і підвищують стійкість до відмов. Водночас, така децентралізація ускладнює реалізацію глобальної процесної логіки та контроль дотримання нефункціональних вимог на рівні бізнес-сервісів. Відсутність єдиного центру управління призводить до фрагментації відповідальності за виконання процесів і знижує передбачуваність поведінки системи в цілому [61–63; 69].

Застосування процесних рушіїв у мікросервісних і подієво-орієнтованих архітектурах створює додаткові виклики щодо узгодження рівнів абстракції. Процесні моделі, орієнтовані на бізнес-логіку, не завжди коректно відображають архітектурні особливості розподілених систем, такі як асинхронність, часткові відмови та затримки у взаємодії компонентів. Унаслідок цього спостерігається невідповідність між формальною моделлю процесу та фактичною поведінкою бізнес-сервісу під час виконання [64; 67; 68].

Аналіз архітектурних рішень свідчить, що більшість сучасних платформ автоматизації не забезпечують повноцінної підтримки управління виконанням бізнес-сервісів на всіх етапах їх життєвого циклу. Наявні механізми керування зазвичай зосереджені на окремих аспектах, зокрема конфігурації сервісів або контролі доступу, і не формують цілісної архітектурної моделі, здатної поєднати процесну логіку, ресурсні обмеження та нефункціональні вимоги [70; 72]. Як наслідок цього можливості системного управління виконанням бізнес-процесів у довгостроковій перспективі залишаються обмеженими.

Особливої актуальності набуває проблема спостережуваності та прозорості виконання бізнес-сервісів у складних архітектурах. Подієво-орієнтовані та мікросервісні підходи ускладнюють відстеження глобального стану процесів і причин відхилень від очікуваних показників якості. Без інтегрованих архітектурних механізмів спостережуваності неможливо забезпечити ефективний контроль виконання та своєчасне реагування на порушення нефункціональних вимог [62; 69; 72].

Проблема технічної прив'язаності до конкретних платформ і інструментів також є суттєвим архітектурним обмеженням сучасних систем автоматизації. Використання специфічних моделей, механізмів оркестрації та подієвих інфраструктур ускладнює перенесення бізнес-сервісів між платформами та стримує еволюцію архітектури. Відсутність узгодженої взаємодії між компонентами та стандартизованих архітектурних рішень може створювати ризики для довгострокової стійкості бізнес-сервісів [58; 59].

Інтеграція інтелектуальних і адаптивних механізмів у архітектури платформ автоматизації розглядається як перспективний напрям підвищення ефективності виконання бізнес-процесів. Водночас наукові дослідження вказують на те, що такі механізми ускладнюють формальну верифікацію та знижують передбачуваність поведінки системи, особливо у критичних сценаріях виконання. Це створює додаткові вимоги до архітектурних рішень, спрямованих на поєднання адаптивності з керованістю [65; 66; 71; 75].

Сучасні архітектурні рішення платформ автоматизації проектування та виконання бізнес-сервісів поєднують вимоги до гнучкості, масштабованості та швидкої адаптації з потребою забезпечення стабільного, керованого й передбачуваного виконання бізнес-процесів. Узгодження цих вимог становить архітектурну задачу, що проявляється на рівні системної організації платформи та не розв'язується лише за рахунок локальних оптимізацій або розширення функціональних можливостей [58–63; 72; 77].

1.4. Дослідження виконання бізнес-процесів у платформах автоматизації проектування та виконання бізнес-сервісів

Етап виконання бізнес-процесів у сучасних платформах автоматизації проектування та виконання бізнес-сервісів відіграє ключову роль у життєвому циклі процесно-орієнтованих систем, оскільки саме на ньому формалізовані моделі набувають практичної реалізації у вигляді операційної поведінки інформаційних систем. На цьому етапі виявляються особливості та обмеження архітектурних, процесних і технологічних рішень, визначених під час проектування. Ефективність виконання бізнес-процесів визначається не лише коректністю процесної моделі, а й здатністю платформи забезпечувати узгоджене, передбачуване та кероване виконання процесів у реальних умовах експлуатації [78; 80].

Сучасні платформи автоматизації орієнтовані на виконання бізнес-процесів у динамічних середовищах, де ресурси, сервіси та зовнішні системи можуть змінювати свій стан у часі. У таких умовах виконання бізнес-процесів набуває

характеру безперервної взаємодії між процесною логікою та середовищем виконання. Виконання процесів, засновані на статичній інтерпретації моделей, не завжди здатні адекватно реагувати на зміни навантаження, доступності ресурсів або зовнішніх подій [78; 81].

Однією із основних характеристик виконання бізнес-процесів є управління потоками екземплярів процесів, які можуть виконуватися паралельно та конкурувати за спільні ресурси. Із зростанням кількості паралельних процесів різко зростає складність забезпечення стабільного виконання і дотримання заданих часових та сервісних обмежень [78; 82]. У таких сценаріях незначні відхилення в розподілі ресурсів можуть призводити до накопичення черг, деградації продуктивності та порушення очікуваних показників якості.

Виконання бізнес-процесів у малокодових та безкодових платформах має додаткові особливості, пов'язані з високим рівнем абстракції та орієнтацією на нефармових користувачів. Такі платформи значно спрощують проектування процесів, але часто приховують внутрішні механізми виконання, що ускладнює аналіз і контроль поведінки процесів у критичних сценаріях [83; 84]. Це обмежує можливості точного налаштування виконання та ускладнює забезпечення стабільності бізнес-сервісів у довгостроковій перспективі.

Важливим аспектом виконання бізнес-процесів є спосіб інтерпретації або трансляції процесних моделей у виконувану логіку. У сучасних платформах використовуються різні рішення, включаючи безпосередню інтерпретацію моделей, їх часткову компіляцію або гібридні механізми. Дослідження показують, що інтерпретаційні підходи забезпечують високу гнучкість і швидке внесення змін, але можуть поступатися за продуктивністю у сценаріях з високим навантаженням [78; 85]. Натомість компіляційні та гібридні підходи дозволяють підвищити ефективність виконання, але обмежують можливості оперативної адаптації процесів.

У контексті виконання бізнес-процесів дедалі більшої уваги набувають інтелектуальні та адаптивні механізми, спрямовані на оптимізацію поведінки процесів у режимі реального часу. Застосування методів машинного навчання та

аналізу даних дозволяє прогнозувати тривалість виконання процесів, виявляти потенційні вузькі місця та адаптувати стратегії розподілу ресурсів [79; 82]. Інтеграція таких механізмів у платформи автоматизації ускладнює забезпечення передбачуваності та формальної верифікації виконання бізнес-процесів.

Подієво-орієнтовані архітектури відіграють важливу роль у виконанні бізнес-процесів у сучасних платформах. Події використовуються як тригери для ініціації процесів, переходу між станами та взаємодії між компонентами. Подієва модель виконання підвищує гнучкість і реактивність процесів, але водночас ускладнює глобальний контроль і аналіз їх поведінки [81; 91]. Це створює додаткові виклики для забезпечення узгодженого виконання бізнес-сервісів у складних розподілених середовищах.

Виконання бізнес-процесів у міжорганізаційних і розподілених сценаріях характеризується підвищеним рівнем невизначеності та залежності від зовнішніх систем. У таких умовах процеси можуть включати сервіси з різними характеристиками доступності та продуктивності, що ускладнює дотримання заданих сервісних рівнів. Відсутність єдиних механізмів контролю виконання у міжорганізаційних процесах є одним з основних обмежень сучасних платформ автоматизації [88; 89].

Виконання бізнес-процесів у платформах автоматизації є складним і багатовимірним процесом, який залежить від поєднання процесної логіки, архітектурних рішень і механізмів управління ресурсами. Виявлені особливості та обмеження виконання створюють передумови для подальшого аналізу проблем масштабованості, стабільності та керованості бізнес-процесів [78; 82; 93; 94; 96].

Виконання бізнес-процесів у платформах автоматизації нерозривно пов'язане з управлінням ресурсами та дотриманням угод про рівень надання послуг, які визначають допустимі межі продуктивності, часу відгуку та надійності бізнес-сервісів. У сучасних цифрових середовищах екземпляри бізнес-процесів виконуються в умовах обмежених і динамічних ресурсів, що потребує постійного узгодження процесної логіки з можливостями середовища виконання. Відсутність

явних механізмів інтеграції УРНП-показників у виконання процесів є однією з ключових причин деградації сервісів у реальних умовах експлуатації [78; 93; 94].

Одним із викликів виконання бізнес-процесів є забезпечення стабільної продуктивності за умов змінного навантаження. У платформах автоматизації паралельне виконання великої кількості екземплярів процесів призводить до конкуренції за обчислювальні, мережеві та організаційні ресурси. Дослідження показують, що без адаптивних механізмів розподілу ресурсів система не здатна підтримувати стабільні часові характеристики виконання, що безпосередньо впливає на дотримання заданих сервісних рівнів [78; 82; 96].

Проблема черг і затримок є особливо актуальною для виконання бізнес-процесів, які містять синхронні точки взаємодії або залежності між завданнями. У таких сценаріях затримка виконання одного завдання може призводити до каскадних затримок у всьому процесі. Сучасні платформи автоматизації часто не мають вбудованих механізмів прогнозування таких ефектів, що ускладнює проактивне управління виконанням бізнес-процесів [78; 82].

Виконання бізнес-процесів у малокодових та безкодових платформах характеризується додатковими обмеженнями, пов'язаними з універсальністю та стандартизацією середовища виконання. Такі платформи орієнтовані на широкий спектр сценаріїв використання і не завжди дозволяють оптимізувати виконання процесів під конкретні вимоги до продуктивності. У дослідженнях підкреслюється, що відсутність тонкого налаштування виконання в ПНОП може призводити до зниження ефективності та нестабільності бізнес-сервісів у сценаріях з високим навантаженням [86; 87; 96].

Особливе місце у виконанні бізнес-процесів займає питання моніторингу та аналізу виконання. Для забезпечення дотримання УРНП необхідно мати актуальну інформацію про стан процесів, використання ресурсів і поточні показники якості сервісів. У наукових публікаціях зазначається, що більшість сучасних платформ реалізують моніторинг як окрему функцію, не інтегровану безпосередньо з механізмами управління виконанням, що обмежує можливості оперативного реагування на відхилення [93; 95].

Використання методів прогнозного моніторингу розглядається як перспективний підхід до підвищення керованості виконання бізнес-процесів. Аналіз історичних даних про виконання процесів дозволяє прогнозувати тривалість виконання, імовірність порушення УРНП та потенційні вузькі місця. Такі методи можуть підвищувати стабільність бізнес-сервісів, але їх впровадження у сучасних платформах ускладнюється обмеженим доступом до внутрішніх механізмів виконання і високим рівнем абстракції [82; 95].

У розподілених і міжорганізаційних процесах виконання бізнес-процесів ускладнюється необхідністю узгодження різних УРНП і характеристик сервісів, що належать різним сторонам. У таких сценаріях порушення виконання одного сервісу може призводити до деградації всього бізнес-процесу. Сучасні платформи автоматизації не забезпечують достатніх механізмів координації та контролю виконання у міжорганізаційних середовищах [88; 89; 94].

Виконання бізнес-процесів у подієво-орієнтованих архітектурах створює додаткові виклики для забезпечення дотримання УРНП. Події можуть надходити нерівномірно, що призводить до пікових навантажень і складності прогнозування використання ресурсів. Без спеціалізованих механізмів управління подієвими потоками платформи автоматизації не здатні забезпечити стабільне виконання бізнес-процесів у таких умовах [81; 91].

Виконання бізнес-процесів у платформах автоматизації проектування та виконання бізнес-сервісів суттєво обмежується відсутністю інтегрованих механізмів управління ресурсами, УРНП та моніторингу. Наявні підходи орієнтовані переважно на реактивне усунення наслідків перевантаження та порушень сервісних рівнів, що не дозволяє гарантувати стабільність і передбачуваність виконання бізнес-процесів у довгостроковій перспективі [78; 93; 94; 96].

Виконання бізнес-процесів у подієво-орієнтованих і мікросервісних середовищах характеризується підвищеною складністю керування та контролю, що зумовлено децентралізованою природою таких архітектур. У сучасних платформах автоматизації бізнес-сервіси дедалі частіше реалізуються як сукупність автономних

компонентів, взаємодія між якими відбувається асинхронно через події або повідомлення. Така модель виконання сприяє масштабованості та стійкості до відмов, але водночас істотно ускладнює глобальне управління виконанням бізнес-процесів [81; 90; 91].

Однією із проблем виконання бізнес-процесів у мікросервісних середовищах є відсутність єдиного централізованого уявлення про стан процесу. У традиційних процесних рушіях стан виконання зберігається централізовано, що дозволяє відстежувати прогрес і забезпечувати контроль переходів між станами. У мікросервісних і подієво-орієнтованих архітектурах стан може бути розподіленим між кількома сервісами або представленим у вигляді послідовності подій, що ускладнює відновлення глобального контексту виконання [90; 91].

Виконання бізнес-процесів у таких умовах часто ґрунтується на принципах зрештою узгодженість (eventual consistency), що означає відмову від жорсткої узгодженості на користь підвищеної доступності та масштабованості. Використання слабкої узгодженості дозволяє системам ефективно працювати в розподілених середовищах, але водночас ускладнює забезпечення детермінованої поведінки бізнес-процесів і контроль виконання нефункціональних вимог [90; 92].

Проблема компенсації та відновлення виконання бізнес-процесів у разі збоїв. У подієво-орієнтованих і мікросервісних архітектурах традиційні транзакційні механізми часто є непридатними через розподілену природу системи. У наукових публікаціях підкреслюється, що реалізація компенсаторних механізмів на рівні бізнес-процесів є складною і потребує спеціальних архітектурних рішень, які не завжди підтримуються сучасними платформами автоматизації [90; 92].

Керованість виконання бізнес-процесів у розподілених середовищах також обмежується складністю кореляції подій і дій, що належать до одного екземпляра процесу. У наукових дослідженнях зазначається, що відсутність стандартизованих механізмів кореляції ускладнює аналіз виконання процесів і виявлення причин порушення УРНП або деградації сервісів [81; 91]. Це обмежує можливості ефективного моніторингу та управління виконанням у реальному часі.

Виконання бізнес-процесів у міжорганізаційних сценаріях додатково ускладнюється необхідністю взаємодії між автономними платформами та організаціями. У таких сценаріях кожна сторона може використовувати власні архітектурні рішення, політики безпеки та механізми управління ресурсами. Відсутність узгоджених механізмів контролю виконання між організаціями є однією з основних перешкод для гарантованого виконання міжорганізаційних бізнес-процесів [88; 89].

Проблема прозорості виконання бізнес-процесів набуває особливої актуальності у складних архітектурах, де значна частина логіки виконання прихована за сервісними інтерфейсами та подієвими потоками. У наукових дослідженнях зазначається, що обмежена прозорість ускладнює як оперативне управління виконанням, так і подальший аналіз ефективності бізнес-процесів [81; 95]. Це створює додаткові бар'єри для вдосконалення процесів і підвищення якості бізнес-сервісів.

Інтеграція інтелектуальних механізмів у виконання бізнес-процесів розглядається як один із шляхів подолання частини зазначених обмежень. Методи машинного навчання можуть використовуватися для виявлення аномалій, прогнозування збоїв і підтримки прийняття рішень під час виконання процесів. Проте, використання таких методів у розподілених і подієво-орієнтованих середовищах ускладнює забезпечення інтерпретованості та керованості виконання бізнес-процесів [79; 80].

Аналіз виконання бізнес-процесів у подієвих і мікросервісних середовищах свідчить, що сучасні платформи автоматизації мають обмеження щодо керованості, прозорості та передбачуваності виконання. Такі обмеження зумовлюються як особливостями архітектурних рішень, так і недостатньою формалізацією механізмів поєднання процесної логіки з управлінням ресурсами та сервісними рівнями [78; 81; 90–92].

Незважаючи на високий рівень формалізації процесних моделей і розвиток архітектурних підходів, реальна поведінка бізнес-процесів під час виконання значною мірою визначається характеристиками середовища виконання,

доступністю ресурсів і механізмами управління взаємодією між компонентами [78; 80].

Аналіз показує, що більшість сучасних платформ автоматизації орієнтовані на забезпечення коректності виконання процесної логіки, тоді як питання стабільності, передбачуваності та керованості виконання бізнес-процесів розглядаються вторинно. Відсутність явної інтеграції між процесними моделями, управлінням ресурсами та УРНП-призводить до того, що дотримання сервісних рівнів часто залежить від зовнішніх механізмів або ручного втручання [78; 93; 94].

Виконання бізнес-процесів у малокодових та безкодових платформах додатково ускладнюється високим рівнем абстракції, який обмежує прозорість і контроль внутрішніх механізмів виконання. Такі платформи спрощують проєктування та впровадження процесів, вони не завжди надають достатні засоби для аналізу поведінки процесів у критичних сценаріях або для точного управління ресурсами під час виконання [83; 84; 96].

Подієво-орієнтовані та мікросервісні архітектури, які широко застосовуються у сучасних платформах автоматизації, суттєво підвищують гнучкість і масштабованість виконання бізнес-процесів. Водночас вони ускладнюють забезпечення глобального контролю виконання, прозорості станів процесів і дотримання нефункціональних вимог. Відсутність централізованих механізмів управління у таких архітектурах обмежує можливості гарантованого виконання бізнес-процесів у межах заданих УРНП [81; 90–92].

Моніторинг і аналіз виконання часто здійснюються після факту виникнення порушень, що не дозволяє запобігти деградації сервісів у реальному часі. Методи прогнозного моніторингу та інтелектуального аналізу даних розглядаються як перспективні, їх інтеграція у платформи автоматизації залишається фрагментарною і недостатньо формалізованою [82; 95].

Виконання міжорганізаційних бізнес-процесів виявляє додаткові обмеження, пов'язані з відсутністю узгоджених механізмів контролю та координації між різними сторонами. У таких сценаріях порушення виконання одного компонента може призводити до деградації всього бізнес-сервісу, а наявні платформи

автоматизації не забезпечують достатніх інструментів для управління такими ситуаціями [88; 89; 94].

Узагальнення матеріалів підрозділу дозволяє відзначити наявність окремих розбіжностей між зростаючими вимогами до гнучкості, масштабованості та адаптивності бізнес-процесів і наявними можливостями сучасних платформ щодо забезпечення керованого та передбачуваного виконання бізнес-сервісів. Такі розбіжності проявляються насамперед на етапі виконання, де поєднуються наслідки архітектурних рішень, організації управління ресурсами та використання механізмів моніторингу [78; 81; 93; 97].

1.5. Постановка завдання дослідження та часткових завдань дисертації

Розглянутий у підрозділах 1.1–1.4 аналіз сучасних підходів до проєктування, виконання та управління бізнес-процесами у платформах автоматизації проєктування та виконання бізнес-сервісів демонструє, що сучасні платформи дозволяють швидко створювати бізнес-процеси, проте під час їхнього виконання в реальних умовах виявляються системні проблеми:

- відсутність замкненого циклу керування процесами (проєктування - виконання - моніторинг - проєктування);
- неможливість оперативного ЦПРС-контролю при динамічному навантаженні;
- обмеженість інструментів для експериментального дослідження та тестування політик диспетчеризації завдань.

Наукова проблема

Метою дисертаційної роботи є підвищення ефективності виконання кроків бізнес-процесів у платформах автоматизації проєктування та виконання бізнес-сервісів в умовах змінного навантаження.

Основне наукове завдання: Основним науковим завданням дослідження є обґрунтування, розроблення та експериментальна перевірка методів моніторингу та ресурсно-орієнтованого планування виконання кроків бізнес-процесів у

платформах автоматизації проєктування та виконання бізнес-сервісів, що забезпечують підвищення стабільності роботи та дотримання вимог до якості надання послуг в умовах змінного навантаження.

Об'єкт дослідження - процес виконання кроків бізнес-процесів у платформах автоматизації проєктування та виконання бізнес-сервісів в умовах змінного навантаження.

Предмет дослідження - моделі, методи та програмні засоби підвищення ефективності виконання кроків бізнес-процесів, що забезпечують моніторинг використання ресурсів, стабільність роботи бізнес-сервісів у платформах автоматизації проєктування та виконання бізнес-сервісів.

Гіпотеза дослідження: передбачається, що поєднання ресурсно-орієнтованих методів планування із моніторингом виконання кроків бізнес-процесів у платформах автоматизації проєктування та виконання бізнес-сервісів дозволяє забезпечити стійкіший режим за динамічного та нерівномірного навантаження, що проявляється у зменшенні частки порушень часових вимог терміну старту, і відповідно, у підвищенні виконання вимог до якості надання послуг.

Часткові наукові завдання

Для досягнення мети дисертаційного дослідження необхідно розв'язати такі часткові наукові завдання.

По-перше, провести порівняльний аналіз методів і засобів організації сучасних платформ автоматизації проєктування та виконання бізнес-сервісів для визначення особливостей та проблем їх реалізації.

По-друге, механізми моніторингу виконання кроків бізнес-процесів у платформі автоматизації проєктування та виконання бізнес-сервісів, визначити вимоги до телеметрії, інструментованого контролю ресурсів та виявлення порушень стабільності.

По-третє, розробити метод реактивного мульти-токенового планування задач за дедлайном старту, який дозволяє знизити ризик перевантаження середовища та зменшити частку порушень дедлайнів старту в умовах змінного та пікового навантаження.

По-четверте, розробити метод планування задач за правилом "перший прийшов - перший оброблений", який дозволяє зменшити ризик аварійних збоїв середовища виконання, зокрема, спричинених перевищенням обмеження оперативної пам'яті.

По-п'яте, розробити модель моніторингу та виконання кроків бізнес-процесів у ресурсно-обмеженому середовищі.

По-шосте, розробити програмне забезпечення системи управління та перевірки виконання кроків бізнес-процесів у платформі автоматизації проєктування та виконання бізнес-сервісів

Очікувані результати дослідження

У результаті дослідження очікується отримання узагальненої моделі виконання кроків бізнес-процесів у платформах автоматизації проєктування та виконання бізнес-сервісів, яка описує взаємозв'язок між характеристиками задач, ресурсними обмеженнями та параметрами навантаження. Модель має забезпечити можливість формального аналізу стабільності системи та умов допустимості виконання задач у динамічному середовищі.

Передбачається розроблення методу ресурсно-орієнтованого планування, який враховує дедлайни, профілі споживання ресурсів і поточний стан системи, забезпечуючи зменшення частки порушених дедлайнів та підвищення узгодженості виконання процесів. Очікується також формування критеріїв оцінювання ефективності планування та їх застосування для порівняння різних політик.

Також очікується формулювання теоретично обґрунтованих підходів до аналізу динаміки потоків задач, визначення умов стійкості та допустимості виконання у багаторесурсному середовищі й розроблення методичних принципів побудови систем моніторингу, здатних відображати взаємозв'язок бізнес-подій з технічними параметрами функціонування. Отримані результати мають забезпечити підґрунтя для подальшої розробки інструментальних засобів та процедур оцінювання ефективності планування й контролю у платформах автоматизації бізнес-процесів.

Наукова новизна

У межах дослідження отримано такі наукові результати, що становлять новизну.

Вперше розроблено метод реактивного мульти-токенового планування задач за дедлайном старту, який відрізняється від існуючих поєднанням етапів перевірки ресурсів для запуску задачі за мультиресурсною токенною моделлю з резервуванням "видано/повернуто" та вибором допустимої задачі з найменшим дедлайном старту, зазначений метод дозволяє знизити ризик перевантаження середовища та зменшити частку порушень дедлайнів старту в умовах змінного та пікового навантаження.

Удосконалено метод планування задач за правилом "перший прийшов - перший оброблений", який відрізняється введенням механізму ресурсного допуску задач перед запуском і правилом відкладеного старту за нестачі ресурсів та вибором допустимої задачі зі збереженням черговості обробки із очікуванням вивільнення ресурсів, що дозволяє зменшити ризик аварійних збоїв середовища виконання, зокрема спричинених перевищенням обмежень оперативної пам'яті.

Удосконалено модель моніторингу та виконання кроків бізнес-процесів у ресурсно-обмеженому середовищі, яка відрізняється від існуючих тим, що подає крок процесу як задачу з багатовимірними ресурсними потребами та часом постановки в чергу і дедлайном старту, який відокремлюється правилом пріоритету від механізму багаторесурсної перевірки здійсненності запуску, що задає метод вибору задачі із існуючої черги як фіксований параметр моделі та дозволяє порівняння альтернативних пріоритетів за показником частки порушень дедлайнів старту.

Набула подальшого розвитку інформаційна технологія автоматизації проектування та виконання бізнес-процесів у частині багаторівневої інтеграції засобів моделювання, виконання, моніторингу, аналітики та керування, що відрізняється реалізацією у замкненому контуру "виконання - телеметрія - аналітика - керування - вплив на виконання", розробленого методу реактивного мульти-токенового планування задач за дедлайном старту та удосконалено методу

планування задач за правилом "перший прийшов - перший оброблений", що дозволяє підвищити обґрунтованість підтримання стабільності виконання процесів за змінного навантаження.

Практичне значення результатів полягає у створенні інформаційної системи для моніторингу й керованого виконання бізнес-процесів за змінного навантаження та обмежених ресурсів. Результати тестування підтверджують її придатність для оцінювання ресурсів, цільового показника рівня сервісу, порівняння методів планування й удосконалення платформ автоматизації проєктування та виконання бізнес-сервісів.

Висновки до розділу 1

Здійснено узагальнений аналіз методів, засобів і технологічних підходів, що застосовуються для автоматизації проєктування та виконання бізнес-сервісів. Показано, що сучасні платформи цього класу є багаторівневими інформаційними системами, у яких необхідно забезпечувати узгодженість між моделями бізнес-процесів, механізмами їх виконання та інструментами контролю параметрів функціонування. Встановлено, що зростання складності цифрових сервісів, збільшення обсягу операцій та поява ресурсомістких задач формують високі вимоги до стабільності, керованості та передбачуваності роботи таких платформ.

Аналіз наукових джерел засвідчив, що значна частина існуючих рішень орієнтується на оптимізацію окремих компонентів системи, таких як моделювання, управління потоками задач або моніторинг ресурсів, проте не забезпечує їхньої повної інтеграції у спільний контур керування. У межах сучасних підходів недостатньо враховуються особливості динамічного навантаження та багатовимірні обмеження на використання ресурсів, що часто призводить до деградації продуктивності, зростання затримок, накопичення черг та порушення встановлених часових вимог.

Встановлено, що управління ресурсами та потоками задач у більшості платформ здійснюється за спрощеними сценаріями, які не дозволяють у повному

обсязі врахувати реальну доступність оперативної пам'яті, навантаження на процесор та інтенсивність операцій введення-виведення. Це ускладнює забезпечення відповідності параметрам якості надання послуг та створює ризики нестабільного функціонування у разі змінного або пікового навантаження. Проведений аналіз підкреслив потребу у застосуванні формальних моделей, здатних описувати взаємодію між ресурсними параметрами та поведінкою виконання кроків бізнес-процесів.

Окрему увагу приділено дослідженню систем моніторингу. Показано, що телеметрія, логування та трасування подій є необхідними складовими для контролю стану системи, однак їхня ефективність значною мірою залежить від можливості кореляції бізнес-подій із технічними показниками. Виявлено, що більшість інструментів моніторингу фокусуються на технічних метриках, тоді як відсутність узгодженості з моделями бізнес-процесів знижує інформативність аналізу та обмежує можливості своєчасного виявлення відхилень.

Проведений огляд платформ з низьким рівнем програмування показав, що попри успіхи у зменшенні обсягу ручної розробки та підвищенні швидкості впровадження рішень, у цих середовищах залишається невирішеною проблема адаптивного реагування на зміни ресурсоемності задач та коливання навантаження. Це підкреслює необхідність розроблення методів, здатних враховувати реальний стан системи під час прийняття рішень щодо планування виконання бізнес-процесів.

РОЗДІЛ 2 ДОСЛІДЖЕННЯ МЕХАНІЗМІВ МОНІТОРИНГУ ВИКОНАННЯМ БІЗНЕС-ПРОЦЕСІВ У ПЛАТФОРМАХ АВТОМАТИЗАЦІЇ ПРОЄКТУВАННЯ ТА ВИКОНАННЯ БІЗНЕС-СЕРВІСІВ

2.1. Аналіз засобів підтримки безпечного та стабільного виконання бізнес-процесів у платформах автоматизації проєктування та виконання бізнес-сервісів

Під безпечним і стабільним виконанням бізнес-сервісів у платформах автоматизації проєктування та виконання бізнес-процесів розуміється дотримання визначених ЦПРС за умов обмежених обчислювальних ресурсів і змінного характеру навантаження. У межах подальшого розгляду УРНП пов'язуються не лише з показниками латентності та пропускну здатності, а й із передбачуваністю запуску кроків, зокрема з дотриманням дедлайнів старту виконання, а також із недопущенням неконтрольованого зростання черги та перевищення ресурсних меж. Для середовищ, чутливих до ресурсів, важливо розрізняти жорсткі обмеження (гранично допустимі значення використання ресурсів і місткості черги) та м'які цільові рівні (бажаний профіль споживання ресурсу, наприклад підтримання використання пам'яті нижче заданої частки від максимального значення), які використовують як сигнали для превентивного реагування. У середовищах із мінімальним або відсутнім програмуванням, що підтримують нотацію моделювання бізнес-процесів та нотацію моделей рішень, забезпечення стабільності базується на поєднанні таких механізмів, як керовані пули обробників, черги завдань, політики повторного виконання та ідемпотентності, саги для узгодженості в розподілених операціях, а також наскрізна телеметрія для трасування кроків бізнес-процесу й кореляції технічних подій [88; 101-103; 105; 107; 114–116].

Засоби підтримки стабільності в розподілених і ресурсно обмежених системах доцільно класифікувати на п'ять категорій, кожна з яких спрямована на запобігання

конкретним типам ризиків і забезпечення передбачуваності виконання бізнес-процесів [90; 110; 111; 118].

Перша категорія - керування навантаженням, що охоплює методи регулювання інтенсивності оброблення запитів: зворотний тиск, обмежувачі приймання, черги з пріоритетами та механізми буферизації подій. Ці підходи потрібні для уникнення перевантаження й підтримання керованого стану черги, зокрема для запобігання переповненню та зростанню затримки запуску кроків або завдань за умов високої мінливості потоку подій [91; 104; 105].

Друга категорія - керування ресурсами, що передбачає застосування бюджетів на споживання процесорного часу, оперативної пам'яті та операцій введення-виведення. Такі механізми підвищують прогнозованість виконання та запобігають неконтрольованій деградації продуктивності; у практичних сценаріях важливо утримувати використання ресурсів у допустимих межах і, за потреби, задавати м'який цільовий рівень, який слугує сигналом для раннього реагування ще до досягнення жорстких обмежень. У контексті хмарних платформ ці підходи пов'язують із політиками автоматичного масштабування та плануванням навантаження [82; 100; 105; 106; 112; 115].

Третя категорія - засоби узгодженості та стійкості, що охоплюють саги, транзакції, шаблони outbox/inbox, механізми доставки повідомлень із гарантіями доставлення та методи координації станів у розподілених системах; їх призначенням є забезпечення коректного виконання бізнес-операцій за умов можливих часткових збоїв [103; 116; 120].

Четверта категорія - моніторинг, журналювання та трасування - включає стандарт OpenTelemetry, засоби APM, моніторинг на основі eBPF, профілювання та аналіз затримок; ці інструменти забезпечують спостережуваність стану процесів, дають змогу виявляти аномалії, діагностувати затримки, аналізувати поведінку компонентів і формувати підґрунтя для оптимізації, зокрема через зіставлення бізнес-подій із технічними вимірюваннями [107–109; 113; 117; 119].

П'ята категорія - відновлення та самовідновлення, яка охоплює механізми повторних спроб із випадковою затримкою, запобіжники, реплікацію станів, механізми аварійного перемикавання та автоматичне переналаштування маршрутів виконання; ці підходи спрямовані на запобігання каскадним відмовам і підтримання безперервності роботи системи [103; 106; 110; 111; 118].

Сукупність зазначених категорій формує комплексний набір механізмів, що забезпечують стабільність виконання бізнес-процесів за умов високого навантаження, неоднорідних ресурсів та мінливого середовища [104; 105; 110; 111; 118]. Кожна категорія контролює окремий тип ризиків: перевантаження, дефіцит пам'яті, часткові відмови, затримки, деградацію та каскадні збої. Однією з ключових вимог для ПНОП-платформ є конфігураційна доступність стабілізаційних механізмів, що дозволяє активувати політики обмеження ресурсів, механізми деградації або УРНП-контролю без модифікації програмного коду. Важливим також є відображення бізнес-подій (екземпляри BPMN, правила DMN) на технічні метрики, такі як використання ЦП, оперативної пам'яті, операції введення-виведення, довжина та вік черг [81; 82; 88; 107].

Розмежування бізнес-орієнтованих і ресурсно-орієнтованих метрик забезпечує прозорість управління виконанням процесів і визначає пріоритети адаптаційних сценаріїв; при цьому метрики латентності (p50/p95/p99) доцільно трактувати як індикатори деградацій і вузьких місць, тоді як для задач із дедлайнами запуску окрему роль відіграє контроль фактів/частки порушення дедлайну старту та подій відхилення задач через переповнення черги.

Табл. 2.1.

Порівняльна характеристика методів та засобів підтримки стабільності

Метод/засіб	Суть	Переваги	Ризики/ обмеження	Рекомендації

Механізм зворотного тиску	Регулювання прийому за станом черг/ресурсів	Захист ядра виконання	Зростання затримки (зокрема затримки запуску)	Увімкнути завжди; прив'язати до довжини/віку черги [104; 105]
Механізми обмеження ресурсів	Бюджети ЦП, ОП, В/В на крок/процес	Уникнення нестачі оперативної пам'яті, прогнозованість	Недовантаження при завищених запасах	Калібрувати за історією, адаптивно [100; 105; 115]
Сага/транзакції	Компенсації у розподілених діях	Стійкість до часткових збоїв	Вартість компенсацій	Для міжсервісних процесів [103; 116]
OTel/APM	Наскрізне трасування і метрики	Видимість вузьких місць	Накладні витрати збору	Зняття показників у часі, агрегування [107-109; 113; 117; 119]
DMN-політики	Декларативні УРНП-правила	Прозорість рішень	Потреба в якісних даних	Оцифрувати УРНП у DMN [88; 114; 115]

Наведені методи/засоби (Табл. 2.1) відображено їхні переваги, обмеження та рекомендації щодо застосування. Зокрема, механізми зворотного тиску є базовим інструментом захисту від перевантаження й стабілізації стану черги; обмежувачі ресурсів забезпечують прогнозованість виконання та контроль використання ЦП/ОП/В/В; саги підвищують стійкість до часткових збоїв; засоби трасування

сприяють забезпеченню спостережуваності; DMN-політики дають змогу формально описувати правила центрів прийняття рішень і сценарії деградації [88; 100; 103-105; 107; 114-116].

До ключових параметрів стабільності виконання бізнес-процесів належать латентність (p50/p95/p99), пропускна здатність, частка відмов і повторів, використання обчислювальних ресурсів, довжина черг та дотримання дедлайнів УРНП; у задачах, де дедлайни визначаються для моменту старту кроку/задачі, критичним є контроль затримки запуску та частки порушень дедлайну старту поряд із контролем черги та ресурсів. Забезпечення стабільності потребує одночасного контролю інтенсивності вхідних потоків, обсягів доступних ресурсів і стану пулів обробників [98; 99; 104; 105; 114].

Отже, стабільне та безпечне виконання бізнес-процесів у платформах із мінімальним або відсутнім програмуванням потребує інтеграції механізмів керування навантаженням, бюджетування ресурсів і наскрізної телеметрії, із обов'язковим відображенням бізнес-подій у технічних метриках та застосуванням правил нотації моделей рішень, що описують сценарії деградації й відновлення. Водночас необхідно розрізняти жорсткі обмеження та м'які цільові рівні використання ресурсів і підтримувати керованість стану черги для зменшення ризиків порушення дедлайнів запуску.

2.2. Дослідження системи моніторингу виконання бізнес-процесів в платформах автоматизації проєктування та виконання бізнес-сервісів

До систем моніторингу висуваються такі вимоги: спостереження перебігу BPMN-кроків у реальному часі; збирання технічних метрик, включаючи показники використання процесора, оперативної пам'яті, операцій введення-виведення та параметри черг; ведення журналів і трасування з кореляцією “екземпляр процесу - технічні події”, що забезпечує можливість реконструкції ланцюга виконання; виконання ЦПРС-перевірок і формування сповіщень у разі відхилень; підтримка

аналітичних сценаріїв, включно з аналізом першопричин, прогнозуванням деградацій та виявленням аномалій [81; 88; 94; 95; 107; 113; 117].

У контексті задач керування виконанням бізнес-процесів важливо, щоб ЦПРС-перевірки могли бути узгоджені з постановкою задачі планування: дедлайни трактуються як дедлайни старту кроків/задач, а якість сервісу оцінюється, зокрема, через частку порушень дедлайну старту (включно з випадками відхилення задачі через переповнення черги), поряд із контролем ресурсних і чергових обмежень.

Порівнюються два архітектурні підходи до телеметрії: агентський і інструментований. Агентська телеметрія вирізняється незначним впливом на роботу системи та зручністю розгортання на рівні інфраструктури, проте має обмежені можливості кореляції з бізнес-подіями, оскільки фокусується на технічних параметрах сервісів і вузлів. Інструментована телеметрія, що вбудовується у рушій процесів або окремі сервіси, надає точну відповідність між BPMN-кроками та технічними вимірюваннями, дозволяє відстежувати поведінку кожного екземпляра процесу, проте, вимагає підтримання дисципліни інструментування й оптимізації накладних витрат на збирання даних [107-109; 113].

Для задач, де необхідно оцінювати дотримання дедлайнів старту та причини відхилень/затримок запуску, інструментований підхід є критично важливим через потребу фіксації моментів надходження задач у чергу, моментів їх запуску, причин відхилення (зокрема переповнення черги) та ресурсного профілю задач.

Табл. 2.2

Порівняльна характеристика агентської та інструментованої телеметрії

Критерій	Агентська телеметрія	Інструментована телеметрія
Ступінь втручання	Низька	Середня/висока
Кореляція з BPMN	Обмежена	Висока
Вартість даних	Стабільна	Регульована семплюванням

Сигнали стабільності	Рівень вузла/пулу	Рівень кроку/задачі
Використання	Огляд здоров'я	Аналіз першопричин, ЦПРС-контроль [94; 113; 121]

Модель даних телеметрії охоплює події (start/stop/error), метрики (часові характеристики оброблення та запуску, використання процесора, пам'яті, операцій введення-виведення, довжину та вік черг) та контекстні ознаки: process_instance_id, task_id, trace_id/span_id, ЦПРС-профіль. Для узгодження з постановкою задач планування до контекстних ознак доцільно відносити також параметри, необхідні для оцінювання дедлайнів старту (зокрема момент надходження задачі в чергу та значення відносного дедлайну старту), ресурсний профіль задачі (потреба в ресурсах і очікуваний час виконання), а також службові атрибути рішення виконання (запущено/очікує/відхилено з фіксацією причини).

Доцільно застосовувати принцип агрегування: спочатку аналізують агреговані показники (p50, p95, максимальні значення, частки відхилень), а деталізацію виконують лише після спрацювання тригерів відхилень [107; 109; 113; 122]. Водночас показники затримок (зокрема перцентилі) розглядають як допоміжні індикатори деградації, тоді як основний показник якості для завдань зі стартовими дедлайнами визначають через факт або частку порушення дедлайну старту.

До типових ЦПРС-перевірок і політик належать:

- контроль характеристик запуску та обслуговування задач (зокрема дотримання дедлайнів старту або контроль частки порушень дедлайну старту на інтервалі спостереження);
- контроль довжини та віку черги (обмеження на рівні Q_{max}/A_{max} , а також умови відхилення задач при переповненні черги);
- контроль частки повторень $< R_{max}$; контроль використання пам'яті з розрізненням жорстких і м'яких вимог, а саме, максимальний рівень як жорстке обмеження (наприклад, $\leq M_{max}$) та бажаний профіль споживання як м'яка ціль (наприклад, рівень до 80% від M_{max}), перевищення якої інтерпретується як сигнал для превентивного реагування. Порушення ЦПРС

ініціює застосування політик деградації: контрольоване зменшення пропускної здатності, зміна розміру пулу потоків, перенаправлення чи адаптація маршрутів BPMN/DMN відповідно до наперед заданих сценаріїв [81; 88; 106; 110; 111; 118].

У завданнях, де споживання ресурсів має утримуватися в межах заданого максимуму з допустимим м'яким запасом (з огляду на ризик аварійного завершення у разі перевищення), контроль використання пам'яті та стану черги є необхідним елементом як моніторингу, так і контурів керування.

Для виявлення причин деградацій потрібні узгоджені подання, що поєднують бізнесові та технічні дані. До них належать часові ряди метрик з прив'язкою до екземплярів процесів, діаграми затримок і тривалостей із декомпозицією часу на складові, а також кореляційні залежності типу “довжина черги - пропускна здатність - виконання ЦПРС”, які, у випадку стартових дедлайнів, доповнюються залежностями “параметри черги/ресурсів - затримка старту - частка порушень дедлайну старту” [104; 105; 107; 113; 123].

Для подальшого використання в дискретних контурах керування важливо, щоб агрегування та нормалізація телеметрії забезпечували зіставність даних у тактах дискретного часу та підтримували формування стану системи на кожному кроці.

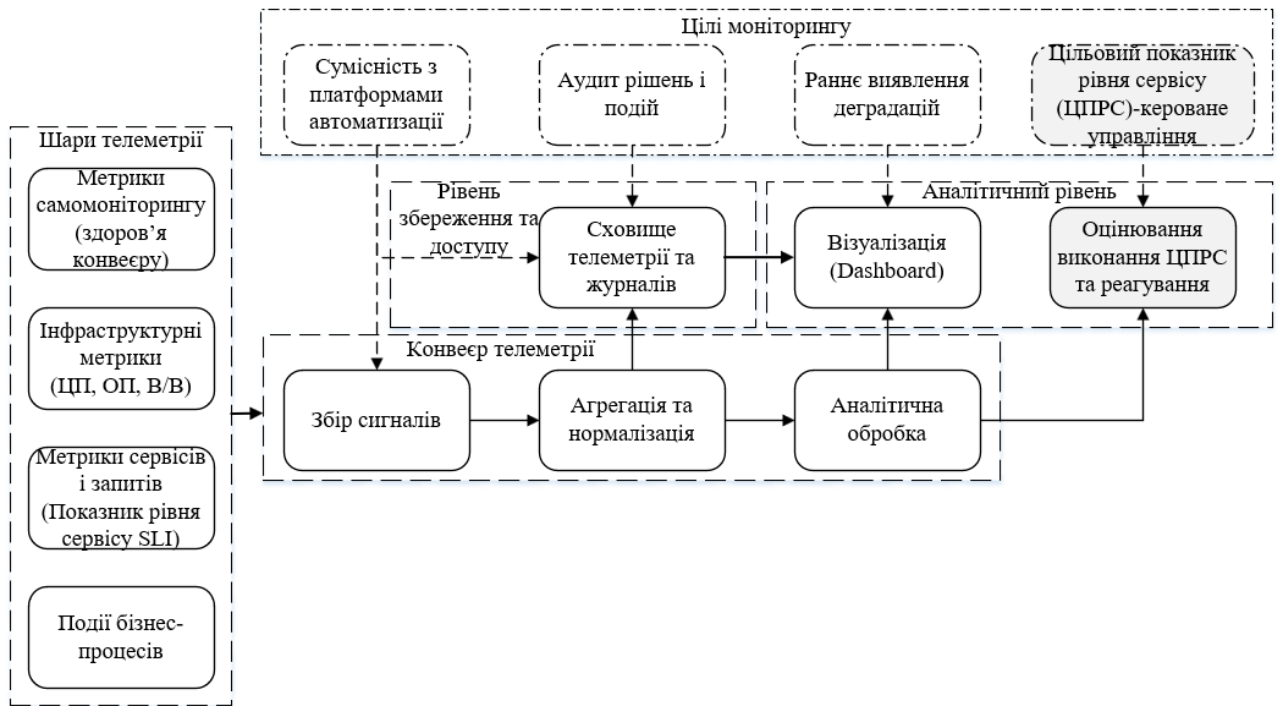


Рис. 2.1. Концептуальна модель підсистеми моніторингу та конвеєра телеметрії виконання бізнес-процесів у платформі проєктування та виконання бізнес-сервісів

Концептуальна модель підсистеми моніторингу та конвеєра телеметрії виконання бізнес-процесів (Рис. 2.1) представлена як конвеєр збирання й оброблення телеметрійних сигналів із подальшим збереженням і доступом до даних, візуалізацією результатів та оцінюванням виконання цільового показника рівня сервісу (ЦПРС). Джерела даних згруповано у блоці “Шари телеметрії”, який охоплює чотири категорії сигналів: метрики самомоніторингу (здоров'я конвеєру) - показники, що характеризують роботу самого конвеєра телеметрії; інфраструктурні метрики (ЦП, ОП, В/В) - показники завантаження процесора, використання оперативної пам'яті та інтенсивності введення-виведення; метрики сервісів і запитів (показники рівня сервісу, ПРС) - вимірювані індикатори якості обслуговування; події бізнес-процесів - подієві сигнали, що відображають виконання бізнес-процесів. У задачах зі стартовими дедлайнами до подієвих сигналів і контексту також належать відомості, необхідні для відтворення траєкторії виконання кроку: момент надходження в чергу, момент запуску, факт/причина відхилення та відповідність старту встановленому дедлайну.

Зібрані сигнали надходять у “Конвеєр телеметрії”, який містить три послідовні функціональні блоки. “Збір сигналів” забезпечує приймання телеметрії з різних шарів та її первинну підготовку до подальшого оброблення у конвеєрі. “Агрегація та нормалізація” виконує узгодження та перетворення первинних сигналів у стандартизований для оброблення вигляд; у межах цього блоку дані приводяться до узгодженої структури та формату (зокрема для порівняння показників між джерелами) і готуються як до збереження, так і до аналітичного використання. “Аналітична обробка” використовує нормалізовані дані для формування узагальнених результатів, потрібних для подальшої інтерпретації стану виконання та якості сервісу, а також для оцінювання досягнення цільових показників роботи системи. Це включає оцінювання дотримання дедлайнів старту й контроль ресурсних та чергових обмежень із розрізненням жорстких меж і м’яких цільових рівнів використання ресурсів.

Паралельно з основним потоком оброблення передбачено дві результуючі гілки. “Сховище телеметрії” отримує дані після блоку агрегації та нормалізації, забезпечуючи їх накопичення та використання для подальшого аналізу й перегляду. “Візуалізація (Dashboard)” отримує дані як із аналітичної обробки, так і зі сховища телеметрії, формуючи узгоджене відображення стану та показників, доступних для спостереження. Окремим виходом конвеєра є блок “Оцінювання виконання ЦПРС та реагування”, куди надходять результати аналітичної обробки. У цьому блоці результати вимірювань ПРС інтерпретуються відносно встановленого ЦПРС і формується результат оцінювання виконання, а також рішення щодо реагування (у межах того, що визначено політикою реагування), при цьому для задач із дедлайнами старту результат оцінювання має включати фіксацію порушень дедлайну старту та подій відхилення через переповнення черги як окремих класів відхилень сервісу.

У верхній частині діаграми виділено блок “Цілі моніторингу”, який задає орієнтири функціонування підсистеми. Пунктирні зв’язки від цілей до окремих елементів відображають, що ці цілі виступають вимогами/напрямами, з якими співвідносяться функції відповідних компонентів: “Сумісність з платформами

автоматизації” пов’язується із блоком збору сигналів та підтримкою накопичення/використання даних у сховищі телеметрії; “Аудит рішень і подій” пов’язується зі сховищем телеметрії, де зосереджуються дані для подальшого перегляду; “Раннє виявлення деградацій” пов’язується з візуалізацією (Dashboard) як засобом оперативного спостереження; “ЦПРС-кероване управління” пов’язується з блоком оцінювання виконання ЦПРС та реагування, який використовує цільові значення для інтерпретації вимірних індикаторів.

Узагальнено, діаграма фіксує потік “джерела телеметрії - збирання - агрегація/нормалізація - аналітика - оцінювання ЦПРС та візуалізація/збереження”, де цілі моніторингу задають вимоги до того, які дані збираються, як вони зберігаються, як відображаються і як використовується оцінювання виконання встановлених цільових показників рівня сервісу, включно з контролем ресурсних меж, стану черги та дотриманням дедлайнів старту задач у процесному русії.

2.3. Розробка моделі моніторингу виконання кроків бізнес-процесів в платформах автоматизації проєктування та виконання бізнес-сервісів на основі модуля контролю виконання бізнес-процесів

У межах дослідження розглядаються платформи з підтримкою BPMN/DMN та можливістю інструментованої телеметрії. Ресурсні обмеження визначаються квотами оперативної пам’яті, частками процесорного часу та лімітами операцій введення-виведення; навантаження є стабільним у часі з окремими піковими періодами. Основною метою є забезпечення керованої деградації та прогнозованої стабільності виконання за умови мінімізації порушень УРНП/ЦПРС [78; 81; 82; 88; 92].

Нехай система функціонує з дискретним часом $k \in N$ та кроком Δt . У системі визначено скінченний набір бізнес-сервісів S , кожен з яких розглядається як функціонально завершена одиниця цінності, що характеризується цільовими показниками якості виконання (УРНП/ЦПРС) та обмеженнями на використання

ресурсів. Кожен бізнес-сервіс **реалізується** одним або кількома бізнес-процесами, що належать до глобального набору процесів P . Для сервісу s множини процесів, задіяних у його реалізації, позначається як $P_s \subseteq P$. Кожен бізнес-процес $p \in P$ описує керовану логіку виконання та **декомпонується** на кроки, які в середовищі виконання інтерпретуються як задачі певних класів. Для процесу p визначається підмножина допустимих класів задач $C_p \subseteq Class$, де $Class$ - скінченна множина типів задач, класифікованих за домінуючим профілем споживання ресурсів (зокрема, ЦП-інтенсивні, ОП-інтенсивні, В/В-інтенсивні).

Кожен крок (екземпляр елемента BPMN під час виконання) відображається на задачу планувальника J_i , що описується профілем:

$$\begin{aligned} J_i &= (r_i, e_i, d_i), \\ r_i &= (c_i, m_i, io_i), \end{aligned} \quad (2.1)$$

де r_i - потреба в ресурсних токенах ЦП, ОП, В/В;

e_i - тривалість виконання;

d_i - дедлайн допустимої затримки старту.

Ресурсні квоти задаються вектором

$$R = (C_{max}, M_{max}, IO_{max}), \quad (2.2)$$

де C_{max} - обмеження щодо використання ЦП,

M_{max} - обмеження щодо використання ОП,

IO_{max} - обмеження щодо використання В/В [92; 100; 105; 110].

Потоки надходжень a_k - вектор інтенсивностей надходження робіт за класами на кроці.

Вектор стану системи:

$$x_k = (q_k, p_k, m_k, c_k, io_k, \hat{L}_k^{p95}), \quad (2.3)$$

де

q_k - довжина черги,

p_k - кількість активних обробників,

m_k - використана оперативна пам'ять,

c_k - завантаження ЦП,

io_k - інтенсивність В/В

$\hat{L}_k^{p95}$ - 95-й процентиль затримок виконання, який є індикатором стабільності й прогнозних властивостей системи [94; 104; 113; 116].

Вектор керування

$$u_k = (\gamma_k, h_k, \lambda_k^{io}, d_k), \quad (2.4)$$

де

$\gamma_k \in [0, 1]$ - масштабування прийому/виконання, це коефіцієнт, який визначає, яку частку від максимально можливого навантаження система реально приймає або виконує.

$\gamma_k = 0$ - повна зупинка (жодне завдання не приймається/не виконується);

$\gamma_k = 1$ - робота на повній пропускній здатності (усі завдання приймаються й виконуються);

$0 < \gamma_k < 1$ - система працює в режимі обмеженого прийому/виконання, тобто масштабує навантаження (наприклад, приймає лише 70% запитів при $\gamma_k = 0.7$).

h_k - розмір пулу обробників,

$\lambda_k^{io} \in [0, IO_{max}]$ - ресурсне обмеження: кожна задача може використовувати лише певний діапазон певного ресурсу, наприклад, В/В, який не перевищує глобального ліміту(квоти) IO_{max} ;

d_k - вектор дедлайнів класів C задач.

$$q_{k+1} = \max\{0, q_k + \gamma_k \alpha_k - \mu_k(h_k, \rho_k) \Delta t\}, \quad (2.5)$$

де α_k - сукупна інтенсивність надходження,

$\mu_k = \mu_0 h_k \psi(\rho_k)$ - продуктивність пулу обробників,

μ_0 - номінальна продуктивність одного обробника (кількість задач на секунду),

ρ_k - вектор завантажень ресурсів,

$\psi(\cdot) \in (0, 1]$ - коефіцієнт ефективності, що враховує вплив ресурсних обмежень [100; 104; 105; 123].

Баланс ресурсів:

$$m_{k+1} = m_k + \Delta m_k(h_k, d_k) - release_k, \quad (2.6)$$

$$c_{k+1} = \min(C_{max}, \max(0, c_k + \Delta c_k(h_k, d_k) - \xi_k^{cpu})),$$

$$io_{k+1} = \min(IO_{max}, \max(0, io_k + \Delta io_k(h_k, d_k, \lambda_k^{io}) - \xi_k^{io})),$$

де m_k - поточний обсяг оперативної пам'яті;

$\Delta m_k(h_k, d_k)$ - приріст пам'яті через задачі, що виконуються на кроці k (склад і кількість цих задач визначаються h_k та правилом відбору за дедлайном);

$release_k$ - обсяг пам'яті, що звільнився на кроці k (завершення задач, звільнення об'єктів, збір сміття тощо);

c_k - поточний рівень використання ЦП;

$\Delta c_k(h_k, d_k)$ - додаткове ЦП-навантаження від задач, які реально виконуються на кроці k ;

ξ_k^{cpu} - зменшення/компенсація ЦП-навантаження на кроці k (наприклад, спад фонові активності або частина навантаження, що зникає через завершення робіт);

io_k - поточне використання В/В-ресурсу (диск/мережа/БД тощо);

$\Delta io_k(h_k, d_k, \lambda_k^{io})$ - приріст В/В-навантаження від задач на кроці k , з урахуванням на кожну задачу обмеження λ_k^{io} ;

ξ_k^{io} - зменшення/компенсація В/В-навантаження на кроці k .

GC-пауза/латентність ідентифікуються емпірично [124; 125]:

$$g_{k+1} = \min(1, \max(0, \alpha_g \phi_{mem} \left(\frac{m_k}{M_{max}} \right) + \beta_g g_k + \varepsilon_k^{(g)})) \quad (2.7)$$

$$\hat{L}_{k+1}^{p95} = \alpha_L \phi_q(q_{k+1}) + \beta_L \phi_{cpu} \left(\frac{c_{k+1}}{c_{max}} \right) + \gamma_L g_{k+1} + \varepsilon_k^{(L)},$$

де $g_k \in [0, 1]$ - нормований стан/індикатор інтенсивності GC-пауз (або "ризик" GC-пауз) на кроці k ;

$\phi_{тет}$ - функція перетворення нормованої пам'яті у внесок у GC-стан;

α_g - вага впливу пам'яті на GC-паузи;

β_g - інерційність (наскільки g_{k+1} “успадковує” попередній стан g_k);

$\varepsilon_k^{(g)}$ - випадкова/неврахована складова, що відображає емпіричні фактори (тип алокацій, патерни роботи, фрагментація тощо).

Для затримки

$\hat{L}_{k+1}^{p95}$ - прогноз/оцінка 95-го перцентилі латентності на кроці $k+1$;

α_L - вага впливу черги на p95-латентність;

β_L - вага впливу ЦП-навантаження на p95-латентність;

γ_L - вага впливу GC-стану на p95-латентність (GC-паузи прямо додають затримки);

$\varepsilon_k^{(L)}$ - емпірична похибка (невраховані фактори: В/В, блокування, конкурентні ресурси, мережа тощо).

Ефективність системи в умовах ресурсних обмежень описується функцією $\psi(\rho_k)$, яка відображає зменшення пропускної здатності зі зростанням нормованого дефіциту ресурсів. Для цього використовується модель штрафу за перевантаження, у якій внесок кожного ресурсу (ОП, ЦП та В/В) враховується окремо через нормовані показники навантаження $\phi_{тет}, \phi_{сри}, \phi_{іо} \in [0, 1]$ та вагові коефіцієнти $\sigma_{тет}, \sigma_{сри}, \sigma_{іо} \geq 0$, тоді ефективність визначається як:

$$\psi(\rho_k) = \max(0, 1 - \sigma_{тет}\phi_{тет} - \sigma_{сри}\phi_{сри} - \sigma_{іо}\phi_{іо}), \quad (2.8)$$

де $\psi(\rho_k)$ - функція, яка показує, наскільки зменшується пропускна здатність елемента або системи, коли зростає нормований дефіцит ресурсу [100; 105; 123];

$\phi \in [0, 1]$ - нормування завантаження відносно квот і ЦПРС;

$\sigma \geq 0$ ваги впливу дефіцитів ресурсів.

Для узагальненої оцінки стану системи на кроці k формується вектор ознак:

$$x_k = (l_k, p_k, q_k, m_k, c_k, io_k), \quad (2.9)$$

який агрегує ключові показники якості обслуговування та поточного навантаження. Нормування компонент вектора ознак виконується відносно граничних значень (квот) і цільових рівнів продуктивності, після чого вводяться вагові коефіцієнти $\omega_r \geq 0$ для узгодженого врахування важливості різних складових за умови $\sum_r \omega_r = 1$. На цій основі визначається індикатор стабільності

$$I_{s,k} = 1 - \sum_{r \in (l,p,q,m,c,io)} \omega_r \phi_r(x_k), \quad (2.10)$$

який зменшується зі зростанням нормованих відхилень/навантажень і може застосовуватися для моніторингу та порівняння режимів роботи системи в динаміці.

Із порогоми I_s^{warn}, I_s^{crit} калібруються на історичних даних, із урахуванням невизначеностей, характерних для мікросервісів [92; 104; 105; 117]. Якщо $I_{s,k} \leq I_s^{warn}$, активуються політики деградації: регулювання швидкості прийому чи виконання задач; масштабування ресурсів (динамічне збільшення або зменшення обсягу виділених обчислювальних ресурсів); визначення пріоритетів виконання задач (на основі критичності, УРНП); обмеження операцій В/В (встановлення верхніх меж на кількість чи обсяг В/В для задач або процесів) [100; 105; 106; 110; 111; 126].

Табл. 2.3

Мета керування, сигнали та дії МКВ

Мета	Сигнали	Дії МКВ	Очікуваний результат
Уникнути нестачі	$\frac{m_k}{M_{max}}, g_k$	зменшити γ_k , відкласти нетермінові кроки,	Менше збоїв/перевищення використання ОП,

оперативної пам'яті		знижити h_k для memory-bound етапів	стабільність [76; 100; 105]
Знизити p95	q_k , вік у черзі, $\hat{L}_k^{p95}$	Адаптувати h_k , d_k (пріоритети, дедлайни УРНП-критичних класів)	Кращі p95/p99 [94; 100; 104; 105; 116]
Обмежити операції введення-виведення	IOPS/MBps, ϕ_{io}	Встановити λ_k^{io} тимчасово паузити В/В-важкі кроки	Уникнення голодування ЦП [100; 105; 123]
Узгодженість саг	частка повторів/компенсацій	контроль саг, виконати повторні спроби з випадковою затримкою	Менше компенсаційних транзакцій [61; 81; 88; 89]

ЦПРС/УРНП-обмеження та допустима область

ЦПРС-обмеження

$$\hat{L}_k^{p95} \leq \hat{L}_*^{p95}, g_k \leq g_*, err_k \leq \varepsilon_*, m_k \leq M_{max}, c_k \leq C_{max}, io_k \leq IO_{max}, \quad (2.11)$$

нормовані $\phi_r(x_k) \in [0, 1]$.

УРНП трактуємо як імовірнісні гарантії виконання ЦПРС за вікном W (доля дотриманих запитів $\geq 1-\delta$) [77; 93; 94; 114].

2.3.1. Організація засобів моніторингу та зворотного зв'язку у платформі автоматизації проєктування та виконання бізнес-сервісів

Концептуальна схема рівневої організації платформи автоматизації проєктування та виконання бізнес-сервісів (Рис. 2.2) відображає послідовну організацію спостереження, оброблення даних і керуючих впливів у межах платформи. Схема поєднує функціональні рівні, взаємодія яких забезпечує простежуваність життєвого циклу бізнес-процесу: від підготовки моделей до фіксації подій виконання, накопичення телеметрії, її аналітичного опрацювання та формування рішень для адаптації параметрів роботи. У межах такої побудови моніторинг розглядається не як збір метрик, а як складова узгодженого контуру, у якому аналітичні результати та контроль за цільовими показниками рівня сервісу використовуються для налаштувань і обмежень, спрямованих на підтримання стабільності виконання під навантаженням.

Рівень проєктування охоплює створення й редагування моделей у редакторі BPMN/DMN, перевірку та валідацію моделей, керування версіями (зокрема моделей і DMN-таблиць), а також збереження артефактів у репозиторії моделей і сховищі моделей. Репозиторій використовується для збереження актуальних проєктних артефактів, що дає змогу відтворювати конфігурації та підтримувати узгодженість між проєктними моделями й їх використанням під час виконання. Зв'язок із рівнем виконання реалізується через застосування підготовлених моделей під час запуску процесів і рішень, що спрямовано на зменшення розриву між проєктною та експлуатаційною частинами системи.

Рівень виконання включає рушій DMN, рушій виконання BPMN-процесів, сервіси та запити, а також інфраструктурний контур (ресурси, черги, навантаження). У процесі роботи формуються основні сигнали телеметрії: бізнес-події (старт, завершення, помилки, таймери), характеристики взаємодії із сервісами (затримки, помилки, граничний час очікування) та інфраструктурні метрики (ЦП, ОП, В/В). Окремо виділено модуль застосування політик, який пов'язаний із

виконанням та інфраструктурою і використовується як точка реалізації керуючих обмежень, що відповідають режимам керованої деградації.

Підсистема моніторингу, представлена конвеєром телеметрії (Рис. 2.1), приймає події та метрики з рівня виконання, забезпечує їх збирання та подальше оброблення перед збереженням. У межах цього конвеєра дані приводяться до узгоджених форматів і структур, що забезпечує можливість їх подальшого аналізу та зіставлення між різними класами сигналів. Результати конвеєра спрямовуються до рівня збереження та доступу до даних телеметрії та журналів, де розділено зберігання журналів подій (логів), часових рядів метрик, трасувань і слідів виконання процесу, а також каталогу (словника) подій та атрибутів як опорного шару семантики та структури даних.

Аналітичний рівень використовує накопичену телеметрію для формування панелей і звітів стану та продуктивності, а також для виявлення відхилень/аномалій і їх локалізації. Окремим елементом виступає ЦПРС-контроль і сповіщення, який пов'язує результати аналітичного опрацювання з оцінюванням виконання цільових показників рівня сервісу та ініціюванням процедур реагування.

Рівень керування включає модуль керування політиками, політики масштабування та конфігурації виконання, а також акторів змін, які ініціюють або погоджують налаштування потоку робіт, паралельності та квот. Сукупно це формує замкнений контур зворотного зв'язку, у якому аналітика та контроль ЦПРС підтримуються керуючими політиками, що застосовуються на рівні виконання.

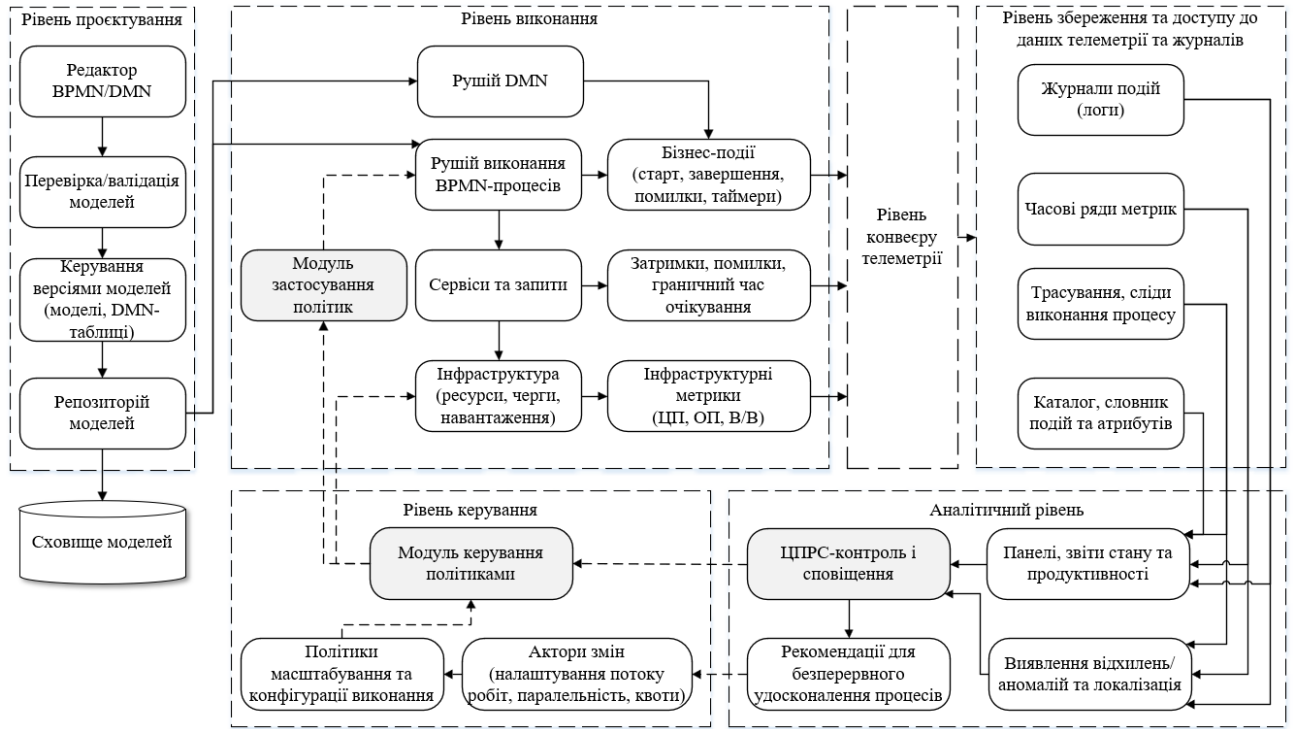


Рис. 2.2. Концептуальна схема рівневої організації платформи автоматизації проєктування та виконання бізнес-сервісів

Для представлення стану системи використовується вектор вимірювань

$$y_k = (q_k, p_k, m_k, c_k, io_k, \hat{L}_k^{p95}) \quad (2.12)$$

Цей вектор є основою для подальших розрахунків і оцінювання стану системи. Для уніфікації різномірних даних застосовуються нормувальні функції $\phi_r(\cdot)$, які переводять значення показників у діапазон $[0, 1]$. Наприклад, нормування латентності визначається як

$$\phi_{lat}(l_k) = \min(1, \max(0, \frac{l_k}{L_*^{p95}})) \quad (2.13)$$

Використання оперативної пам'яті описується виразом

$$\phi_m(m_k) = \min(1, \max(0, \frac{m_k}{M_{max}})) \quad (2.14)$$

Аналогічно визначаються функції для ЦП та В/В. Вагові коефіцієнти w_r для окремих метрик визначаються експериментальним шляхом із застосуванням планування експерименту та аналізу чутливості за перцентильними показниками затримки (p95/p99). Це дозволяє визначати вагу кожного ресурсу з урахуванням

невизначеностей, притаманних середовищам безсерверних обчислень (FaaS) і графам взаємодії мікросервісів. Калібрування параметрів α , β , σ здійснюється на основі історичних логів і потокового майнінгу процесних трас, що дозволяє виконувати оцінювання без модифікації застосунку [103; 107; 113; 117].

Отримані нормовані дані накопичуються у спеціалізованих сховищах телеметрії та доступу до даних: журналах подій (логах), базах часових рядів метрик, сховищах трасування зі слідами виконання процесу, а також у каталозі (словнику) подій та атрибутів. Розділення цих сховищ підтримує незалежний аналіз логів, метрик і процесних слідів та спрощує їх зіставлення. Зазначені сховища формують основу аналітичного рівня, який забезпечує побудову панелей і звітів стану та продуктивності, виконує виявлення відхилень/аномалій і їх локалізацію, підтримує ЦПРС-контроль і сповіщення, а також формує рекомендації для безперервного удосконалення процесів.

Аналітичний рівень забезпечує моніторинг через панелі та сповіщення і підтримує використання накопичених даних для подальшого аналізу. Виявлення відхилень реалізується через зіставлення показників із цільовими межами, що дає змогу локалізувати проблемні ділянки у виконанні процесів, сервісів і запитів. ЦПРС-контроль реалізується шляхом оцінювання наближення до допустимих меж і ризику порушення цільових показників, що забезпечує своєчасне реагування на потенційні проблеми [95; 113; 117; 121; 127].

Керування формує замкнений контур зворотного зв'язку, забезпечуючи вироблення політик масштабування та конфігурації виконання і їх застосування. Політики узгоджуються у модулі керування політиками та реалізуються через акторів змін (налаштування потоку робіт, паралельності, квот), а також через модуль застосування політик на рівні виконання. Застосування політик впливає на виконання BPMN-процесів, сервіси/запити та інфраструктурні ресурси, поєднуючи моніторинг із керуванням [110; 111; 118; 126].

Побудова засобів моніторингу передбачає стійкість до збоїв за рахунок використання черг із підтвердженням, ідемпотентних обробників і механізмів

повторних спроб. Також передбачається контроль доступу, захист бізнес-даних через псевдонімізацію в процесних трасах і підтримання аудиту змін DMN-правил.

Представлена концептуальна схема (Рис. 2.2) відображає багаторівневу організацію засобів моніторингу та керування виконанням бізнес-процесів у платформі автоматизації проєктування та виконання бізнес-сервісів. Вона поєднує рівень проєктування моделей, рівень виконання (рушій BPMN і рушій DMN), підсистему моніторингу (конвеєр телеметрії), сховища даних, аналітичний рівень і рівень керування в узгоджений контур, орієнтований на підтримання стабільності виконання в умовах змін навантаження.

2.3.2. Організація модуля контролю запропонованої системи моніторингу виконання бізнес-процесів

Організація модуля контролю в запропонованій системі моніторингу ґрунтується на замкненому контурі “спостереження - оцінювання стану - вибір керуючої дії - виконання – аудит”, у якому керуючі рішення формуються на основі подій життєвого циклу кроків бізнес-процесів та телеметрії ресурсів.

У загальному випадку керування виконанням може описуватися вектором керування, що включає

- 1) масштабування прийому/виконання;
- 2) рівень паралельності (кількість потоків/виконавців);
- 3) ліміти введення-виведення;
- 4) правила пріоритезації (наприклад, класи задач).

Параметри політик виконання (рівень експлуатації) - фіксуються/оновлюються адміністраторами системи (акторами) у вигляді регламентів і профілів роботи, а саме, гранична паралельність пулу, коефіцієнти дроселювання прийому/виконання, обмеження введення-виведення, тощо.

Оперативне керування (рівень контролера/диспетчера на кожному кроці k - зводиться до вибору, які саме задачі запускати зараз у межах доступної паралельності та ресурсних обмежень.

Таке розмежування вводиться з міркувань надійності виконання у транзакційно-насиченому середовищі: динамічне підвищення паралельності (керування кількістю потоків у реальному часі) може призводити до непропорційного зростання конкуренції за блокування та транзакційних конфліктів на рівні бази даних (зростання часу очікування блокувань, повторів транзакцій, відкатів), що, у підсумку, погіршує дотримання часових обмежень запуску. Тому в рамках даного модуля контролю паралельність вважається заданою політикою (керується акторами), а адаптація в онлайні реалізується через правило відбору/упорядкування задач та допуск за ресурсами.

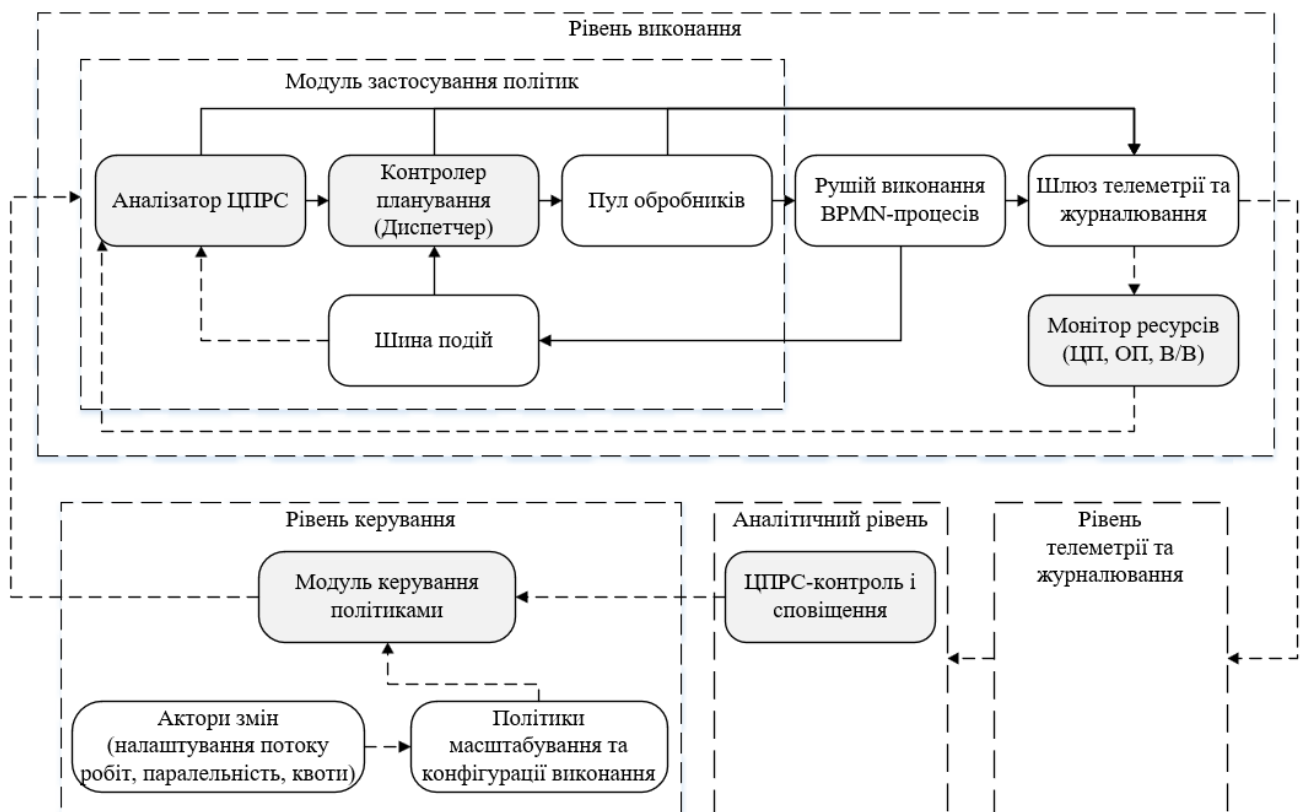


Рис. 2.3. Схема взаємодії компонентів модуля застосування політик у контурі виконання бізнес-процесів

На рівні виконання (Рис. 2.3) розміщено рушій виконання BPMN-процесів і пул обробників, які взаємодіють під час обробки задач. Пул обробників виконує задачі, що надходять від контролера планування (диспетчера), та повертає результати до рушія виконання. Рушій передає події до шини подій, а дані спостереження - до шлюзу телеметрії та журналювання. Через шлюз формуються потоки до монітора ресурсів (ЦП, ОП, В/В) та на рівень телеметрії і журналювання.

Контур контролю на рівні виконання включає аналізатор ЦПРС, контролер планування (диспетчер), шину подій і монітор ресурсів. Шина подій відображає безперервний зв'язок між виконанням і контуром керування: вона акумулює події життєвого циклу процесів і передає їх диспетчеру. Монітор ресурсів надає показники використання ресурсів для аналізатора. Аналізатор ЦПРС оцінює стан виконання відносно сервісних угод і передає результати диспетчеру. Диспетчер, у свою чергу, керує подачею робіт у пул обробників. Позначення “модуль застосування політик” у схемі відображає вплив політик на зазначені компоненти контуру виконання.

На рівні керування політиками відображено акторів змін, політики масштабування та конфігурації виконання, а також модуль керування політиками. Модуль керування політиками приймає та застосовує визначені політики і взаємодіє з блоком “ЦПРС-контроль і сповіщення” аналітичного рівня. Рівень телеметрії та журналювання забезпечує надходження даних до аналітичних компонентів, що використовується для контролю ЦПРС, формування сповіщень і підтримання зворотного зв'язку для керування політиками.

Нехай на кроці k у черзі перебуває множина задач Q_k . Для кожної задачі задано її профіль $J_i = (r_i, e_i, d_i)$.

Оперативне рішення контролера на кроці k подається як вектор керування запуском:

$$u_k = l_k = (l_1(k), l_2(k), \dots, l_N(k)), \quad (2.15)$$

де $l_i(k) \in \{0, 1\}$, $l_i(k) = 1$ означає “запустити задачу J_i на кроці k ”, $l_i(k) = 0$ - “не запускати”.

При цьому доступна кількість запусків на кроці k обмежена величиною (кількість вільних виконавців/слотів запуску), яка визначається політикою паралельності та поточним завантаженням.

Для визначення розміру черги та кількості активних потоків для стану x_k використовуються співвідношення:

$$\begin{aligned} q_{k+1} &= \max(0, q_k + |A_k| - |S_k|), \\ p_{k+1} &= \max(0, p_k + |S_k| - |F_k|), \end{aligned} \quad (2.16)$$

де A_k - множина задач, що надійшли (активовані тригерами/подіями процесу) на кроці k ;

S_k - множина задач, що стартували на кроці k ;

F_k - множина задач, що завершилися на кроці k .

Токен-баланс (оновлення вільних токенів) задається як:

$$(C_{k+1}, M_{k+1}, IO_{k+1}) = (C_k, M_k, IO_k) - \sum_{J_i \in S_k} \hat{r}_i(k) + \sum_{J_i \in F_k} \hat{r}_i(k), \quad (2.17)$$

де $\hat{r}_i(k)$ – очікуване використання ресурсів задачею J_i на кроці k . При старті задачі токени резервуються, після завершення - повертаються.

Правило токен-фільтрації визначається як:

$$\begin{aligned} \hat{r}_i(k) &\leq (C_k, M_k, IO_k), \\ J_i &\in D_k, \end{aligned} \quad (2.18)$$

де D_k - множина задач, допущених після токен-фільтра на кроці k .

Після токен-фільтрації задачі з D_k упорядковуються за правилом, наприклад, за зростанням d_i .

Визначення кількості запусків на кроці. Кількість задач, що можуть бути запуснені на кроці k , визначається $S_k = \min(P_{max} - p_k, |D_k|)$.

Метою керування є мінімізація сумарних очікуваних втрат:

$$\min_{l(k), \dots, l(k+H-1)} \sum_{j=0}^{H-1} (w_1 miss_{k+j} + w_2 delay_{k+j} + w_3 \|z_{k+j} - \bar{z}\|^2 + w_4 q_{k+j}) \quad (2.19)$$

де $w_1 \dots w_4$ - вагові коефіцієнти;

$\bar{z}$ - бажаний/плановий профіль використання ресурсів;

$miss_{k+j}$ - індикатор/кількість дедлайн-пропусків;

$delay_{k+j}$ - середня затримка старту;

q_{k+j} - розмір черги;

$\|z_{k+j} - \bar{z}\|^2$ - штраф за відхилення від бажаного стану;

$$z_k = (C_k, M_k, IO_k), z_k^{(n)} = \left(\frac{C_k}{C_{max}}, \frac{M_k}{M_{max}}, \frac{IO_k}{IO_{max}} \right)$$

Приклад для компоненти пам'яті (нормоване перевищення над еталоном)

$$\varphi(M_k) = \left(\max \left(0, \frac{M_e - M_{k+j}}{M_{max}} \right) \right)^2$$

M_e - еталонне (бажаний) мінімальний запас оперативної пам'яті.

Оптимізація виконується за умов динаміки системи та ресурсних/часових обмежень для $j=0, \dots, H-1$:

Динаміка (2.20)

$$x_{k+j+1} = f(x_{k+j}, u_{k+j})$$

Обмеження пулу потоків:

$$p_{k+j} + |S_{k+j}| - |F_{k+j}| \leq P_{max}, 0 \leq p_{k+j} \leq P_{max}$$

Ресурсна здійсненність запуску:

$$\sum_{i \in S_{k+j}} c_i \leq C_{k+j}, \sum_{i \in S_{k+j}} m_i \leq M_{k+j}, \sum_{i \in S_{k+j}} io_i \leq IO_{k+j}$$

Межі ресурсів:

$$0 \leq C_{k+j} \leq C_{max}, 0 \leq M_{k+j} \leq M_{max}, 0 \leq IO_{k+j} \leq IO_{max}$$

Задача оптимізації виконується на скінченному горизонті H із врахуванням часових та ресурсних обмежень. Після отримання плану керування до виконання застосовують лише керуючу дію першого кроку, а надалі задачу оптимізації перераховують для оновленого стану системи.

Продуктивність контуру контролю підтримується тим, що в схемі (Рис. 2.3) виділено окремі засоби телеметрії та журналювання, які виводять дані зі сторони виконання через шлюз телеметрії та журналювання. Збір показників ресурсів (ЦП,

ОП, В/В) здійснюється монітором ресурсів, а події життєвого циклу процесів передаються через шини подій у контур аналізу та планування. Таке розмежування потоків спостережуваності та контуру виконання зменшує накладні витрати та обмежує вплив процесів моніторингу на безпосереднє виконання бізнес-процесів.

Висновки до розділу 2

Виконано дослідження механізмів моніторингу та підтримання стабільного виконання бізнес-процесів у платформах автоматизації проєктування та виконання бізнес-сервісів за умов змінного навантаження й обмежених обчислювальних ресурсів. Показано, що ключовою передумовою безпечного виконання є поєднання механізмів керування навантаженням і ресурсами з наскрізною спостережуваністю, яка забезпечує кореляцію технічних метрик із подіями бізнес-рівня та дозволяє обґрунтовано застосовувати політики деградації.

Розглянуто засоби підвищення стабільності у п'ятих категоріях: керування навантаженням, керування ресурсами, узгодженість і стійкість (включно з компенсаційними шаблонами), моніторинг/логування/трасування, а також відновлення та самозцілення. Така класифікація дозволила виділити типові ризики (перевантаження, дефіцит ОП, каскадні збої, аномальні затримки) і відповідні засоби протидії на рівні платформи.

Проаналізовано архітектурні підходи до телеметрії - агентський та інструментований - і обґрунтовано їхню роль у задачах спостережуваності: агентський підхід доцільний для загального контролю здоров'я інфраструктури, тоді як інструментована телеметрія є необхідною для точного зіставлення BPMN-кроків із технічними вимірюваннями та підтримки аналізу першопричин на рівні екземплярів процесів.

Сформовано вимоги до моделі даних моніторингу (події, метрики, контекстні атрибути кореляції) та обґрунтовано принцип агрегування “від загального до деталізації”, за якого первинно аналізуються агреговані показники (зокрема p_{95}), а

детальний розбір виконується після спрацювання тригерів відхилень. Запропоновано концептуальну модель підсистеми моніторингу і конвеєра телеметрії, що охоплює збір сигналів, агрегацію/нормалізацію, аналітичну обробку, збереження та візуалізацію, а також блок оцінювання виконання ЦПРС і реагування.

Розроблено узгоджену організацію контуру зворотного зв'язку платформи, яка поєднує рівні проєктування, виконання, спостережуваності, аналітики та керування політиками. Уточнено організацію модуля контролю як компонента, що перетворює телеметрійні сигнали у керуючі впливи (дроселювання прийому/виконання, зміна паралельності, обмеження введення-виведення, пріоритезація класів задач) та забезпечує керовану деградацію з аудитом рішень. Отримані результати є основою для подальшого розділу, у якому розглянуто методи планування та керування виконанням кроків бізнес-процесів з урахуванням дедлайнів і багатовимірних ресурсних обмежень.

РОЗДІЛ 3 РОЗРОБКА МЕТОДУ РОЗПОДІЛУ ВИКОНАННЯ КРОКІВ БІЗНЕС-ПРОЦЕСІВ У ПЛАТФОРМІ АВТОМАТИЗАЦІЇ ПРОЄКТУВАННЯ ТА ВИКОНАННЯ БІЗНЕС-СЕРВІСІВ

3.1. Вибір і обґрунтування моніторингової моделі у платформі автоматизації проєктування та виконання бізнес-сервісів

Сучасні системи автоматизації бізнес-сервісів потребують інтеграції двох взаємопов'язаних процесів: автоматизованого проєктування, що визначає структурні та поведінкові характеристики сервісів, та виконання бізнес-процесів, які забезпечують досягнення операційних цілей у реальному середовищі з обмеженими ресурсами. Для цього необхідні методи, що враховують багатофакторність впливів - часові дедлайни старту кроків, обмеження оперативної пам'яті, ЦП та В/В, а також угоди про рівень надання послуг (ЦПРС) [128; 129].

У розділі 2 сформовано модель моніторингу, конвеєр телеметрії та контур “оцінювання ЦПРС - реагування”, в якому для задач зі стартовими дедлайнами ключовими є фіксація моментів надходження та запуску кроків, контроль порушень дедлайну старту, кореляція бізнес-подій з технічними метриками.

Зазначену моніторингову модель у цьому розділі застосовано як рівень моніторингу під час розробки методу планування та диспетчеризації кроків бізнес-процесів з урахуванням ресурсних обмежень.

У сучасних платформах автоматизації проєктування та виконання бізнес-сервісів кожен бізнес-процес моделюється за допомогою нотації BPMN [130], що забезпечує формалізований опис потоків робіт, взаємодій і залежностей між кроками БП. Кожен крок БП розглядається як елементарна одиниця виконання (задача планувальника), для якої відомі очікувані характеристики: профіль споживання ресурсів, дедлайн допустимої затримки старту та тривалість обробки. У результаті платформу можна розглядати як динамічну систему керування

ресурсами, у межах якої потрібно забезпечити своєчасний запуск і стабільне виконання множини кроків.

Розглянемо дискретний час із кроком Δt , де моменти часу задаються як $t = k\Delta t$. На кроці k доступний вектор ресурсних квот:

$$R_k = (R_k^{CPU}, R_k^{MEM}, R_k^{IO}), \quad (3.1)$$

Множина задач (кроків бізнес-процесів), що очікують запуску на кроці k , позначається як

$$J_k = \{j_i = (r_i, D_i, e_i, t_i^{arr}) \mid i = 1, \dots, n\}, \quad (3.2)$$

де

$r_i = (r_i^{CPU}, r_i^{MEM}, r_i^{IO})$ - очікуване споживання ресурсів,

D_i - дедлайн допустимої затримки старту,

e_i - очікувана тривалість виконання;

t_i^{arr} - момент надходження задачі в чергу (фіксується інструментованою телеметрією).

Для задачі j_i абсолютний дедлайн старту визначається як

$$d_i = t_i^{arr} + D_i$$

Виконання кроків здійснюється за політикою планування Π_k , яка використовує: токен-допуск за ресурсами, правило впорядкування (наприклад, за зростанням d_i / D_i), обмеження паралельності пулу обробників. Вектор керування (2.4) інтерпретується як параметризація Π_k , зокрема через масштабування прийому/виконання γ та налаштування пулу/лімітів.

Кожне завдання виконується атомарно - або повністю в межах виділених ресурсів або відхиляється. Виконання здійснюється за політикою планування Π , що враховує дедлайни запуску, пріоритети та ресурсні обмеження.

Задача полягає у пошуку такого розкладу Π , який мінімізує кількість відмов, забезпечуючи запуск більшості завдань до настання їхніх дедлайнів:

$$\min_{\Pi} \sum_{i=1}^n \delta_i \quad (3.3)$$

де $\delta_i = 1$ - якщо j_i -задача не була запущена до моменту d_i і $\delta_i = 0$, якщо задача запущена вчасно. Такий показник узгоджується з використанням “індикатора/кількості дедлайн-пропусків” у функції втрат контуру керування (2.19), $\delta_i = 0$ - якщо задача запущена вчасно.

Далі приймаються такі припущення:

- ресурси системи обмежені та задаються наперед у вигляді вектора доступності (квот);
- задачі не поділяються на підзадачі (атомарність на рівні рішення “запуск/не запуск”);
- моменти надходження задач є довільними, при цьому система підтримує чергу очікування;
- дедлайни задаються в одній часовій шкалі й інтерпретуються як дедлайни початку виконання (старту).

У цьому контексті формулюються дві базові теореми, які визначають умови допустимості та стійкості системи виконання бізнес-процесів.

Теорема 1. Умова локальної ресурсної допустимості для запуску на кроці k .

Твердження.

Нехай $A_k \subseteq J_k$ - множина задач, обраних політикою Π_k для запуску на кроці k . Якщо для кожного ресурсу r виконується:

$$\sum_{j_i \in A_k} \frac{r_i^k}{R_k} \leq 1 \quad (3.4)$$

то запуск усіх задач із множини A_k на кроці k є ресурсно допустимим, тобто сумарна потреба у ресурсі r не перевищує доступну квоту R_k у цьому такті.

Доведення.

Нерівність (3.4) є багатовимірним узагальненням умови [131] для одноресурсних систем реального часу з періодичними задачами (критерій завантаження $\sum \frac{C_i}{T_i} \leq 1$). Нерівність відображає нормовану умову невиходу за межі доступних ресурсів на кроці k : для кожного ресурсу r сукупний запит задач із A_k не

перевищує квоту R_k . Відповідно, запуск задач із A_k не призводить до перевищення ресурсних обмежень у момент старту. Якщо ж для деякого ресурсу r нерівність (3.4) порушується, то при одночасному запуску всіх задач із A_k квота буде перевищена, а отже такий запуск не є допустимим за заданими обмеженнями.

Умова (3.4) є локальною (для конкретного такту k) та забезпечує відсутність конфлікту за ресурси саме під час запуску обраного набору задач. Вона не є достатньою для гарантування виконання всіх дедлайнів для довільного портфеля задач у часовому горизонті, оскільки дедлайни залежать також від порядку запуску, часу виконання, динаміки черги та пропускної здатності системи. Теоретичною основою є теорія допустимості розкладів у багаторесурсних системах [132, 133], що використовується для аналізу систем із кількома обмеженнями ресурсів.

Теорема 2. Умова стохастичної стійкості потоку задач

Нехай інтенсивність надходження задач дорівнює λ_k (задач/с), $\gamma_k \in [0, 1]$ - коефіцієнт масштабування прийому/виконання, μ_k - середня пропускна здатність пулу обробників на кроці k ,), яка залежить від поточного стану ресурсів. Якщо виконується

$$\mathbb{E}[\gamma_k \lambda_k] < \mathbb{E}[\mu_k], \quad (3.6)$$

то математичне очікування довжини черги задач $\mathbb{E}[q_k]$ є скінченним, тобто система перебуває у стійкому стані:

$$\lim_{t \rightarrow \infty} \mathbb{E}[q_k] < \infty. \quad (3.7)$$

Доведення.

Розглянемо систему масового обслуговування типу M/G/1 з дисципліною обслуговування за пріоритетами. Коефіцієнт завантаження системи визначається як

$$\rho = \frac{\mathbb{E}[\gamma_k \lambda_k]}{\mathbb{E}[\mu_k]} \quad (3.8)$$

Згідно з класичною теорією черг [146], якщо

$\rho < 1$, то система є ергодичною: розподіл довжини черги має скінченне математичне очікування, а середній час очікування задач обмежений.

$\rho \geq 1$ - черга зростає необмежено, що відповідає перевантаженню ресурсів і порушенню дедлайнів. Таким чином, умова (3.6) є критерієм стохастичної стійкості системи керування потоками задач.

Теорема ґрунтується на положеннях теорії систем масового обслуговування, які описують динаміку черг за змінної інтенсивності надходження запитів. Вона задає критерій узгодження між інтенсивністю вхідного потоку задач і пропускну здатністю платформи в поточному стані ресурсів. Виконання умови (3.6) означає, що середня інтенсивність обслуговування перевищує ефективну інтенсивність надходження задач, тому середня довжина черги залишається скінченною; при цьому висновок не вимагає фіксованого розподілу тривалостей виконання задач.

Розглянемо приклад бізнес-процесу, який складається з трьох кроків із нормованими профілями споживання ресурсів:

$$S_1 = (r^{cpu} = 0.3, r^{mem} = 0.4, r^{io} = 0.1),$$

$$S_2 = (r^{cpu} = 0.2, r^{mem} = 0.4, r^{io} = 0.2),$$

$$S_3 = (r^{cpu} = 0.1, r^{mem} = 0.4, r^{io} = 0.1).$$

Сумарні вимоги становлять:

$$\sum_{i=1}^3 r_i^{cpu} = 0.6 < 1, \sum_{i=1}^3 r_i^{mem} = 1.2 > 1, \sum_{i=1}^3 r_i^{io} = 0.4 < 1$$

Отже, за обмеженням на оперативну пам'ять одночасний запуск усіх трьох кроків є недопустимим, оскільки сумарна потреба перевищує доступний ресурс. У такій ситуації необхідне планування, яке визначає порядок запуску задач з урахуванням локальної ресурсної допустимості (3.4) та критерію стійкості потоку задач (3.6).

Теореми 1 та 2 формують формальну основу для побудови методу планування та диспетчеризації задач у платформах автоматизації проєктування та виконання бізнес-сервісів. Перша задає детерміновану умову локальної допустимості запуску множини задач за ресурсними обмеженнями, друга — стохастичну умову стійкості для потоку задач у разі змінної інтенсивності надходження. Сукупно ці умови визначають область допустимих режимів роботи, у межах яких можна зменшувати

частку порушень дедлайнів старту за умови підтримання стабільного навантаження.

Виконання обох умов одночасно характеризує збалансований режим, у якому рішення щодо запуску кроків бізнес-процесів приймаються без перевищення ресурсних квот, а динаміка черги залишається керованою за наявності випадкових навантажень. Це створює теоретичне підґрунтя для подальшого формування ресурсно-орієнтованої моделі планування задач у межах автоматизованої платформи проектування та виконання бізнес-сервісів.

3.2. Розробка алгоритму ресурсно-орієнтованого планування задач із використанням гібридного підходу

Розробка алгоритму планування у платформі автоматизації проектування та виконання бізнес-сервісів потребує врахування ресурсних обмежень, дедлайнів старту задач, пріоритетів, а також динамічної зміни навантаження. У попередньому підрозділі сформульовано дві базові умови: локальну ресурсну допустимість запуску (детермінована складова) та стохастичну стійкість потоку задач (динамічна складова). На їх основі формується гібридний алгоритм планування, який поєднує реактивне керування та підходи, орієнтовані на дедлайни, що узгоджується з висновками оглядових робіт, де розподіл ресурсів у BPM і процес-майнінгу розглядається як багатокритеріальна задача, що поєднує часові цілі, ресурсні обмеження та правила призначення виконавців [134]. Метою алгоритму є підвищення частки задач, запущених до їхніх дедлайнів старту, за одночасного збереження керованої динаміки черги та зменшення відмов/відкладень запуску через дефіцит ресурсів.

Гібридний метод передбачає інтеграцію двох взаємопов'язаних компонентів:

- прогнозно-оцінювального компонента, що оцінює потенційне навантаження та перевіряє локальну ресурсну допустимість запуску;

- реактивного компонента, який у режимі реального часу коригує порядок запуску задач залежно від поточного стану ресурсів і черги.

Поділ на прогнозно-оцінювальний і реактивний контури узгоджується з роботами, у яких в архітектурі систем керування бізнес-процесами (BPMS) виокремлюється компонент керування ресурсами для ухвалення рішень на основі стану системи та політик призначення [135]. Водночас формалізація ресурсних обмежень і допустимих призначень у вигляді правил та обмежень планування є типовим способом задання коректних рамок для побудови розкладу в робочих процесах [136].

Відмінність цього методу від традиційних алгоритмів “першим – найраніший дедлайн” (EDF) або перший “прийшов-перший оброблений” (FCFS) полягає у використанні механізму ресурсних токенів, що дає змогу враховувати не лише часову, а й ресурсну складову задачі. Кожна задача при надходженні характеризується вектором очікуваних потреб у ресурсах $r_i = (r_i^{CPU}, r_i^{MEM}, r_i^{IO})$, на основі якого формується набір токенів ϕ_i , що відображають її споживання у пропорції до доступного ресурсу в момент часу t . Таким чином, алгоритм діє як система розподілу обмежених токенів між задачами з урахуванням їхніх дедлайнів старту d_i та пріоритетів pr_i . Концептуально така постановка узгоджується з відомими підходами багаторесурсного розподілу, де коректність порівняння запитів до різних типів ресурсів досягається через нормування і врахування “домінуючого” ресурсу для кожної задачі [138].

Для кожної задачі J_i у черзі визначається коефіцієнт придатності до виконання:

$$\psi_i = \alpha \frac{1}{d_i - k} + \beta \frac{pr_i}{Pr_{max}} + \gamma \left(1 - \frac{\|r_i^k\|}{\|R_k\|}\right) \quad (3.9)$$

де α, β, γ - вагові коефіцієнти, що задають відносну важливість часової, пріоритетної та ресурсної складових. Функція ψ_i визначає узагальнену міру “готовності” задачі до запуску. Задача з максимальним значенням ψ_i у межах доступних ресурсів обирається для виконання. У випадку, коли кілька задач мають однакові значення ψ_i , додатковим критерієм виступає найменший векторний надлишок ресурсу $\Delta r_i^k = R_k - r_i^k$, що забезпечує більш повне використання

потужностей без ризику перевантаження. Обраний варіант відповідає постановкам багатовимірного “пакування” (packing) у планувальниках, у яких враховується потреба зменшення фрагментації ресурсів у багаторесурсному середовищі [139].

Система працює в циклічному режимі: на кожній ітерації відбувається оцінка ресурсної доступності, обчислення показників ψ_i , вибір задачі та оновлення стану ресурсів. Якщо поточне значення $\sum_i \frac{r_i^k}{R^k}$ перевищує одиницю хоча б для одного типу ресурсу k , відповідно до умови (3.4), система тимчасово призупиняє запуск нових задач, переходячи у режим реактивного балансування. У цьому режимі алгоритм виконує відкладене планування - задачі з меншими дедлайнами, але високими ресурсними вимогами, можуть бути тимчасово відсунуті на користь задач із меншою ресурсоемністю, що підвищує пропускну здатність системи в умовах дефіциту ресурсів.

Перевагою такого підходу є можливість підтримання рівноваги між двома протилежними цілями: мінімізацією кількості невиконаних задач і збереженням стабільності системи. При дотриманні умови стохастичної стійкості (3.4) алгоритм забезпечує обмежене зростання черги навіть при нерівномірному потоці задач. Відмінною рисою є динамічна адаптація параметрів α , β , γ на основі метрик моніторингу, що відображають стан ресурсів: при зростанні навантаження коефіцієнт γ збільшується, посилюючи вагу ресурсної складової, тоді як у режимі низького навантаження підвищується вагомість часової компоненти α .

Для коректності вибору задачі у гібридному режимі алгоритм перевіряє дві умови:

- локальну допустимість (наявність достатніх ресурсів для запуску задачі);
- стохастичну стійкість (оцінка завантаження/черги відповідно до (3.6)–(3.8)).

Якщо перша умова порушується, задача відкладається або не допускається до запуску; якщо друга умова наближається до межового режиму, застосовуються обмеження запусків (зменшення інтенсивності прийому/старту нових задач або обмеження черги). Таким чином реалізується принцип адаптації до перевантаження з урахуванням керованості системи.

Загальна послідовність кроків гібридного планування може бути подана у вигляді псевдокоду:

Алгоритм: Гібридне ресурсно-орієнтоване планування

поки $S_k \neq \emptyset$ робити

оновити R_k

для кожної задачі J_i у S_k робити

якщо $r_i \leq R_k$ тоді

додати J_i до A_k

якщо $A_k \neq \emptyset$ тоді

для кожної задачі J_i у A_k робити

обчислити $\psi_i(k) = \alpha/(d_i - k) + \beta \cdot (pr_i/Pr_{max}) + \gamma \cdot (1 - \|r_i\|/\|R_k\|)$

вибрати $J^* = \max_{\{J_i \in A_k\}} \psi_i(k)$

зарезервувати токени ресурсів r^* для J^*

запустити J^*

$R_k = R_k - r^*$

інакше

очікування звільнення ресурсів

оновити $S_{\{k+1\}}$, прибрати запущені/завершені, додати нові

оновити показники угод про рівень надання послуг та статистику черги

$k := k + 1$

Якщо середнє значення коефіцієнта завантаження (3.8) наближається до 1, застосовується режим обмеження запусків, за якого нові задачі допускаються переважно після звільнення ресурсів. У такий спосіб підтримується стабілізаційний

контур керування, що спирається на принципи теорії систем масового обслуговування.

Табл. 3.1.

Порівняльна характеристика алгоритмів планування задач у ресурсно-обмежених системах

Ознака порівняння	Класичний EDF	Реактивне планування	Гібридний ресурсно-орієнтований підхід
Орієнтація	Часова: пріоритет за дедлайном завершення	Ресурсна: орієнтація на поточний стан ресурсів у реальному часі.	Комбінована: враховує дедлайн старту/початку обробки та ресурсну допустимість.
Критерій вибору задачі	Обрати задачу з найменшим дедлайном серед готових.	Обрати задачу, яка є ресурсно допустимою за поточних умов та забезпечує найефективніше використання доступних ресурсів без перевищення встановлених обмежень.	Обрати задачу з максимальною корисністю ψ_i
Враховується ЦПРС	Частково: дедлайни є, але немає гарантій, що старт/виконання	Динамічно: ЦПРС опосередковано підтримується через обмеження за ресурсами (менше перевантажень).	Повністю із зворотним зв'язком: ЦПРС напряду входить у та/або правила

	не зірвуться через дефіцит ресурсів.		допуску і коригується за телеметрією.
Стійкість до перевантажень	Низька: при перевантаженні черга росте, дедлайни масово порушуються, можливі затримки.	Висока: система намагається не запускати зайве при дефіциті ресурсів	Керована: поведінка задається політикою; залежить від завантаження ρ та правил допуску (можна тримати систему в стабільній зоні)
Реалізація у ПНОП	Потрібно реалізувати диспетчер/черги.	Потрібен моніторинг, правила гальмування/відкладення, власні механізми керування запуском у багатьох місцях.	Реалізується як універсальний компонент допуску: профіль ресурсів задач, менеджер токенів, правила старту.

Ефективність запропонованого підходу оцінюється через дві ключові метрики: частку своєчасно запущених задач та середню довжину черги очікування. Додатково, завдяки токен-орієнтованому механізму, виконується балансування між ЦП, ОП та В/В, що зменшує ризик фрагментації ресурсів у багатовимірному просторі [139].



Рис. 3.2. Взаємодія компонентів гібридного планування кроків бізнес-процесів

Гібридний планувальник може бути реалізований як сервіс BPM-модуля, який взаємодіє з чергою бізнес-процесів через API. Механізми прийняття рішень доцільно реалізовувати у вигляді плагінів із налаштуваннями правил вибору задач без необхідності перепрограмування системи. При цьому правила призначення/вибору можуть уточнюватися на основі аналізу журналів подій, де з подієвих даних видобуваються закономірності розподілу робіт і формуються рекомендації щодо ресурсного призначення [137]. Такий спосіб узгоджується з практикою використання процес-майнінгу для підтримки рішень у BPM [134].

Узагальнюючи, гібридний алгоритм ресурсно-орієнтованого планування забезпечує:

- поєднання детермінованого контролю за дедлайнами старту з динамічним балансуванням ресурсів;
- адаптацію до зміни навантаження через саморегулювальні коефіцієнти;
- дотримання умов теорем, які формалізують допустимість і стійкість розкладу.

Таким чином, розроблений гібридний метод створює основу для подальшого моделювання сценаріїв виконання бізнес-процесів.

3.3. Моделювання сценаріїв виконання кроків бізнес-процесів у середовищі з обмеженими ресурсами

Моделювання сценаріїв виконання кроків бізнес-процесів у середовищі з обмеженими ресурсами є етапом перевірки ефективності та коректності методів планування, описаних у попередніх підрозділах. Його мета полягає у визначенні поведінки системи за різних співвідношень між кількістю задач, рівнем доступних ресурсів, інтенсивністю потоку запитів і вимогами угод про рівень надання послуг. У літературі з моделювання та симуляції бізнес-процесів зазначається, що сценарне відтворення виконання процесів на основі процедурних графічних моделей є придатним для аналізу продуктивності та пошуку вузьких місць за умов ресурсних обмежень [140]. Окремо підкреслюється доцільність охоплення як детермінованих, так і стохастичних чинників навантаження під час оцінювання можливості підтримання стабільного режиму роботи [141].

У задачах розподілу ресурсів у BPMS урахування залежностей між задачами (наприклад, обмежень сумісності, передумов чи конфліктів доступу) розглядається як фактор, що впливає на можливість коректного призначення ресурсів і формування черги [145]. У роботах, присвячених пріоритетному розподілу людських ресурсів у бізнес-процесах, наголошується на важливості узгодження правил пріоритизації з політикою запуску задач у часі [146]. Система має обмежений вектор ресурсів R (2.2), який у процесі виконання змінюється внаслідок динамічного розподілу ресурсів між задачами. При моделюванні припускається, що час надходження задач описується стохастичним потоком з параметром інтенсивності $\lambda(t)$, а тривалість виконання - випадковою величиною з математичним очікуванням $E[T]$ та дисперсією $D[T]$ тривалості виконання задачі (час від моменту старту задачі до моменту її завершення).

$$K_{plan} = \frac{N_{miss}}{N_{total}} \quad (3.11)$$

де N_{miss} - кількість задач, для яких порушено вимогу: стартувати до своїх дедлайнів,

N_{total} - загальна кількість задач.

Для виконання ЦПРС необхідно, щоб $K_{plan} \geq P_{SLO}$, де P_{SLO} - заданий поріг ЦПРС (наприклад, 0.01 або 0.15). Додатково аналізується середня довжина черги очікування $E[q(t)]$ і коефіцієнт використання ресурсів $\rho_k = E[\sum_i \frac{r_i^k}{R_k}]$ для кожного типу ресурсу.

Загальна модель середовища виконується у вигляді ітераційного процесу, де на кожному кроці часу здійснюється оновлення стану ресурсів, запуск доступних задач, збір статистики виконання та корекція черги. При моделюванні використовується дискретно-подієвий підхід (discrete-event simulation), який дозволяє враховувати події надходження, старту, завершення або відхилення задач [141]. Для підвищення інтерпретованості моделі доцільним є явне подання черг, атрибутів заявок та точок прийняття рішень, що керуються даними, оскільки це полегшує узгодження процесної структури з логікою симуляції та аналізом метрик [142]. З позиції формалізації переходу від процесної моделі до симуляційної, в літературі розглядаються підходи перетворення BPMN-описів у дискретно-подієві формалізми, що спрощує відтворюваність експериментів і трасування подій [143]. Для задач оцінювання політик у ресурсно-обмежених умовах також використовуються ресурсно-орієнтовані розширення профілю із застосуванням шаблонів ресурсів робочих процесів (workflow resource patterns) [144]. У сукупності це забезпечує баланс між точністю та обчислювальною ефективністю, що важливо для практичного застосування у платформах автоматизації проєктування та виконання бізнес-сервісів із великою кількістю бізнес-процесів [140].

Результати моделювання дозволяють побудувати залежності ключових показників ефективності від рівня доступних ресурсів, інтенсивності потоку задач і параметрів алгоритму. Аналіз показує, що зі збільшенням навантаження (зростання $\lambda(t)$) система переходить через три характерні режими: нормальну експлуатацію, порогову стабільність і перевантаження. У першому режимі виконується умова (3.4), і всі задачі стартують до дедлайнів. У другому режимі показник ЦПРС K_{plan} поступово збільшується, але система ще зберігає стійкість

згідно з умовою (3.5). У третьому режимі черга стрімко зростає, що свідчить про порушення стохастичної стабільності.

Табл. 3.2.

Класифікація сценаріїв навантаження системи та їх відповідність теоретичним умовам стабільності

Тип сценарію	Характеристика навантаження	Відповідність теоремам	Очікувана поведінка системи
S_1 - номінальний	$\lambda < \lambda_{st}$ ресурси не обмежені	(3.4) і (3.5) виконуються	Система стабільна, $\text{ПРС} < 0.01$
S_2 - граничний	$\lambda \approx \lambda_{st}$ ресурси частково дефіцитні	(3.4) виконана, (3.5) на межі	ЦПРС збільшується, черга коливається
S_3 - перевантажений	$\lambda > \lambda_{st}$ ресурси в дефіциті	(3.4) порушена	Черга зростає, ЦПРС > 0.5
S_4 - стохастично нерівномірний	$\lambda(t)$ має піки та спади	часткове виконання (3.5)	Перемикання між стабільними та нестабільними станами
S_5 - реактивно збалансований	$\lambda(t)$ динамічно адаптується	обидві теореми виконуються	Система стабільна при коливаннях ресурсів

$$\lambda_{st}^{(k)} = \frac{R_k}{\mathbb{E}[r_k]} \quad (3.12)$$

$$\lambda_{st} = \min_k \lambda_{st}^{(k)}$$

Взаємозв'язок (Рис. 3.5) між середнім рівнем використання ресурсів та УРНП-задоволеністю у межах різних сценаріїв.

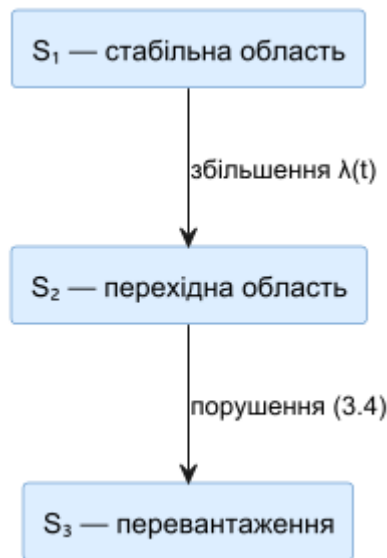


Рис. 3.5. Діаграма областей стабільності при зміні інтенсивності потоку задач

Припустимо, що існує два класи задач: ЦП-орієнтовані (переважно аналітичні обчислення) та В/В-орієнтовані (запити до сховищ або зовнішніх API). Окрім продуктивності, оцінюється також стабільність системи в часі. Для цього розраховується коефіцієнт варіації довжини черги:

$$C_v = \frac{\sqrt{D[q(t)]}}{E[q(t)]} \quad (3.13)$$

Низьке значення $C_v < 0.25$ свідчить про рівномірну роботу системи, тоді як $C_v > 0.5$ означає високі флуктуації, що є індикатором нестабільності або неузгодженості ресурсів. За результатами моделювання, при застосуванні гібридного підходу середнє значення C_v становить 0.22, тоді як у класичному EDF - 0.48, що свідчить про вищу стабільність і передбачуваність реактивної моделі.

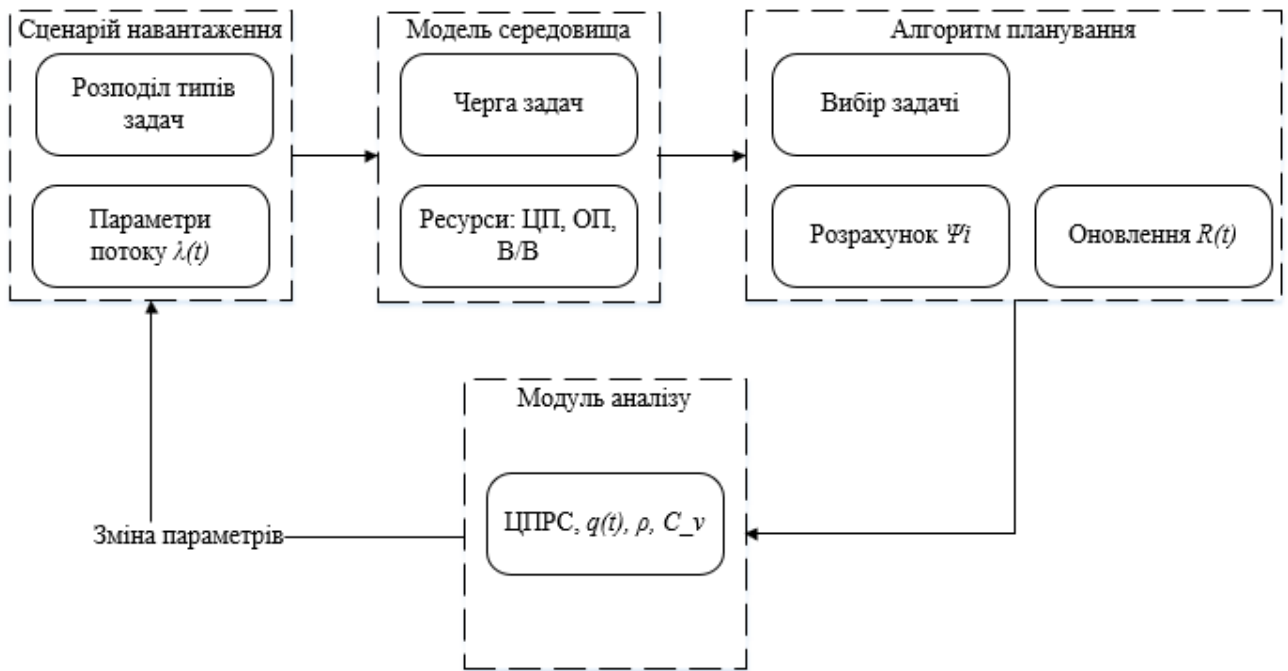


Рис. 3.6. Схема циклу моделювання виконання задач

Отримані результати свідчать, що моделювання сценаріїв дозволяє не лише перевірити математичну коректність алгоритму, але й адаптувати його параметри під реальні умови експлуатації. Завдяки можливості відтворення різних типів навантажень - від рівномірного до пікового - дослідник отримує засіб кількісного оцінювання стійкості, пропускну здатності та відповідності ЦПРС. Це створює основу для подальшого вдосконалення механізмів адаптивного управління.

Таким чином, моделювання сценаріїв виконання кроків бізнес-процесів у середовищі з обмеженими ресурсами підтверджує ефективність розробленого гібридного алгоритму планування та дозволяє кількісно оцінити поведінку системи у межах допустимих режимів. Результати розрахунків демонструють, що система залишається стабільною за виконання умов теорем (3.4)–(3.5), а її показники ЦПРС та середня довжина черги відповідають вимогам реального середовища автоматизації бізнес-сервісів.

3.4. Розробка модифікованого методу “перший прийшов – перший оброблений” з урахуванням багатовимірних ресурсів

У задачах жорсткого реального часу застосовують політики фіксованих пріоритетів, зокрема дедлайн-монотонний підхід (deadline-monotonic) [147], а для періодичних задач аналізуються алгоритмічні властивості та складність відповідних постановок [148]. Водночас, у теорії масового обслуговування дисципліна “першим прийшов - першим оброблений” (first-come-first-served, FCFS) використовується як базова для опису порядку обслуговування в системах черг [123]. У цьому підрозділі розглядається модифікація FCFS, яка перед запуском задачі виконує перевірку допустимості за багатовимірним вектором ресурсів (пам'ять, ЦП, В/В), не змінюючи послідовність надходження задач.

Аналіз сучасних публікацій, присвячених плануванню задач у розподілених, гібридних та ресурсно-обмежених середовищах, свідчить про значну різноманітність методів керування навантаженням і оптимізації використання ресурсів. У джерелі [149] наведено систематизацію методів планування для хмарних та fog-інфраструктур, зокрема розглянуто евристичні, метаевристичні та стохастичні стратегії. Водночас у цьому огляді не висвітлено модифікації FCFS, що передбачають перевірку багатовимірних ресурсних обмежень перед запуском задачі. Основний акцент зроблено на підвищенні ефективності використання обчислювальних ресурсів, тоді як ризики перевантаження системи через нерівномірні ресурсні профілі задач залишаються поза межами розгляду.

У роботі [150] описано сучасні засоби оптимізації навантаження на процесор і механізми розподілу обчислювальних ресурсів у хмарних середовищах. Підходи, наведені в огляді, здебільшого орієнтовані на рівномірне завантаження ЦП та мінімізацію часу виконання. Водночас питання взаємодії між різними типами ресурсів, зокрема оперативною пам'яттю та підсистемою введення-виведення, у цих роботах розглядається обмежено. Як наслідок, алгоритми, представлені в

огляді, не адаптовані до сценаріїв, у яких відсутня еластичність ресурсів і необхідно запобігати ситуаціям, коли задача перевищує доступні обсяги пам'яті або В/В.

У дослідженнях [151; 152] значну увагу приділено підвищенню якості обслуговування та скороченню часу відповіді в умовах динамічного навантаження. Запропоновані підходи базуються на адаптивних критеріях, метаевристиках і оптимізаційних механізмах, які дозволяють підвищити пропускну здатність системи та забезпечити кращу відповідність QoS-вимогам. Проте наведені методи припускають достатність ресурсів або передбачають можливість еластичного розширення ресурсної бази, що зменшує їхню придатність для систем із фіксованими або обмеженими обсягами пам'яті та ВВ. Відсутність механізмів попередньої перевірки ресурсної доступності перед запуском задачі є типовою для досліджень цього класу.

У контексті розподілених обчислювальних середовищ також розглядаються підходи до аналізу й порівняння алгоритмів планування ресурсами, зокрема на прикладі оглядових та прикладних досліджень [153]. Разом із тим, такі роботи не орієнтовані спеціально на модифікації FCFS з попереднім контролем багатовимірних ресурсних обмежень перед запуском задачі.

Окремий клас результатів стосується динамічного планування задач у кластерних середовищах [154] та підходів до зменшення неефективних відображень задач на ресурси за рахунок адаптивних планувальників [155]. У цих постановках акцент робиться на адаптації відображення або параметрів запуску, тоді як питання збереження простої послідовної дисципліни FCFS при одночасному урахуванні багатовимірних ресурсних обмежень, як правило, не є основним об'єктом аналізу.

У роботі [156] розглянуто вплив структурних та апаратних обмежень, зокрема обмежень підсистеми введення-виведення та буферів, на ефективність виконання задач у суперкомп'ютерних середовищах. Дослідження демонструє, що навіть за наявності високопродуктивних обчислювальних вузлів черги можуть переходити у нестійкі режими роботи при нерівномірному В/В-навантаженні. Автори пропонують складні планувальники типу plan-based, які враховують

широкий набір апаратних характеристик, однак такі методи суттєво відрізняються від FCFS і не можуть слугувати його прямою модифікацією.

Інші дослідження зосереджуються на суміжних аспектах планування, зокрема на енергетичній та продуктивнісній оптимізації в хмарних дата-центрах [157], на узагальненні проблем та підходів ресурсного планування у хмарних системах [158], на застосуванні метаевристичних методів у середовищах периферійних обчислень (edge), зокрема використанні ройової оптимізації частинок (particle swarm optimization, PSO) у задачах перенесення (offloading) [159], та на плануванні в гетерогенних обчислювальних середовищах [160]. Хоча кожна з робіт робить внесок у розвиток методів планування, спільною рисою є відсутність механізмів контролю багатовимірних ресурсів у найпростішому, послідовному алгоритмі FCFS. Таким чином, проведений аналіз свідчить про те, що нині недостатньо досліджено питання поєднання традиційного FCFS з ресурсно-орієнтованими умовами допустимості, які враховують обмеженість пам'яті, ЦП та В/В. Це дозволяє окреслити відповідний дослідницький геп і визначити обґрунтованість розробки вдосконаленого алгоритму, який зберігає простоту FCFS, але водночас забезпечує операційну стабільність системи в умовах фіксованих ресурсних бюджетів.

У межах раніше сформульованої ресурсної моделі системи планування задач класичний метод FCFS має низку обмежень, які проявляються особливо виразно в середовищах із жорсткими та неоднорідними ресурсними обмеженнями. FCFS ініціює виконання задач відповідно до порядку їх надходження без попередньої перевірки доступності необхідних ресурсів. У результаті задача з підвищеними ресурсними вимогами може затримувати обробку наступних задач у черзі, а за дефіциту пам'яті - призводити до аварійного завершення програмного середовища виконання. Для інтегрованих середовищ виконання бізнес-процесів із динамічними потоками різномірних задач така поведінка є небажаною.

$$\theta_i = \frac{\|r_i\|}{\|R(t)\|} \leq 1 \quad (3.14)$$

Якщо значення θ_i перевищує одиницю хоча б для одного компонента вектора ресурсів, система відкладає запуск задачі, зберігаючи її у буфері очікування та переходячи до аналізу наступної задачі в черзі. При цьому порядок задач у черзі принципово не змінюється, що зберігає основну властивість FCFS, але дозволяє уникнути перевантаження системи. Після вивільнення ресурсів відкладені задачі повертаються у чергу у первинній послідовності.

Ресурсно-орієнтований FCFS працює у циклічному режимі, характерному для гібридного планування. На кожній ітерації здійснюється оцінка доступних ресурсів.

Після перевірки допустимості задача або запускається, або переноситься у буфер очікування. Запуск задачі не повинен призводити до порушення ресурсних умов (3.4). Якщо ресурсів бракує для всіх задач, що очікують у черзі, алгоритм переходить у режим пасивної стабілізації, у якому запуск нових задач тимчасово призупинено до моменту, поки вектор доступних ресурсів не почне задовольняти обмеження. Такий механізм запобігає нестійкому зростанню черги, що узгоджується з умовою (3.6) стохастичної стійкості та базовими підходами теорії масового обслуговування [123].

Важливою особливістю удосконаленого алгоритму FCFS є відсутність рекомбінації або перестановки задач у черзі. Це принципово відрізняє його від гібридного методу, який, відповідно до виразу (3.9), здійснює ранжування задач за узагальненою функцією придатності. Удосконалений FCFS не використовує вагові коефіцієнти α , β , γ , не застосовує пріоритетів та не враховує дедлайни на етапі вибору задачі. Його мета - не оптимізація порядку запуску, а запобігання ресурсному колапсу при збереженні простої й передбачуваної моделі FCFS. За змістом цей підхід не є політикою фіксованих пріоритетів у жорсткому реальному часі [147] і не відтворює постановки періодичного планування реального часу [148].

Такий підхід забезпечує низку переваг. По-перше, система не допускає запуску задачі, яка потенційно може спричинити блокування пам'яті або надмірне завантаження В/В, що традиційно є найслабкішим місцем FCFS. По-друге,

алгоритм гарантує, що навіть при суттєвих флуктуаціях у потоці задач черга не переходить у режим неконтрольованого зростання. По-третє, зберігається сумісність із середовищами, що не підтримують складні механізми планування платформ автоматизації проектування та виконання бізнес-сервісів. По-четверте, алгоритм забезпечує покращення УРНП, оскільки запуск "легких" задач більше не блокується "важкими", що дає змогу вчасно виконувати більший відсоток операцій.

Вхідні данні:

Q – черга задач S_i ;

r_i – очікуване використання ресурсів задачею S_i ;

$R(t)$ - величина доступних ресурсів у момент t ;

$R_{measured}(t)$ – виміряні доступні ресурси на момент t ;

DeferredBuffer - буфер відкладених задач.

Псевдокод алгоритму ресурсно-орієнтованого "перший прийшов – перший оброблений":

Поки черга Q не порожня, виконувати:

1. Оновити доступні ресурси (без удосконалення/згладжування): $R(t)$
2. Взяти першу задачу S_i із черги Q
3. Обчислити нормоване відношення ресурсної потреби до доступного ресурсу:

$$\theta_i \quad (3.14)$$

4. Якщо $\theta_i \leq 1$, то

- Запустити задачу S_i
- Зменшити доступні ресурси на час виконання задачі $R(t) = R(t) - r_i$
- Дочекатися завершення задачі
- Повернути ресурси після завершення: $R(t) = R(t) + r_i$

Інакше $\theta_i > 1$:

- Перемістити задачу S_i у буфер відкладення *DeferredBuffer*

5. Якщо черга Q порожня і *DeferredBuffer* не порожній, то:

- Повернути всі відкладені задачі з *DeferredBuffer* назад у чергу *Q* зі збереженням початкового порядку.

Кінець циклу.

Запропонований метод може бути інтегрований у платформу інфраструктуру керування бізнес-процесами, де важливо забезпечувати стабільність ресурсного середовища для дотримання ЦПРС. Його застосування сприяє підтриманню передбачуваності виконання кроків бізнес-процесів у змінних умовах навантаження.

3.5. Порівняльні аналітичні оцінки методу планування виконання кроків бізнес-процесів в платформі автоматизації проєктування та виконання бізнес-сервісів

Розглянемо аналітичний експеримент для порівняльної інтерпретації двох методів запуску кроків бізнес-процесів у ресурсно-обмеженому середовищі: гібридного ресурсно-орієнтованого планування та модифікованого підходу “перший прийшов – перший оброблений” із попередньою перевіркою багатовимірної ресурсної допустимості. У межах прийнятої моделі кожне завдання (крок) розглядається як атомарне рішення рівня “запуск/не запуск”, тобто задача або запускається в межах виділених ресурсів, або відхиляється/відкладається відповідно до правил допуску.

Табл. 3.3.

Вхідні налаштування експерименту

Назва	Значення
Дискретний час	$K = 0, 1, 2, \dots, \Delta t = 1\text{с.}$
Тривалість виконання	Для всіх задач $e_i = 1\text{с.}$
Одночасність	На одному кроці можна запустити не більше $P_{max} = 3$ задач

Обмеження пам'яті (токени ОП)	Сума токенів ОП, запущених задач на кроці k не перевищує $M_{max} = 10$.
Обмеження зовнішньої черги	Якщо після надходжень розмір черги $Q_{max} = 10$, то надлишок відхиляються найновіші задачі - це стандартне правило для обмеженої зовнішньої черги.
Інтенсивність надходження задач	5 задач/с
Функціонал штрафів	
	$miss =$ (кількість відхилених + кількість із пізнім стартом)/5

Табл. 3.4.

Вхідні параметри профілів задач

Назва профілю задачі	Токени ОП	Очікувана тривалість, с	Термін очікування початку старту задачі, с
J1	2	1	1
J2	3	1	2
J3	4	1	1
J4	5	1	1
J5	3	1	1
J6	1	1	1
J7	4	1	1
J8	2	1	3
J9	1	1	2
J10	5	1	1
J11	7	1	3
J12	8	1	2

J13	2	1	1
J14	4	1	1
J15	7	1	1
J16	9	1	1
J17	1	1	3
J18	3	1	4
J19	3	1	1
J20	3	1	1

Результати (Табл. 3.5) аналітичного моделювання гібридного алгоритму за функцією придатності ψ із $\alpha = \gamma = 0.5$, $\beta = 0$, враховується лише ОП (токени пам'яті), тобто $\frac{\|r_i\|}{\|R_k\|} \rightarrow \frac{m_i}{R_k}$, $D_i = a_i + d_i$ абсолютний дедлайн старту, щоб уникнути ділення на нуль у дискретному часі, використано $arg_1 = \frac{1}{\max(1, D_i - k)}$, $arg_2 = 1 - \frac{m_i}{R_k}$, тоді $\psi_i = \alpha \cdot arg_1 + \gamma \cdot arg_2$;
 a_i - крок надходження задачі;
 R_k - залишок доступних токенів ОП перед запуском відповідної задачі в межах кроку k ;
 m_k - сума вже використаних токенів ОП на кроці k після запуску цієї задачі (тобто накопичено в межах кроку).

Табл. 3.5.

Результати моделювання гібридного методу

k	Старт	Токени ОП	m_k	R_k	$miss$	arg_1	arg_2	ψ_i
0	J1	2	2	10	0	1	0.8	0.9
0	J5	3	5	8	0	1	0.625	0.812
0	J3	4	9	5	0	1	0.2	0.6
1	J6	1	1	10	0	1	0.9	0.95
1	J2	3	4	9	0	1	0.667	0.833
1	J9	1	5	6	0	0.5	0.833	0.667
2	J13	2	2	10	0	1	0.8	0.9
2	J7	4	6	8	0	1	0.5	0.75
2	J8	2	8	4	0	0.5	0.5	0.5
3	J19	3	3	10	0.2	1	0.7	0.85

3	J14	4	7	7	0.2	1	0.429	0.714
3	J17	1	8	3	0.2	0.5	0.667	0.583
4	J4	5	5	10	0.4	1	0.5	0.75
4	J10	5	10	5	0.4	1	0	0.5
5	J11	7	7	10	0	1	0.3	0.65
5	J18	3	10	3	0	0.5	0	0.25
6	J15	7	7	10	0.2	1	0.3	0.65
7	J12	8	8	10	0.2	1	0.2	0.6
8	J16	9	9	10	0.2	1	0.1	0.55

Частка порушення дедлайнів

пізній старт: $5/20 = 0.25$ (25%)

(пізній старт + відхилені): $(5 + 1)/20 = 0.30$ (30%)

Табл. 3.6.

Результати моделювання модифікованого методу “перший прийшов – перший оброблений”

k	Старт (профіль)	Токени ОП, m_i	m_k (після старту)	R_k (перед стартом)	$miss$
0	J1	2	2	10	0.0
0	J2	3	5	8	0.0
0	J3	4	9	5	0.0
1	J4	5	5	10	0.0
1	J5	3	8	5	0.0
1	J6	1	9	2	0.0
2	J7	4	4	10	0.0
2	J8	2	6	6	0.0
2	J9	1	7	4	0.0
3	J10	5	5	10	0.4
3	J13	2	7	5	0.4
3	J17	1	8	3	0.4
4	J18	3	3	10	0.0
4	J19	3	6	7	0.0
5	J11	7	7	10	0.0
6	J12	8	8	10	0.2
7	J14	4	4	10	0.2
8	J15	7	7	10	0.2
9	J16	9	9	10	0.2

Частка порушення дедлайнів

пізній старт: 5 задач (J10, J12, J14, J15, J16) із 20, $5/20 = 0.25$ (25%)

(пізній старт + відхилені): $(5 + 1)/20 = 0.30$ (30%)

Однаковий результат за часткою порушень у цьому експерименті пояснюється тим, що під заданими обмеженнями “вузьке місце” визначається не стільки правилом пріоритизації, скільки пропускнуою здатністю системи та жорстким допуском за ресурсами. За кроку $\Delta t = 1\text{с}$ і максимальної одночасності $P_{\max}=3$ система фізично не може запускати більше трьох задач за секунду, тоді як надходить 5 задач/с; отже, накопичення черги й неминучі ізні старти виникають незалежно від того, як саме впорядковуються задачі. Додатково ліміт токенів оперативної пам’яті $M_{\max}=10$ обмежує допустимі комбінації задач на кроці, тому реальний набір “можна запустити зараз” часто майже однаковий для обох методів.

Висновки до розділу 3

Обґрунтовано методи планування й управління виконанням кроків бізнес-процесів у платформі автоматизації проєктування та виконання бізнес-сервісів, що працює в умовах обмежених і неоднорідних ресурсів. У розділі застосовано моніторингову модель і контур “оцінювання ЦПРС - реагування” як основу для фіксації надходження/запуску кроків і контролю порушень дедлайнів старту.

На базі формальної ресурсної моделі (вектори квот, профілі задач і дедлайни старту) сформульовано критерії коректності: локальну ресурсну допустимість запуску та умову стохастичної стійкості потоку задач, що визначають межі стабільної роботи системи.

Розроблено гібридний ресурсно-орієнтований алгоритм планування на основі механізму ресурсних токенів і функції придатності, який поєднує дедлайнові та реактивні рішення з урахуванням часової, пріоритетної та ресурсної складових.

Перевірку ефективності виконано через дискретно-подієве моделювання за метриками частки своєчасно запущених задач (ЦПРС/УРНП-задоволеність) і середньої довжини черги в різних сценаріях навантаження; додатково оцінено стабільність за коефіцієнтом варіації довжини черги, який для гібридного алгоритму становить 0.22 проти 0.48 для класичного EDF.

Запропоновано модифікацію “перший прийшов – перший оброблений” із попередньою перевіркою багатовимірної ресурсної допустимості (пам’ять, ЦП, введення-виведення) без зміни порядку надходження задач, із буфером відкладених задач і поверненням у чергу в первинній послідовності.

У порівняльному аналітичному експерименті (одновимірне обмеження за токенами оперативної пам’яті) обидва методи дали однакову частку порушень: пізній старт 25% та (пізній старт + відхилені) 30%. Це узгоджується з тим, що за фіксованих жорстких лімітів паралельності та оперативної пам’яті ключовий вплив на результат має саме допустимість запусків у такті, а не правило впорядкування в межах допустимого набору.

РОЗДІЛ 4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ УПРАВЛІННЯ ТА ПЕРЕВІРКИ СЕРЕДОВИЩА ПРОЄКТУВАННЯ ТА ВИКОНАННЯ БІЗНЕС-СЕРВІСІВ

4.1. Розробка спеціалізованого фреймворку для дослідження параметрів та способів контролю за виконанням бізнес-процесів в платформах проєктування та виконання бізнес-сервісів

Розробка спеціалізованого фреймворку спрямована на створення уніфікованого середовища для дослідження процесів керування ресурсами та аналізу ефективності виконання бізнес-процесів у гібридних ПНОП. Запропонований підхід базується на концепції адаптивного контролю процесів у середовищах, де ресурси (ЦП, ОП) розподіляються між динамічними наборами задач. Фреймворк дозволяє формалізувати взаємодію між компонентами системи, провести інструментальні експерименти з політиками розподілу задач і забезпечує наскрізний моніторинг продуктивності.

Основою створеної системи є модель інтеграції рівнів проєктування, виконання та моніторингу, що реалізує зворотний зв'язок між параметрами ресурсного стану середовища й механізмами диспетчеризації задач. Вона включає три основні етапи функціонування: ініціалізацію тестового середовища, генерацію процесних моделей та аналіз результатів виконання. На кожному з етапів фреймворк забезпечує можливість автоматичного збору метрик використання ресурсів, визначення відхилень від очікуваних параметрів і побудову профілів навантаження для подальшого аналізу.

Інформаційна технологія, що реалізована у межах фреймворку, передбачає автоматизований контроль стану ресурсів при виконанні бізнес-процесів. На вході система приймає опис бізнес-моделей, конфігурації середовища, параметри генерації задач і політики диспетчеризації. На виході формується набір показників, що характеризують якість планування, частка відхилень від дедлайнів та частки

використання ресурсів. Взаємодія між модулями системи здійснюється через подієву шину з використанням внутрішнього API.

Фреймворк реалізує модульну архітектуру, де кожен модуль відповідає за окремий аспект функціонування системи. На рис. 4.1 представлено загальну схему інформаційної технології перевірки використання ресурсів при виконанні бізнес-процесів у платформі проектування та виконання бізнес-сервісів. Технологія реалізує замкнений цикл керування, який поєднує етапи ініціалізації середовища, диспетчеризації процесів, моніторингу ресурсів та аналізу результатів. Вона забезпечує автоматизований збір і оцінку показників ефективності використання обчислювальних ресурсів (ЦП, ОП) під час виконання задач різних типів навантаження в умовах ПНОП.

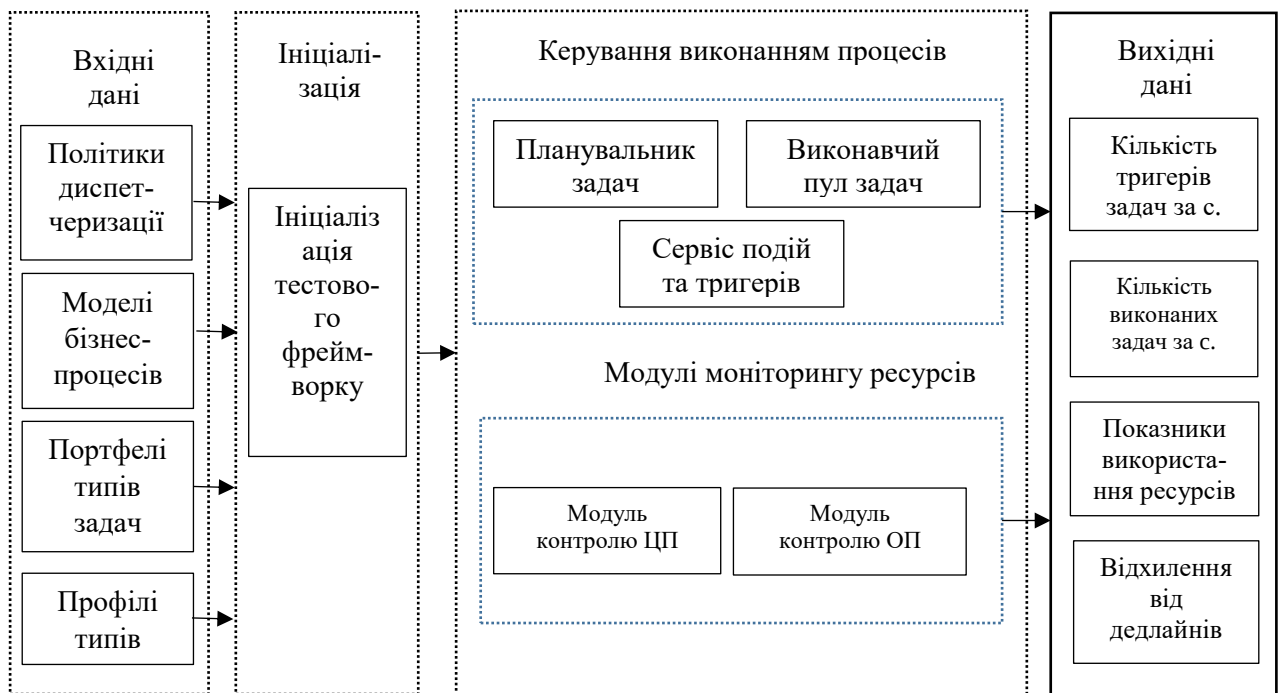


Рис. 4.1. Схема інформаційної технології перевірки використання ресурсів при виконанні бізнес-процесів у платформі проектування та виконання бізнес-сервісів

Блок вхідних даних визначає параметри експериментального середовища, включаючи політики диспетчеризації, моделі бізнес-процесів, портфелі типів задач і профілі типів, що задають початкові умови моделювання. На етапі ініціалізації

здійснюється розгортання вузлів середовища та налаштування тестового фреймворку для забезпечення відтворюваності експериментів.

Блок керування виконанням процесів містить планувальник задач, виконавчий пул і сервіс подій, які спільно формують динамічний механізм диспетчеризації та синхронізації задач у реальному часі. Паралельно працює блок моніторингу ресурсів, що охоплює модулі контролю ЦП та ОП і забезпечує збір телеметрії для оцінки стану системи.

На виході формується набір результуючих показників, які відображають інтенсивність активації та виконання задач, ефективність використання ресурсів і відхилення від дедлайнів. Отримані дані використовуються для порівняння алгоритмів планування, верифікації політик диспетчеризації та побудови рекомендацій щодо оптимізації використання ресурсів у середовищі виконання бізнес-сервісів.

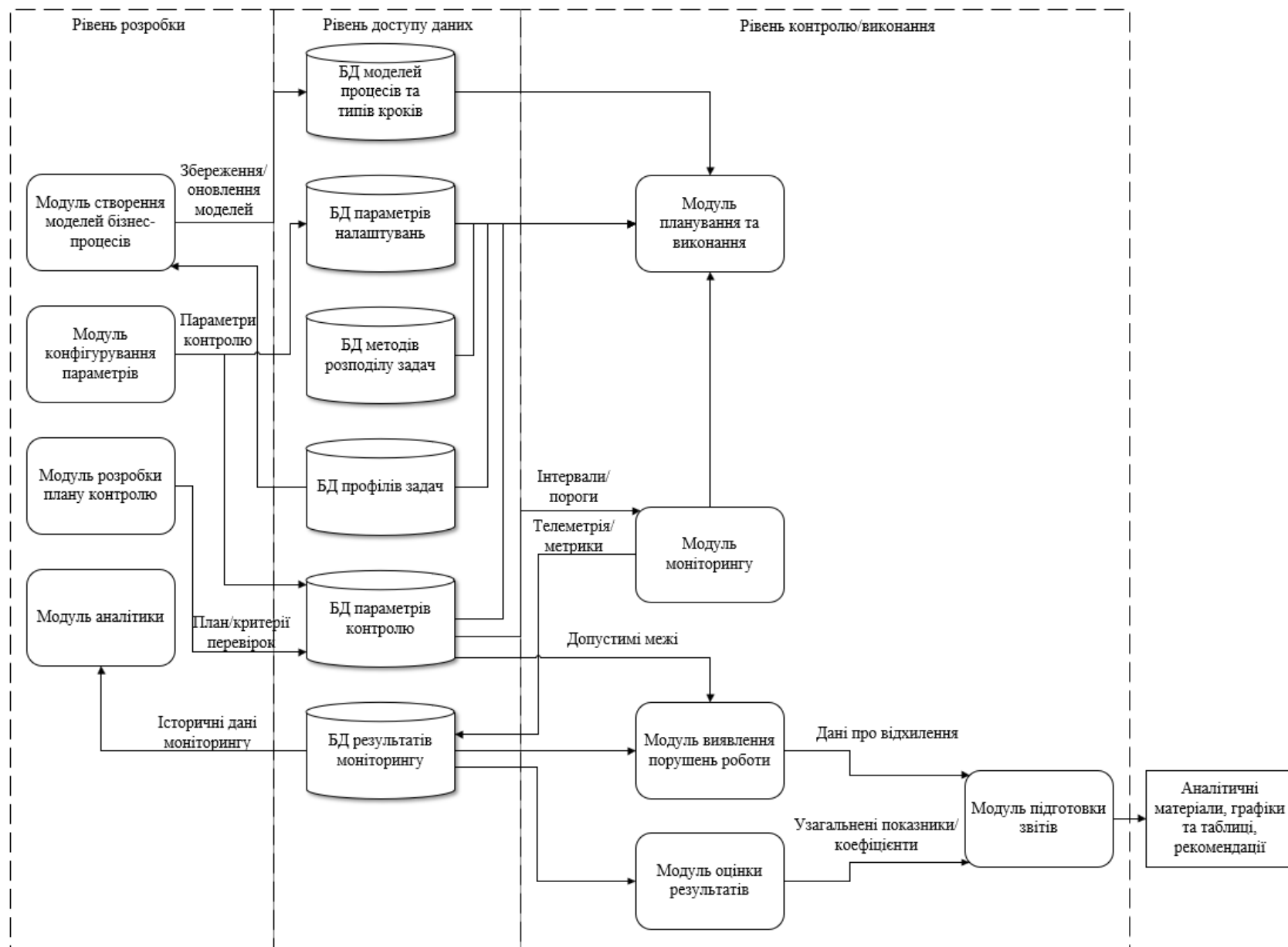


Рис. 4.2. Функціональна схема інформаційної системи контрольованого виконання бізнес-процесів у серовищі проєктування та виконання бізнес-сервісів.

Наступним етапом є формування функціональної схеми інформаційної системи контрольованого виконання бізнес-процесів у середовищі проектування та виконання бізнес-сервісів. Запропонована схема відображає узгоджену взаємодію компонентів фреймворку та організована у вигляді трьох логічних рівнів: рівня розробки, рівня доступу даних та рівня контролю/виконання, які забезпечують наскрізний цикл забезпечення контрольованості — від підготовки моделей і параметрів до збору телеметрії, виявлення відхилень, оцінювання результатів і формування звітних матеріалів. Рівень розробки відповідає за створення та актуалізацію артефактів, які визначають правила та умови контрольованого виконання: модуль створення моделей бізнес-процесів формує моделі та ініціює їх збереження/оновлення у базі моделей процесів і типів кроків; модуль конфігурування параметрів задає параметри налаштувань і параметри контролю, що записуються відповідно в базу параметрів налаштувань та базу параметрів контролю; модуль розробки плану контролю визначає план і критерії перевірок та передає їх до бази параметрів контролю; модуль аналітики використовує історичні дані моніторингу, отримані з бази результатів моніторингу, для узагальнення попередніх спостережень і підтримки прийняття рішень на етапі підготовки та уточнення параметрів контролю. Рівень доступу даних реалізує функції централізованого зберігання та обміну інформацією між усіма модулями і включає шість ключових сховищ: базу моделей процесів та типів кроків, базу параметрів налаштувань, базу методів розподілу задач, базу профілів задач, базу параметрів контролю та базу результатів моніторингу; ці сховища формують єдиний інформаційний простір, у якому з одного боку накопичуються й підтримуються актуальними вхідні дані для виконання та контролю, а з іншого — зберігаються результати функціонування для подальшого аналізу. Рівень контролю/виконання є операційним рівнем і забезпечує безпосередню реалізацію контрольованого виконання бізнес-процесів: модуль планування та виконання отримує з рівня даних моделі процесів і типів кроків, параметри налаштувань, методи розподілу задач та профілі задач і на їх основі формує порядок виконання та запускає виконання відповідно до обраних правил; модуль моніторингу працює за інтервалами та

порогами, визначеними в базі параметрів контролю, безперервно збирає телеметрію та метрики під час виконання, і передає їх у базу результатів моніторингу; модуль виявлення порушень роботи використовує дані з бази результатів моніторингу та зіставляє їх із допустимими межами, заданими в базі параметрів контролю, формуючи дані про відхилення; модуль оцінки результатів отримує накопичені результати моніторингу та виконує їх узагальнення, формуючи показники та коефіцієнти, які характеризують перебіг контрольованого виконання; далі модуль підготовки звітів агрегує дані про відхилення та узагальнені показники/коефіцієнти і формує вихідні матеріали у вигляді аналітичних матеріалів, графіків, таблиць та рекомендацій. Таким чином, функціональна схема забезпечує формалізовану взаємодію між підсистемами розробки, централізованого доступу до даних та операційного контролю/виконання, а інформаційні потоки "збереження/оновлення моделей", "параметри контролю", "план/критерії перевірок", "інтервали/пороги", "телеметрія/метрики", "допустимі межі", "дані про відхилення" та "узагальнені показники/коефіцієнти" забезпечують повний цикл отримання, накопичення, інтерпретації та представлення результатів контрольованого виконання бізнес-процесів.

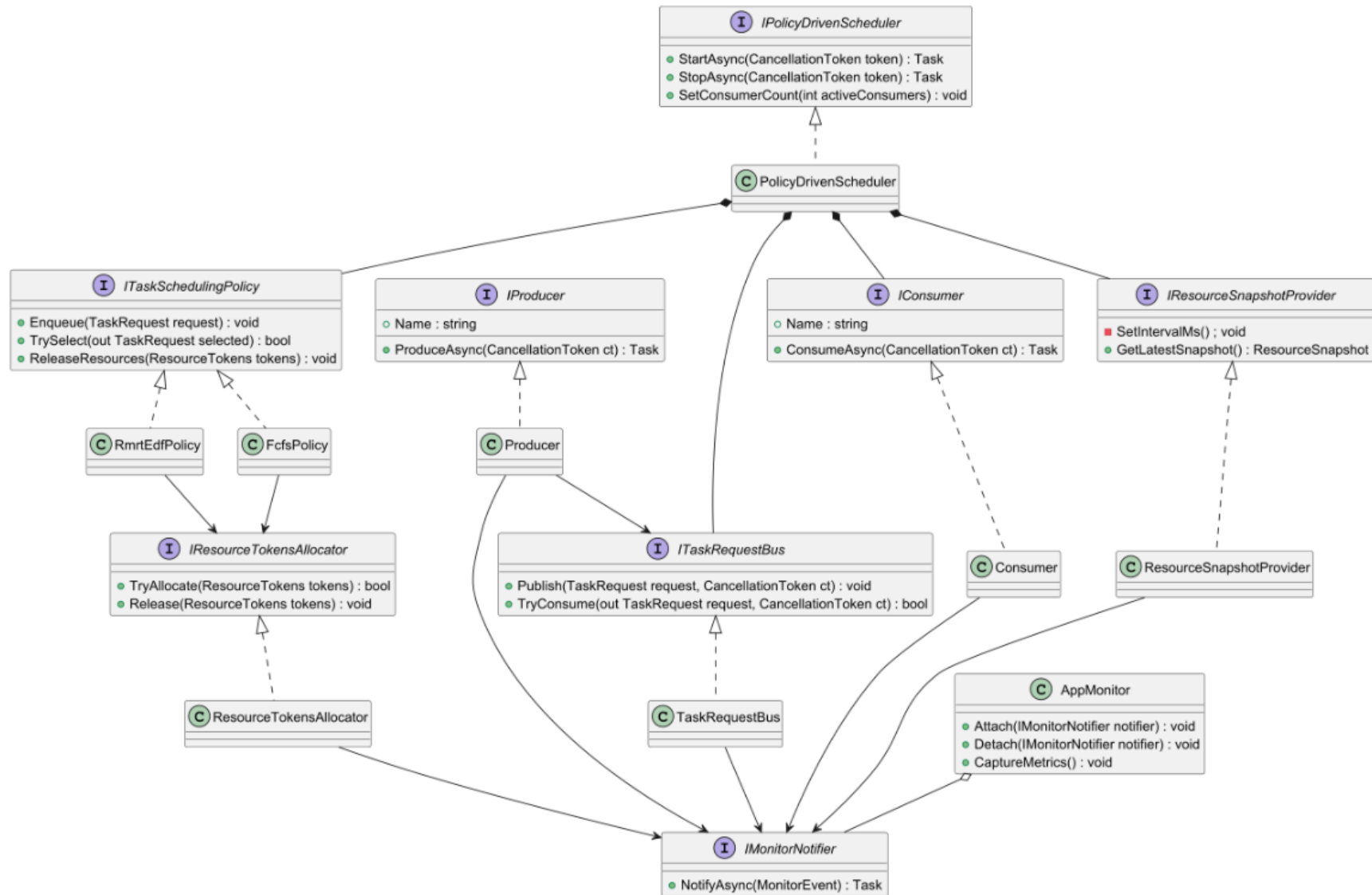


Рис. 4.3. Діаграма класів розробленого фреймворку ресурсно-орієнтованого планування та моніторингу виконання кроків бізнес-процесів

На діаграмі (Рис. 4.3.) зображено структуру системи планування та керування виконанням задач, побудованої на основі принципу політик керування та інверсії залежностей. Структура системи орієнтована на розподіл відповідальностей між окремими компонентами, які взаємодіють через чітко визначені інтерфейси, що забезпечує високу гнучкість, масштабованість і можливість динамічного адаптування до змін умов виконання.

Центральним елементом системи є клас `PolicyDrivenScheduler`, який реалізує інтерфейс `IPolicyDrivenScheduler`. Цей компонент виконує функції керування процесом планування та виконання задач, забезпечуючи асинхронний запуск (`StartAsync`), зупинку (`StopAsync`) і регулювання кількості активних споживачів (`SetConsumerCount`). `PolicyDrivenScheduler` виступає як координаційний центр, який приймає рішення на основі заданої політики планування, поточного стану ресурсів і моніторингових даних. Такий підхід реалізує принцип керування через політики, де логіка прийняття рішень визначається не жорстко закодованими правилами, а правилами. Визначеними в політиках розподілу задач, котрі можуть бути реалізовані та підключені до системи.

Інтерфейс `ITaskSchedulingPolicy` визначає базові механізми управління чергою задач: додавання нових запитів (`Enqueue`), вибір наступного запиту для виконання (`TrySelect`) та звільнення використаних ресурсів (`ReleaseResources`). Його реалізують конкретні політики - `RmrtEdfPolicy` на основі запропонованого алгоритму в розділі 2, та модифікованого `FcfsPolicy` (`First Come First Serve`, що реалізує класичний підхід “першим прийшов - першим обслуговується”). Обидві політики взаємодіють із `IResourceTokensAllocator`, який відповідає за виділення (`TryAllocate`) та звільнення (`Release`) обчислювальних або інших системних ресурсів. Його конкретна реалізація - `ResourceTokensAllocator` — виконує функції диспетчера ресурсів у межах системи керування.

Компоненти `IProducer` і `IConsumer` описують відповідно продуцентів і споживачів задач, які реалізують окремі частини керованого процесу. Кожен із них має власне ім'я та методи для асинхронної генерації (`ProduceAsync`) і споживання (`ConsumeAsync`) задач. Реалізації `Producer` і `Consumer` обмінюються даними через

шину запитів, представлену інтерфейсом `ITaskRequestBus`, що відповідає за публікацію (`Publish`) і вибірку (`TryConsume`) запитів. Клас `TaskRequestBus` виступає центральним комунікаційним каналом системи, забезпечуючи узгоджене керування потоками запитів.

Для забезпечення зворотного зв'язку в системі керування застосовується інтерфейс `IMonitorNotifier`, який через метод `NotifyAsync` інформує про події моніторингу або зміну стану системи. Його реалізація використовується у класі `AppMonitor`, який надає засоби для підключення або відключення моніторів (`Attach`, `Detach`) та збору системних метрик (`CaptureMetrics`). Інтерфейс `IResourceSnapshotProvider` забезпечує періодичне отримання знімків стану ресурсів (`GetLatestSnapshot`) із заданим інтервалом (`SetIntervalMs`). Клас `ResourceSnapshotProvider` реалізує цю функцію, забезпечуючи актуальні дані для прийняття керуючих рішень у рамках зворотного контуру.

Таким чином, представлена діаграма реалізує замкнутий контур керування, у якому `PolicyDrivenScheduler` виступає керуючим елементом, політики планування - модулем прийняття рішень, а компоненти моніторингу - сенсорним підсистемами, що надають інформацію про стан системи.

Фреймворк реалізовано з допомогою мови програмування `C#` у середовищі `JetBrains Rider` з використанням `.NET 8`. Результати логуються в CSV-файли, які згодом піддаються статистичному аналізу у середовищі `Python`. Для кожного експерименту формується окрема сесія із зафіксованими параметрами: кількість задач, тип навантаження (ЦП-, ОП- інтенсивні) та тривалістю `T` с. Така уніфікація дозволяє забезпечити відтворюваність експериментів та порівнянність результатів.

4.2. Графічний інтерфейс розробленого спеціалізованого середовища системи управління та перевірки

Графічний інтерфейс користувача (ГІК) розробленого програмного забезпечення є важливим компонентом, який забезпечує взаємодію користувача з

усіма підсистемами фреймворку - від налаштування параметрів симуляції до моніторингу виконання бізнес-процесів у режимі реального часу. Його основне призначення полягає у створенні єдиного інтерактивного середовища для конфігурування, моделювання, керування ресурсами та аналітики результатів експериментів.

Інтерфейс побудовано за принципом модульної організації, де кожна вкладка відображає окрему підсистему платформи: загальні налаштування; генерація моделей бізнес-процесів; редактор моделей; виконання.

Кожна вкладка виконує функцію логічного інтерфейсного модуля, який взаємодіє з ядром системи через REST API. Візуальні елементи реалізовано засобами Blazor Server Components із використанням архітектури Model–View–Controller (MVC), що дозволяє відокремити представлення даних від логіки обробки подій. Завдяки цьому платформа має кросплатформну сумісність і може розгортатися як локально, так і в контейнеризованому середовищі (Docker/Kubernetes).

[Налаштування середовища](#)
[Генерація моделей процесів](#)
[Редагування моделей](#)
[Виконання](#)

Налаштування середовища

Метод та обмеження ресурсів

Метод розподілу завдань

Програмне обмеження оперативної пам'яті (МБ)

Удосконалений метод "перший прийшов-перши

550

Кількість виробників та споживачів

Мінімальна кількість виробників

Максимальна кількість виробників

Мінімальна кількість споживачів

Максимальна кількість споживачів

1

4

2

8

Часові параметри

Граничний термін очікування старту кроку за замовчуванням (с)

Інтервал опитування стану ресурсів (мс)

1

500

Застосувати

Відновити налаштування за замовчуванням

Рис. 4.4. Вкладка “Налаштування середовища” платформи проектування та моніторингу виконання бізнес-процесів

На вкладці (Рис. 4.4) Налаштування середовища користувач задає основні параметри симуляційного середовища, які визначають поведінку системи під час тестування алгоритмів планування задач. До таких параметрів належать:

Політика розподілу завдань - політика диспетчеризації: удосконалений метод “перший прийшов – перший оброблений”, метод планування задач із дедлайн орієнтованим-впорядкуванням;

Програмне обмеження оперативної пам'яті (МБ) - порогове значення використання оперативної пам'яті, після якого активується механізм обмеження задач;

Мінімальна та максимальна кількість виробників/споживачів - межі кількості генераторів та обробників;

Граничний термін очікування старту кроку за замовчуванням (с) - базовий час дедлайну задач;

Інтервал опитування стану ресурсів (мс) - інтервал опитування ресурсів системи.

Кнопки “Застосувати” та “Відновити налаштування за замовчуванням” дозволяють відповідно зберігати поточні налаштування або повернути їх до стандартних. Дані вводяться у типізованих формах із валідацією вхідних значень.

Генерація моделей бізнес-процесів

Назва профілю бізнес-сервісу	Префікс назви моделі	Кількість моделей
	БПМодель	10
Кількість кроків	Частка кроків із ЦП (0–1)	Частка кроків із ОП (0–1)
10000	0.4	0.6
Мінімальна кількість кроків	Максимальна кількість кроків	Пам'ять для блокування (МБ)
3	10	128
☐ Спочатку завдання з ОП		
Згенерувати моделі		

Реєстр профілів бізнес-сервісів

Назва профілю бізнес-сервісу	Кількість моделей	Загальна кількість задач	Параметри	
Сервіс обробки з профілем 80/20 (ЦП/ОП)	670	5000	--cpuRatio 0.8 --memRatio 0.2 --min 3 --max 10 --blockMemMb 128	Видалити
Сервіс обробки з профілем 50/50 (ЦП/ОП)	700	5000	--cpuRatio 0.5 --memRatio 0.5 --min 3 --max 10 --blockMemMb 128	Видалити

Рис. 4.5. Вкладка Business Process Generation платформи проектування та моніторингу виконання бізнес-сервісів

Вкладка (Рис. 4.5) призначена для налаштування параметрів і запуску формування набору моделей, а також для перегляду й керування створеними профілями бізнес-сервісів. Інтерфейс складається з двох основних частин: форми генерації у верхній частині сторінки та таблиці реєстру в нижній. У формі користувач задає назву профілю бізнес-сервісу, префікс назви моделі та кількість моделей, що мають бути згенеровані. Далі вказується кількість кроків, частка кроків із процесорною операцією (ЦП) та частка кроків з операцією з пам'яттю (ОП), які задаються у межах 0–1. Для контролю варіативності структури встановлюються мінімальна та максимальна кількість кроків у моделі. Поле “Пам'ять для блокування (МБ)” використовується для задання обсягу пам'яті, пов'язаного з блокуванням, а прапорець “Спочатку завдання з ОП” задає початковий порядок формування кроків з операцією з пам'яттю. Після заповнення параметрів натискання кнопки “Згенерувати моделі” запускає формування моделей. Нижче подано розділ “Реєстр профілів бізнес-сервісів” у вигляді таблиці, де для кожного запису показано назву профілю, кількість моделей, загальну

кількість задач і рядок параметрів у компактному вигляді, а також кнопку “Видалити” для видалення відповідного профілю з реєстру. У таблиці кожен рядок відповідає окремому профілю; у стовпці “Параметри” відображаються значення часток ЦП/ОП, меж кроків і пам’яті для блокування, що узгоджуються з налаштуваннями генерації. Кнопка “Видалити” розташована праворуч і дає змогу вилучити профіль.

Редагування та налаштування

Виберіть профіль бізнес-сервісу

Сервіс обробки з профілем 80/20 (ЦП/ОП)

Виберіть модель бізнес-процесу

БПМодель_1

Кроки в моделі бізнес-процесу: БПМодель_1

№ позиції	Назва	Тип	Тривалість (мс)		
1	ЦП_Крок_1	Крок з використанням ЦП	800	Редагувати	Видалити
2	ОП_Крок_1	Крок з використанням ОП	1200	Редагувати	Видалити
3	ЦП_Крок_2	Крок з використанням ЦП	300	Редагувати	Видалити
4	ЦП_Крок_3	Крок з використанням ЦП	250	Редагувати	Видалити

Додати крок

Крок з використанням ЦП

Вставити в позицію (0–N)

0

Додати крок

Рис. 4.6. Вкладка “Редагування моделей” платформи проектування та моніторингу виконання бізнес-сервісів

Вкладка (Рис. 4.6) призначена для вибору профілю бізнес-сервісу, вибору конкретної моделі бізнес-процесу та перегляду і коригування її кроків. У верхній частині розміщено два списки вибору: “Виберіть профіль бізнес-сервісу” та “Виберіть модель бізнес-процесу”, які задають контекст редагування. Нижче відображається розділ “Кроки в моделі бізнес-процесу”, де наведено таблицю кроків обраної моделі із зазначенням номера позиції, назви кроку, типу кроку та тривалості в мілісекундах. Для кожного кроку доступні дії “Редагувати” та “Видалити”, що дозволяють змінювати параметри або вилучати елемент зі

структури моделі. Під таблицею розміщено блок "Додати крок", у якому вибирається тип нового кроку та задається позиція вставлення ($0-N$), після чого додавання виконується кнопкою "Додати крок". У нижній частині сторінки відображається форма "Налаштування вибраного кроку", яка показує поточний крок та містить поля для його редагування: назву, тип, очікувану тривалість (мс), очікуване використання оперативної пам'яті (байт) і коефіцієнт завантаження процесора ($0-1$). Збереження змін виконується кнопкою "Зберегти". Таким чином, вкладка забезпечує редагування складу та параметрів кроків моделі у межах вибраного профілю і моделі.

Кроки в моделі бізнес-процесу: БПМодель_1

№ позиції	Назва	Тип	Тривалість (мс)	
1	ЦП_Крок_1	Крок з використанням ЦП	800	Редагувати Видалити
2	ОП_Крок_1	Крок з використанням ОП	1200	Редагувати Видалити
3	ЦП_Крок_2	Крок з використанням ЦП	300	Редагувати Видалити
4	ЦП_Крок_3	Крок з використанням ЦП	250	Редагувати Видалити

Додати крок

Крок з використанням ЦП

Вставити в позицію (0-N)

0

Додати крок

Налаштування вибраного кроку: ОП_Крок_1

Назва

ОП_Крок_1

Тип

Крок з використанням ОП

Очікувана тривалість (мс)

1200

Очікуване використання ОП (Байт)

5483520

Коефіцієнт завантаження ЦП (0-1)

0

Зберегти

Рис. 4.7. Вкладка "Редагування моделей" платформи проектування та моніторингу виконання бізнес-сервісів у режимі редагування кроку

На вкладці відображено структуру вибраної моделі бізнес-процесу "БПМодель_1". У таблиці подано кроки з номером позиції, назвою, типом і тривалістю в мілісекундах; для кожного рядка доступні дії "Редагувати" та "Видалити". Нижче розміщено блок "Додати крок", де вибирається тип кроку,

задається позиція вставлення ($0-N$) і виконується додавання кнопкою "Додати крок". У секції "Налаштування вибраного кроку" показано параметри кроку ОП_Крок_1: назву, тип, очікувану тривалість, очікуване використання оперативної пам'яті (байт) та коефіцієнт завантаження ЦП ($0-1$), із збереженням кнопкою "Зберегти". Заголовок таблиці містить назву моделі, а значення тривалості подано числом. Поля редагування представлені як текстові та числові введення, що дозволяють змінювати вибраний крок за потреби.

[Налаштування середовища](#)
[Генерація моделей процесів](#)
[Редагування моделей](#)
[Виконання](#)

Виконання

Вибрати профіль бізнес-сервісу Сервіс обробки з профілем 80/20 (ЦП/ОП) ▼

Загальна інформація

Кількість моделей бізнес-процесів: **670**

Кількість кроків: **5000**

Кроки з використанням ЦП: **79.8 %** | Кроки з використанням ОП: **20.2 %**

Запустити тест
Відмінити тест

Лог виконання

```
[16:48:10] [INFO] Запити на виконання задач опубліковано: 10% (500/5000) задач опубліковано
[16:48:11] [INFO] Запити на виконання задач оброблено: 10% (500/5000) задач завершено
[16:49:59] [INFO] Запити на виконання задач опубліковано: 20% (1000/5000) задач опубліковано
[16:50:00] [INFO] Запити на виконання задач оброблено: 20% (1000/5000) задач завершено
[16:51:48] [INFO] Запити на виконання задач опубліковано: 30% (1500/5000) задач опубліковано
[16:51:49] [INFO] Запити на виконання задач оброблено: 30% (1500/5000) задач завершено
```

Рис. 4.8. Вкладка “Виконання” платформи проєктування та моніторингу виконання бізнес-сервісів.

Вкладка (Рис. 4.8.) призначена для запуску та контролю тестового виконання вибраного профілю бізнес-сервісу, а також для перегляду поточного стану виконання у вигляді журналу. У верхній частині сторінки розміщено елемент вибору "Вибрати профіль бізнес-сервісу", за допомогою якого задається профіль, для якого проводиться тест. Нижче відображається інформаційний блок "Загальна

інформація", що містить узагальнені характеристики обраного профілю: кількість моделей бізнес-процесів, кількість кроків і частки кроків із використанням процесора (ЦП) та оперативної пам'яті (ОП), подані у відсотках. Під блоком загальної інформації розміщено кнопки керування тестом: "Запустити тест" для ініціювання виконання та "Відмінити тест" для припинення виконання. Нижче подано розділ "Лог виконання", у якому відображається текстовий журнал повідомлень. Записи журналу містять часові позначки та рівень повідомлення, після чого наведено короткий опис стану обробки. У журналі фіксуються відомості про прогрес, зокрема частка виконання у відсотках та співвідношення значень у форматі "виконано/загалом" для опублікованих і завершених задач. Відображення логів забезпечує спостереження за перебігом тесту у реальному часі та дає змогу оцінити динаміку публікації та завершення задач у межах заданого профілю.

Таким чином, розроблений графічний інтерфейс забезпечує налаштування середовища, генерацію та редагування моделей бізнес-процесів, а також кероване виконання тестів із відображенням узагальнених показників і журналу подій. Це підтримує послідовне проведення експериментів і спостереження за перебігом обробки задач у структурованій формі.

4.3. Система управління та перевірки середовища проєктування та виконання бізнес-сервісів

4.3.1. Розробка системи управління в середовищі проєктування та виконання бізнес-сервісів

Система управління у середовищі проєктування та виконання бізнес-сервісів являє собою багаторівневу інтегровану архітектуру, призначену для забезпечення узгодженого функціонування підсистем, що реалізують повний життєвий цикл бізнес-процесів - від моделювання й планування до виконання, моніторингу, аналітики та збереження даних. Така система виконує роль центрального керуючого контуру, у межах якого реалізуються механізми стратегічного

планування, ресурсного балансування, контролю ЦПРС-показників і підтримання стабільності всієї обчислювальної інфраструктури. Її функціонування базується на принципах сервісно-орієнтованої та подієво-орієнтованої архітектур, що забезпечують масштабованість, стійкість і адаптивність до змін зовнішнього середовища.

Система управління (Рис. 4.7) охоплює чотири основні підсистеми: управління та перевірки ресурсів і ЦПРС, обробки та підготовки даних, інтеграції, а також зберігання та отримання даних. Кожна з них виконує власну функціональну роль, але разом вони утворюють цілісний керуючий простір, що підтримує циклічний обмін інформацією, ресурсами та подіями між усіма рівнями архітектури. Таке поєднання забезпечує формування замкненого контурного керування, у межах якого відбувається збір метрик, аналіз стану системи, прийняття рішень та реалізація коригувальних дій.

Основою побудови системи управління є принцип модульної декомпозиції, який дозволяє розподіляти складні функції між спеціалізованими компонентами. Зокрема, підсистема управління та перевірки ресурсів і ЦПРС містить модулі диспетчеризації завдань, моніторингу ресурсів, виявлення некоректної роботи та формування звітності. Підсистема обробки й підготовки даних включає інструменти редагування моделей процесів, параметризації системи, генерації тригерів і збору метрик. Підсистема інтеграції реалізує канали комунікації між усіма частинами комплексу, а підсистема зберігання забезпечує надійне функціонування репозиторіїв моделей, профілів завдань і результатів моніторингу. Така структурна композиція забезпечує розподіл обов'язків і гнучкість конфігурації, що особливо важливо для динамічних виробничих середовищ.

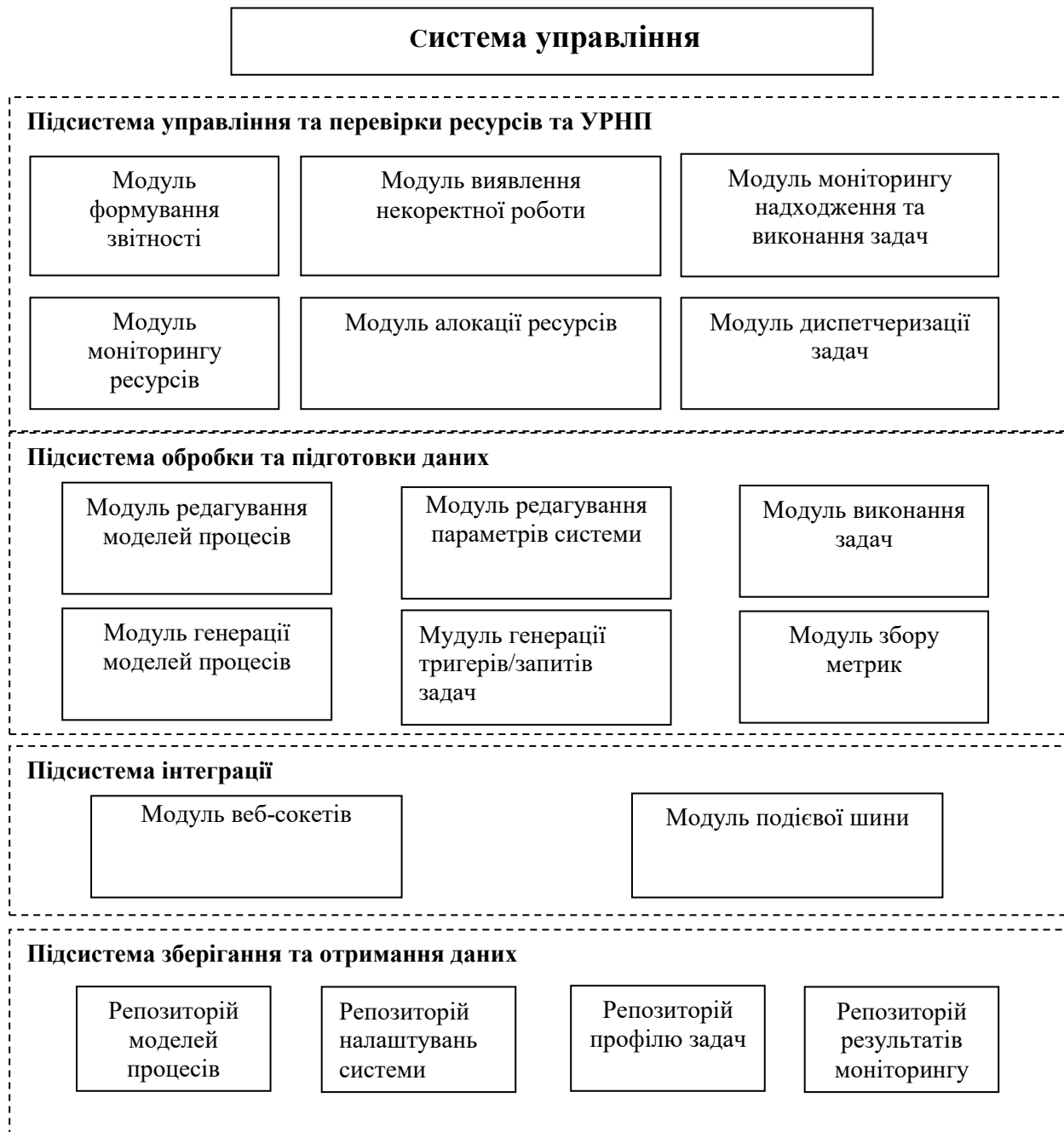


Рис. 4.9. Структурна схема системи управління та перевірки середовища проєктування та виконання бізнес-сервісів

Важливою особливістю системи є єдність керування на всіх рівнях. Верхній рівень відповідає за стратегічне управління - визначення політик диспетчеризації, пріоритетів виконання, параметрів УРНП і допустимих меж навантаження. Середній рівень виконує тактичні функції - моніторинг ресурсного стану, розподіл потоків задач і формування планів виконання. Нижній рівень реалізує операційні дії - виконання бізнес-процесів, обробку даних і фіксацію результатів у сховищах.

Така трирівнева ієрархія дозволяє досягти збалансованості між централізованим контролем і локальною автономією модулів, що забезпечує стабільність і передбачуваність системи навіть за умов високої динаміки навантажень.

Ключовим аспектом є система зворотного зв'язку, яка забезпечує безперервний моніторинг станів і метрик усіх підсистем. Отримані показники використовуються для аналітичного оцінювання ефективності роботи, виявлення аномалій і прогнозування ризиків. У разі виявлення відхилень від нормативних ЦПРС система автоматично ініціює коригувальні дії - перерозподіл задач або відкладене виконання завдань. Таким чином, формується адаптивний цикл керування, який поєднує механізми аналізу, планування й реакції на основі телеметричних даних.

У контексті програмно-технічної реалізації система управління забезпечує сумісність компонентів і використання стандартизованих інтерфейсів обміну. Це дозволяє підключати зовнішні сервіси моніторингу, системи візуалізації, аналітичні платформи або інструменти управління інцидентами без порушення внутрішньої логіки роботи. Через інтеграційну підсистему реалізується підтримка протоколів REST, WebSocket та внутрішніх подієвих каналів. Усі обміни відбуваються уніфіковано - у вигляді повідомлень з описом подій, що гарантує прозорість та відстежуваність бізнес-подій у системі.

Особливу роль у побудові системи відіграє підсистема моніторингу УРНП. Вона контролює дотримання часових, ресурсних та якісних параметрів виконання завдань. У разі перевищення допустимих меж підсистема надсилає сигнали в диспетчерський модуль, який здійснює повторне планування черг або перерозподіл завдань між виконавчими вузлами. Такий підхід реалізує концепцію динамічного керування навантаженням, що дозволяє підтримувати стабільну продуктивність навіть при нерівномірному потоці запитів.

Архітектурно система управління функціонує як єдиний керуючий простір даних. Усі підсистеми обмінюються структурованими об'єктами, які мають уніфіковані схеми опису - контексти, профілі, моделі та метрики. Це забезпечує узгодженість між модулями, зменшує дублювання інформації та спрощує

трасування подій. У результаті створюється логічно зв'язана екосистема, у якій будь-яка зміна на одному рівні миттєво відображається на інших, забезпечуючи актуальність даних і прозорість процесів.

Важливо підкреслити, що система управління виконує не лише функцію координації, а й забезпечує інтелектуальну підтримку прийняття рішень. Завдяки аналітичним модулям вона може формувати прогнозні моделі навантаження, визначати потенційні вузькі місця й оптимізувати використання ресурсів. Такий підхід відповідає сучасним принципам *intelligent process automation*, у межах яких керування здійснюється не лише за поточними показниками, а й з урахуванням прогнозованих тенденцій.

Таким чином, система управління є ядром керованої архітектури бізнес-процесів, у якому реалізовано баланс між централізацією управління, автономністю компонентів і гнучкістю конфігурації. Її багаторівнева структура забезпечує стабільність, масштабованість і передбачуваність роботи всього середовища. Завдяки інтеграції механізмів планування, моніторингу, аналітики й зберігання даних система формує єдине інформаційне поле, у межах якого відбувається управління повним життєвим циклом бізнес-сервісів - від проєктування до аналітичного оцінювання результатів.

4.3.2. Підсистема управління та перевірки ресурсів і УРНП у платформі автоматизації проєктування та виконання бізнес-сервісів

Підсистема управління та перевірки ресурсів і УРНП є ключовою складовою системи управління (Рис. 4.7) і виконує роль операційного центру, який забезпечує координацію виконання завдань, моніторинг використання ресурсів, виявлення відхилень та підтримання відповідності фактичних показників визначеним угодам рівня сервісу. Вона реалізує механізм балансування навантаження, формування політик диспетчеризації, контролю пріоритетів і аналітичного оцінювання ефективності використання обчислювальних потужностей у режимі реального

часу. Основною метою функціонування підсистеми є досягнення стійкого стану системи при динамічній зміні потоків завдань і ресурсних обмежень.

Архітектурно підсистема включає низку взаємопов'язаних модулів: модуль формування звітності, модуль виявлення некоректної роботи, модуль моніторингу надходження та виконання завдань, модуль моніторингу ресурсів, модуль алокації ресурсів і модуль диспетчеризації завдань. Кожен із них виконує спеціалізовану функцію, однак їхня взаємодія формує єдиний замкнений контур керування, у межах якого відбувається планування, контроль, корекція та звітування про результати роботи системи. Такий підхід забезпечує наскрізну відтворюваність усіх бізнес-подій і дозволяє формувати прозору картину ресурсного стану комплексу.

Модуль диспетчеризації завдань є ядром підсистеми, оскільки саме він координує потоки задач, формує чергу виконання, визначає виконавців і здійснює контроль статусів процесів. Модуль підтримує адаптивні алгоритми черговості, що враховують пріоритетність, ресурсомісткість, терміни виконання та рівень важливості бізнес-процесу. За умов перевантаження система автоматично перемикається між стратегіями планування - від фіксованого розподілу до динамічного балансування на основі поточного навантаження. Диспетчеризація реалізується як у синхронному, так і в асинхронному режимі, що дозволяє одночасно обробляти десятки або сотні потоків завдань без втрати керованості.

Функціональність модуля алокації ресурсів полягає у виділенні, резервуванні та поверненні ресурсів, необхідних для виконання задач. У системі передбачено механізм логічного групування ресурсів у так звані ресурсні кошики (resource tokens), які можуть бути призначені окремим бізнес-процесам або класам завдань. Такий підхід запобігає конфліктам при одночасному зверненні до тих самих обчислювальних або пам'ятевих потужностей. Алокаційний модуль працює у тісній зв'язці з диспетчером завдань, коригуючи його рішення відповідно до фактичної доступності ресурсів. Він також формує статистику використання, що надалі застосовується для прогнозування навантаження.

Модуль моніторингу ресурсів реалізує механізм збирання, оброблення та аналізу телеметричних показників щодо стану обчислювального середовища. До таких показників належать завантаження центрального процесора, обсяг використаної оперативної пам'яті, мережеві затримки, час відповіді підсистем введення/виведення та кількість активних потоків. Дані збираються у реальному часі та передаються до модулів аналізу й формування політик диспетчеризації. Особливістю є підтримка адаптивних тригерів: при перевищенні допустимих меж система автоматично ініціює попереджувальні події або викликає механізм перенесення задач на інші вузли.

Модуль виявлення некоректної роботи забезпечує безперервний контроль стабільності виконання процесів і виявлення ситуацій, які можуть призвести до порушення УРНП. До таких належать зависання процесів, надмірне споживання ресурсів, циклічне повторення завдань або невідповідність вхідних і вихідних даних очікуваним форматам. Модуль реалізує політику автоматичної реакції: після виявлення аномалії він ініціює запис події у системний журнал.

Модуль формування звітності виконує функцію аналітичного узагальнення інформації, що надходить із решти компонентів підсистеми. Він формує регулярні звіти про рівень виконання УРНП, середній час реакції системи, частоту помилок, завантаження ресурсів і динаміку змін продуктивності. Звіти можуть формуватися як у табличному вигляді, так і у вигляді графічних панелей для інтеграції з аналітичними інструментами. Цей модуль є важливою складовою процесу ухвалення рішень, оскільки надає керівництву можливість оцінювати ефективність роботи комплексу та приймати стратегічні рішення щодо оптимізації.

Модуль моніторингу надходження та виконання завдань контролює увесь життєвий цикл завдання - від моменту його реєстрації до завершення. Він відстежує час очікування у черзі, тривалість виконання, кількість спроб перезапуску й статуси завершення. Отримані дані використовуються для оцінювання стабільності процесів і побудови прогнозів щодо майбутнього навантаження. Завдяки постійному контролю часу проходження завдань система здатна оперативно

виявляти затримки та ініціювати коригувальні дії, що є критично важливим для підтримання УРНП-показників.

Підсистема реалізує багаторівневий механізм перевірки дотримання угод про рівень сервісу. Для кожного завдання визначаються допустимі межі - час реакції, тривалість виконання, споживання ресурсів, рівень відмов. Порухення хоча б одного з параметрів призводить до автоматичної генерації події типу УРНП Alert, яка обробляється у модулі диспетчеризації. На основі таких подій система може змінювати черговість виконання, призупиняти надходження нових завдань або ініціювати процедури масштабування ресурсів. Це дозволяє підтримувати стабільний рівень обслуговування навіть за умов непередбачуваних навантажень.

Підсистема управління та перевірки ресурсів і УРНП тісно взаємодіє з підсистемою обробки даних і підсистемою інтеграції. З першою вона обмінюється метриками, необхідними для формування політик управління, а з другою - сигналами телеметрії, які передаються до аналітичних панелей. Зокрема, дані про виконання завдань і використання ресурсів можуть бути автоматично візуалізовані у середовищі Grafana або проаналізовані за допомогою зовнішніх інструментів. Така взаємодія створює єдине інформаційне поле, у межах якого керування, аналітика й візуалізація функціонують у синхронному режимі.

З метою підвищення ефективності використання ресурсів підсистема застосовує адаптивні алгоритми балансування навантаження, засновані на принципах динамічного оцінювання стану системи. Зокрема, при зміні кількості активних задач або при досягненні граничних порогів використання пам'яті система може автоматично перерозподіляти потоки виконання між вузлами або ініціювати масштабування у хмарному середовищі. Використання таких алгоритмів забезпечує не лише стабільність, а й високу продуктивність у розподілених конфігураціях.

Усі дії, що виконуються в межах підсистеми, фіксуються у системному журналі. Запис подій включає часові мітки, ідентифікатори завдань, типи операцій, статуси виконання й діагностичні повідомлення. Наявність повного журналу дозволяє не лише здійснювати аудит подій, а й виконувати відтворення сценаріїв

для аналітичного аналізу або тестування. Система підтримує централізоване зберігання журналів із можливістю реплікації, що гарантує надійність і відмовостійкість навіть при великих обсягах даних.

Таким чином, підсистема управління та перевірки ресурсів і УРНП є ядром оперативного керування комплексною системою. Вона поєднує функції моніторингу, диспетчеризації, алокації та звітності, забезпечуючи повний контроль за станом виконання бізнес-процесів і використанням ресурсів. Її модульна структура дозволяє масштабувати функціональність, розширювати набір політик керування та інтегрувати додаткові аналітичні інструменти без зміни базової архітектури. Завдяки цьому забезпечується стабільність роботи комплексу, відповідність ЦПРС-показникам і висока адаптивність системи до динамічних умов експлуатації.

4.3.3. Підсистема обробки та підготовки даних у середовищі проєктування та виконання бізнес-сервісів

Підсистема обробки та підготовки даних є центральною аналітичною складовою системи управління, що забезпечує формування, трансформацію, валідацію та агрегування даних, необхідних для роботи інших підсистем - управління, виконання, інтеграції та зберігання. Її головне завдання полягає у підготовці структурованих інформаційних об'єктів, які використовуються для планування, моніторингу, аналізу та прийняття рішень. Завдяки цій підсистемі забезпечується узгодженість інформаційних потоків, підтримка єдиних форматів даних і їхнє своєчасне оновлення, що є критично важливим для стабільного функціонування всієї системи.

Загальна структура підсистеми базується на принципі модульного поділу функцій, який передбачає розмежування процесів редагування, генерації, виконання та збору метрик. Основними компонентами є: модуль редагування моделей процесів, модуль редагування параметрів системи, модуль виконання завдань, модуль генерації моделей процесів, модуль генерації тригерів або запитів

завдань та модуль збору метрик. Кожен із цих модулів відповідає за конкретний етап підготовки даних, але в сукупності вони реалізують безперервний цикл перетворення інформації - від концептуальної моделі до практичного використання у процесах управління.

Модуль редагування моделей процесів виконує функції створення, коригування та оновлення структур бізнес-процесів, які визначають логіку функціонування сервісів у межах системи. Його основне завдання полягає у формуванні моделей, сумісних із підсистемою виконання завдань. Редагування здійснюється з урахуванням вимог нотацій BPMN, DMN, що забезпечує універсальність і стандартизацію опису процесів. Крім того, модуль підтримує перевірку синтаксичної та семантичної коректності побудови моделей, виявлення циклічних залежностей і розривів у потоках управління, що унеможливорює виникнення логічних помилок на етапі виконання.

Модуль редагування параметрів системи забезпечує налаштування параметрів функціонування системи відповідно до вимог бізнес-логіки. Йдеться про встановлення значень змінних середовища, визначення ресурсних квот, налаштування таймінгів виконання та параметрів моніторингу. Модуль тісно пов'язаний із підсистемою управління, оскільки його зміни впливають на політики диспетчеризації та розподілу ресурсів. Важливо, що він підтримує версіонування параметрів - кожне оновлення фіксується у системному журналі, що забезпечує відтворюваність конфігурацій і можливість повернення до попередніх станів.

Модуль виконання завдань у межах цієї підсистеми має допоміжний характер і відповідає за тестування, перевірку та обчислення проміжних результатів, необхідних для валідації моделей і параметрів. Наприклад, він може запускати симуляційні сценарії бізнес-процесів, виконувати контрольні обчислення або перевіряти реакцію системи на певні вхідні дані. Отримані результати використовуються для налаштування політик керування, прогнозування навантаження та оптимізації ресурсного використання.

Модуль генерації моделей процесів автоматизує створення моделей на основі наявних шаблонів або історичних даних. Він використовує механізми повторного

використання процесних компонентів (reuse), що значно скорочує час моделювання й знижує ризик помилок. Алгоритми модуля можуть формувати базові каркаси процесів, інтегруючи раніше використані завдання, логічні зв'язки та правила переходів. Крім того, він підтримує функцію автоматичної адаптації моделей до змін середовища - наприклад, зміни складу ресурсів чи пріоритетів завдань. У поєднанні з модулем редагування параметрів системи це забезпечує гнучкість і актуальність моделювання.

Функціональність модуля генерації тригерів і запитів завдань полягає у створенні тригерів, які активують виконання завдань. Такі тригери можуть бути як часовими (плановими), так і подієвими, тобто активованими при досягненні визначених умов у системі. Модуль також формує запити до інших підсистем, наприклад, для отримання даних з репозиторіїв або відправлення результатів у сховище. Реалізація механізму тригерів забезпечує реактивність підсистеми, дозволяючи їй оперативно реагувати на зміни стану системи або надходження нових завдань.

Модуль збору метрик виконує збір і агрегацію показників, що характеризують ефективність роботи процесів. Серед таких показників - час виконання операцій, кількість оброблених об'єктів, частота помилок, обсяг використаних ресурсів і рівень навантаження системи. Отримані метрики передаються до аналітичних підсистем або формування звітностей. На основі цих даних формуються узагальнені звіти або застосовуються для автоматичного коригування параметрів виконання. Модуль також підтримує фільтрацію та нормалізацію метрик, що забезпечує їхню порівнянність між різними процесами.

Підсистема обробки та підготовки даних реалізує багатоступеневу систему узгодження потоків інформації. На початковому етапі здійснюється очищення та верифікація даних, потім - синтаксична і семантична нормалізація, а на завершальному - агрегація та передача результатів у репозиторії. Для цього застосовуються методи ETL (Extract, Transform, Load), що забезпечують узгодженість між джерелами й споживачами даних. Кожен етап супроводжується

автоматичними перевітками цілісності, що гарантує достовірність і повторюваність результатів.

Підсистема обробки даних виконує функцію проміжної ланки між рівнями проєктування та виконання. З підсистемою управління вона обмінюється інформацією про ресурси, пріоритети та статуси завдань, тоді як із підсистемою інтеграції - метриками і подієвими сигналами. Отримані результати передаються в підсистему зберігання для подальшого архівування та аналітичної обробки. Така багатоканальна взаємодія забезпечує наскрізний інформаційний потік, який підтримує актуальність і логічну узгодженість усіх даних у системі.

Підсистема обробки та підготовки даних є невід'ємною частиною життєвого циклу бізнес-процесів, оскільки саме вона трансформує моделі, створені на етапі проєктування, у готові до виконання об'єкти. Вона забезпечує перехід від концептуального до операційного рівня представлення процесів. Крім того, вона створює основу для формування політик диспетчеризації, які враховують ресурсну вартість і часові обмеження завдань. Таким чином, підсистема стає ключовою ланкою між бізнес-логікою й технічним виконанням.

Узагальнюючи, можна стверджувати, що підсистема обробки та підготовки даних є інтелектуальним ядром аналітичної підтримки системи управління. Вона поєднує функції створення моделей, налаштування параметрів, генерації завдань і збору метрик, забезпечуючи узгоджене функціонування усіх підсистем комплексу. Її модульна структура дозволяє масштабувати функціональність, розширювати алгоритми аналізу й адаптуватися до нових вимог бізнес-середовища. Завдяки цьому система набуває властивостей гнучкості, передбачуваності й здатності до самооптимізації, що відповідає сучасним концепціям інтелектуального управління бізнес-процесами.

4.3.4. Підсистема інтеграції у середовищі проєктування та виконання бізнес-сервісів

Підсистема інтеграції є комунікаційним ядром системи управління середовищем проєктування та виконання бізнес-сервісів. Її головне завдання полягає у забезпеченні інформаційної взаємодії між внутрішніми підсистемами комплексу, а також у синхронізації даних і подій із зовнішніми системами аналітики, моніторингу та керування інфраструктурою. Вона виконує роль зв'язувальної ланки, яка координує потоки повідомлень, забезпечує сумісність протоколів і підтримує єдині стандарти передавання даних у реальному часі.

Архітектурна побудова підсистеми інтеграції базується на принципах Service-Oriented Architecture (SOA) та Event-Driven Architecture (EDA), що забезпечує асинхронний, слабо зв'язаний обмін між компонентами. Такий підхід дозволяє підсистемі функціонувати незалежно від конкретних технологічних реалізацій інших елементів системи, підтримуючи масштабованість, еластичність і надійність комунікацій. Основними складовими підсистеми є: модуль веб-сокетів, модуль подієвої шини і модуль сервісу журналу процесів.

Модуль веб-сокетів забезпечує двосторонній канал обміну даними між клієнтськими інтерфейсами, підсистемами керування й виконанням процесів. Його головна перевага полягає у підтримці постійного з'єднання, яке дозволяє передавати повідомлення миттєво, без необхідності повторних запитів до сервера. Така комунікаційна модель мінімізує затримки та забезпечує реактивність системи. Завдяки модулю веб-сокетів користувачі отримують оновлення про стан процесів, результати моніторингу чи зміни параметрів у реальному часі, що суттєво підвищує зручність і надійність керування.

Модуль подієвої шини є центральним елементом комунікаційної інфраструктури підсистеми інтеграції. Вона відповідає за маршрутизацію подій, управління чергами повідомлень і гарантію доставки даних між компонентами системи. Реалізація механізму publish–subscribe забезпечує можливість підписки окремих модулів на певні типи подій, що дозволяє забезпечити слабке зв'язування

між ними. Завдяки цьому будь-який компонент може бути оновлений або замінений без зміни логіки роботи решти системи.

Подієва шина підтримує ієрархію каналів, що дає змогу розмежовувати потоки даних за рівнями критичності або типами джерел. Наприклад, події з моніторингу ресурсів мають вищий пріоритет, ніж інформаційні повідомлення користувача. Такий механізм гарантує своєчасне реагування на потенційно небезпечні стани системи, запобігаючи перевантаженням або відмовам.

Модуль сервісу журналу процесів виконує роль посередника між підсистемою виконання бізнес-процесів і зовнішніми інструментами моніторингу. Він акумулює дані про всі події, що виникають у процесі виконання - запуск, завершення, відмову, повторне виконання, зміну статусів або затримку. Модуль уніфікує формат подій і передає їх у подієву шину, де вони можуть бути спожиті іншими компонентами системи. Завдяки цьому забезпечується наскрізна простежуваність усіх процесів і повна історія їхнього виконання.

Таким чином, підсистема інтеграції є невід'ємною складовою системи управління, що забезпечує єдність інформаційного простору та наскрізний потік даних між усіма її рівнями. Завдяки реалізації принципів сервісно-орієнтованої та подієво-орієнтованої архітектури вона забезпечує масштабовану, реактивну та захищену побудову системи комунікацій, у межах якої всі події фіксуються, синхронізуються й аналітично опрацьовуються в реальному часі. Підсистема створює передумови для впровадження гібридних сценаріїв керування, де автоматизовані реакції системи поєднуються з ручними діями оператора, забезпечуючи стійкість, прозорість і адаптивність у високонавантажених розподілених середовищах.

4.3.5. Підсистема зберігання та отримання даних в середовищі проєктування та виконання бізнес-сервісів

Підсистема зберігання та отримання даних є центральною інформаційною ланкою системи управління середовищем проєктування та виконання бізнес-

сервісів. Її головне призначення полягає у забезпеченні цілісного, узгодженого й безпечного збереження всіх структурних елементів даних, що формуються під час роботи інших підсистем комплексу. Вона створює єдиний простір даних, у межах якого здійснюється накопичення, структуризація, актуалізація та надання інформації для моделювання, моніторингу, аналітики та управління.

Підсистема включає чотири основні модулі: репозиторій моделей процесів, репозиторій налаштувань системи, репозиторій профілів задач і репозиторій результатів моніторингу. Кожен із них виконує спеціалізовану функцію в межах загального інформаційного контуру, але всі разом вони утворюють узгоджену структуру з наскрізним контролем даних і двосторонньою взаємодією з іншими підсистемами комплексу. У результаті формується повний цикл оброблення інформації - від створення моделі до збереження результатів її виконання.

Репозиторій моделей процесів є базовим джерелом структурних описів бізнес-процесів, створених у підсистемі обробки та підготовки даних. Кожен запис у репозиторії містить опис потоків виконання, логіку переходів між завданнями, зв'язки між ресурсами, а також параметри управління подіями. Усі моделі мають унікальні ідентифікатори, що дозволяє забезпечити повну трасованість змін у межах життєвого циклу процесу.

Репозиторій підтримує версійність моделей, що дає можливість відстежувати історію редагувань, порівнювати відмінності між версіями та повертатися до попередніх редакцій у разі потреби. Така функціональність є критично важливою для забезпечення відтворюваності експериментів, проведення тестування або аудиту. Крім того, він підтримує семантичну валідацію моделей - автоматичну перевірку відповідності структури процесу внутрішнім правилам системи, що мінімізує ризик появи логічних помилок під час виконання.

Репозиторій налаштувань системи виконує функцію централізованого сховища параметрів конфігурації та системних політик. У ньому зберігаються відомості про структуру ресурсів, обмеження навантаження, часові інтервали, параметри моніторингу, правила диспетчеризації, політики УРНП, а також індивідуальні налаштування підсистем.

Усі конфігураційні об'єкти мають атрибути часу створення, дії та автора, що забезпечує повну відтворюваність станів системи. Завдяки цьому можна відновити попередню конфігурацію в разі помилкових змін або провести аналіз впливу певних параметрів на продуктивність.

Репозиторій налаштувань системи також забезпечує механізм автоматичної актуалізації параметрів - у разі зміни конфігурацій підсистема управління отримує оновлені значення без необхідності ручного втручання. Така модель динамічної синхронізації забезпечує узгодженість усіх компонентів комплексу, навіть у випадку одночасних оновлень.

Репозиторій профілів задач виконує роль довідника типових завдань, що використовуються в межах бізнес-процесів. Кожен профіль містить опис логіки виконання задачі, набір вхідних і вихідних параметрів, залежності від ресурсів, пріоритети, вимоги до середовища. Наявність такого репозиторію забезпечує повторне використання типових компонентів, що значно скорочує час проєктування процесів і знижує ризик помилок під час налаштування.

Крім структурних параметрів, профілі задач можуть містити історичні статистичні дані, які використовуються для аналітичного прогнозування. Наприклад, система може визначати середній час виконання певного типу завдання або рівень його ресурсоємності, що допомагає диспетчерському модулю оптимізувати розподіл навантаження.

Репозиторій профілів задач безпосередньо взаємодіє з підсистемами управління та моніторингу, передаючи інформацію про стандартизовані шаблони і результати їхнього використання.

Репозиторій результатів моніторингу акумулює дані про виконання процесів, стан ресурсів, час реакції, відхилення від УРНП, статистику навантаження й результати аналізу подій. Він є центральним джерелом інформації для підсистем аналітики та інтеграції, що формують візуальні панелі, звіти та прогнозні моделі.

Дані, які надходять до цього репозиторію, проходять попередню обробку: нормалізацію, агрегацію та фільтрацію. Це дозволяє усунути надлишкову інформацію й забезпечити узгодженість метрик між різними джерелами.

Репозиторій підтримує часові мітки та кореляційні ідентифікатори подій, що дозволяє відтворювати історію виконання процесів у хронологічній послідовності.

Особливо важливою функцією є збереження аналітичних вибірок, які використовуються для виявлення трендів і прогнозування навантаження. На їхній основі система може адаптувати свої політики керування, що є складовою реалізації інтелектуальної автоматизації.

Підсистема зберігання та отримання даних функціонує як єдиний інформаційний вузол, який об'єднує дані з усіх підсистем комплексу.

Від підсистеми обробки даних вона отримує моделі процесів і профілі задач.

Від підсистеми управління - конфігураційні параметри, результати виконання й статистичні звіти.

Від підсистеми інтеграції - телеметричні показники, сигнали моніторингу та журнал подій.

Такі зв'язки забезпечують повну прозорість інформаційних потоків і можливість синхронного оновлення даних у всіх рівнях архітектури.

Таким чином, підсистема зберігання та отримання даних виконує функцію інформаційного фундаменту системи управління. Вона забезпечує централізоване накопичення, збереження та надання даних усім іншим підсистемам, створюючи умови для їхньої узгодженої роботи. Завдяки чіткій структурі репозиторіїв - моделей процесів, налаштувань системи, профілів задач і результатів моніторингу - досягається прозорість інформаційних потоків, стабільність і можливість відновлення будь-якого стану системи. Така структура формує основу для побудови високонадійного, масштабованого та аналітично керованого середовища виконання бізнес-процесів.

4.4. Експериментальні дослідження параметрів засобів моніторингу виконання кроків бізнес-процесів

Експериментальні дослідження проводилися з метою кількісного визначення значення критерію ефективності виконання кроків бізнес-процесів у контрольованому середовищі, а саме частки порушення дедлайнів (далі — частки порушень). У межах експерименту під часткою порушень розуміється відношення кількості кроків, для яких зафіксовано невиконання заданого часового обмеження, до загальної кількості кроків, виконаних у відповідному прогоні. Обраний показник є інтерпретованим та придатним для порівняння конфігурацій за умови, що навантаження та обмеження середовища відтворюються однаково, зокрема - при фіксованих лімітах доступної оперативної пам'яті та однаковій структурі сценарію.

Експериментальні дослідження проводилися на віртуальному сервері x64, конфігурація якого включала процесор серверного класу Intel, 32 ГБ оперативної пам'яті та операційну систему Windows 11 Pro. Програмне середовище експериментів базувалося на платформі .NET 8, що забезпечує актуальні механізми керування ресурсами, оптимізовані механізми збирання сміття, сучасну модель виконання багатопотокових обчислень та можливість високоточного вимірювання параметрів системи під час виконання. Обрана апаратно-програмна конфігурація забезпечує достатній запас ресурсів для моделювання навантажувальних сценаріїв і аналізу метрик у контрольованих умовах, що дозволяє розглядати зафіксовані зміни частки порушень насамперед як наслідок різниці в організації виконання та спостереженні за ресурсами, а не як ефект обмеженості базового апаратного забезпечення.

Експериментальні сценарії навантаження побудовані за різним співвідношенням ЦП-інтенсивних та ОП-інтенсивних кроків бізнес-процесів. Було сформовано два типи сценаріїв, які відрізняються часткою кроків із підвищеним використанням оперативної пам'яті. Такий підхід дозволяє окремо оцінити поведінку системи за помірного рівня конкуренції за пам'ять (коли ОП-інтенсивні

кроки є меншістю), а також у режимі підвищеного тиску на пам'ять (коли частка ОП-інтенсивних кроків наближається до половини загального потоку кроків).

Перший сценарій включав 5000 згенерованих кроків, із яких 3992 були ЦП-інтенсивними, а 1008 - ОП-інтенсивними. Це відповідає приблизно 20% кроків із підвищеним використанням оперативної пам'яті та 80% кроків, для яких домінуючим ресурсом є процесорний час. Така структура відображає поширену для багатьох прикладних бізнес-процесів ситуацію, коли більшість кроків становлять обчислювальні або керувальні операції, тоді як менша частина пов'язана з інтенсивним використанням оперативної пам'яті (наприклад, маніпулювання структурами даних, обробка текстових представлень, формування проміжних наборів даних).

Другий сценарій також складався з 5000 згенерованих кроків, однак у цьому випадку співвідношення було близьким до симетричного: 2495 ЦП-інтенсивних та 2505 ОП-інтенсивних кроків (приблизно 50% на 50%). Така структура моделює складніші умови, у яких конкуренція за оперативну пам'ять посилюється, а вплив пам'яті на часові характеристики виконання стає суттєвим для значної частини потоку кроків. Саме цей сценарій використано для оцінювання стійкості конфігурацій у режимі підвищеної інтенсивності операцій, що використовують оперативну пам'ять.

У межах обох сценаріїв виконання кроків досліджувалося при різних обмеженнях доступної оперативної пам'яті - від 300 МБ до 390 МБ з кроком 10 МБ. Це дозволило отримати дискретний набір режимів, який охоплює як зону жорсткого дефіциту пам'яті (нижні значення ліміту), так і зону, у якій обмеження є менш критичним, але все ще впливає на поведінку системи. Важливо зазначити, що обмеження пам'яті застосовувалися програмно й не залежать від реального обсягу оперативної пам'яті робочої станції. Таким чином, середовище виконання є контрольованим, а порівняння між конфігураціями виконується за однакових лімітів та однакового розподілу типів кроків.

Порівняння проводилося між базовою конфігурацією з урахуванням ресурсних обмежень та запропонованою конфігурацією з урахуванням ресурсних

обмежень, у якій застосовано вдосконалені підходи до організації виконання та спостереження за використанням ресурсів кроками бізнес-процесів. Обидві конфігурації оцінювалися на однакових наборах значень доступної пам'яті, що забезпечує коректність порівняння результатів.

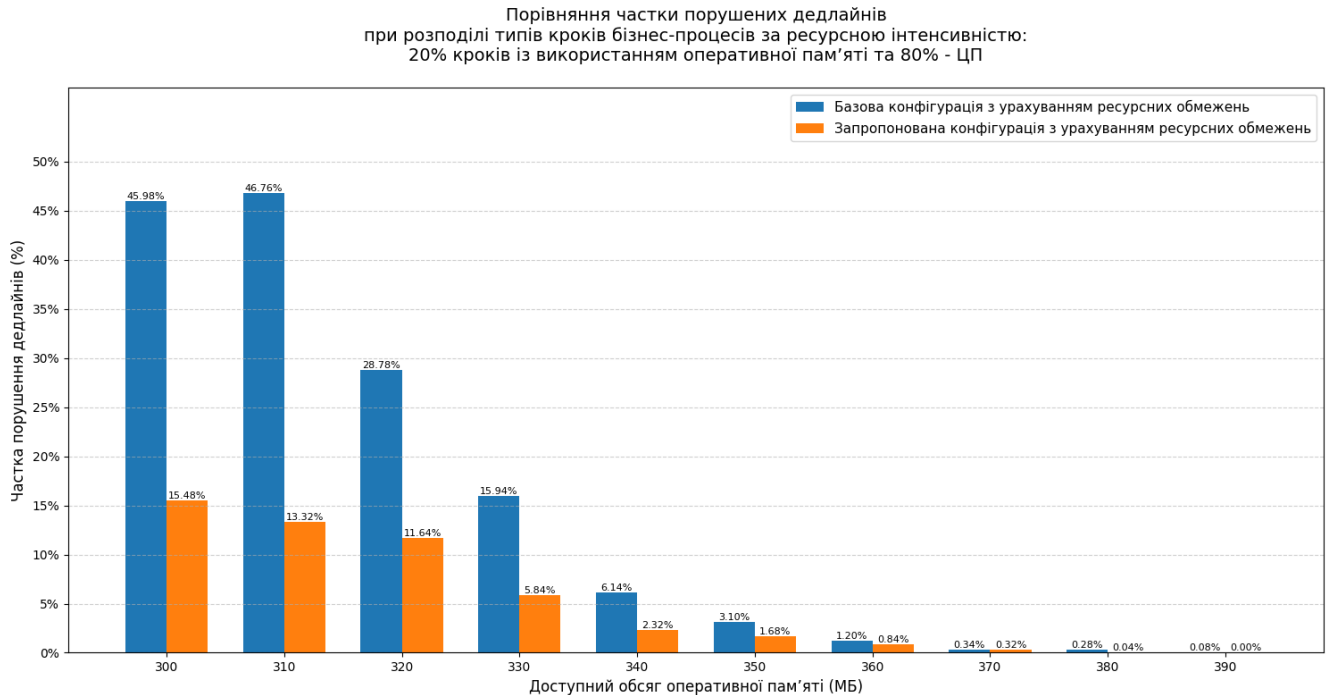


Рис. 4.10. Порівняння частки порушених дедлайнів при розподілі типів кроків бізнес-процесів за ресурсами 20% кроків із використанням оперативної пам'яті та 80% - ЦП

У першому сценарії (Рис. 4.10.) спостерігається зменшення частки порушень із підвищенням доступного обсягу оперативної пам'яті, причому базова конфігурація демонструє більш виражену чутливість у зоні нижніх значень ліміту. За обмежень 300–310 МБ частка порушень у базовому варіанті становить 45.98–46.76%, тоді як в запропонованому - 13.32–15.48%. Для 320 МБ зафіксовано 28.78% порушень у базовій конфігурації та 11.64% - в запропонованій. Надалі зі збільшенням доступної пам'яті до 330 МБ показник зменшується до 15.94% у базовому варіанті та до 5.84% в запропонованому. Для 340 МБ частка порушень становить 6.14% (базова) та 2.32% (запропонована).

У середній та верхній частинах досліджуваного діапазону частка порушень у двох конфігураціях переходить до малих значень. Зокрема, при 350 МБ у базовій конфігурації зафіксовано 3.10%, а в запропонованій - 1.68%; при 360 МБ - 1.20% та 0.84% відповідно. При 370 МБ значення стають близькими (0.34% для базової та 0.32% для запропонованої), що відповідає режиму, у якому обмеження оперативної пам'яті вже менш суттєво впливає на часові характеристики виконання для цього співвідношення типів кроків. На верхній межі діапазону фіксуються поодинокі порушення: при 380 МБ - 0.28% у базовій конфігурації та 0.04% в запропонованій, а при 390 МБ - 0.08% у базовій та 0.00% в запропонованій.

Таким чином, для сценарію з 20% ОП-інтенсивних кроків запропонована конфігурація забезпечує нижчі значення частки порушень у всьому діапазоні 300–390 МБ, а найбільша різниця спостерігається в зоні 300–330 МБ, де відмінності між конфігураціями вимірюються десятками відсоткових пунктів.

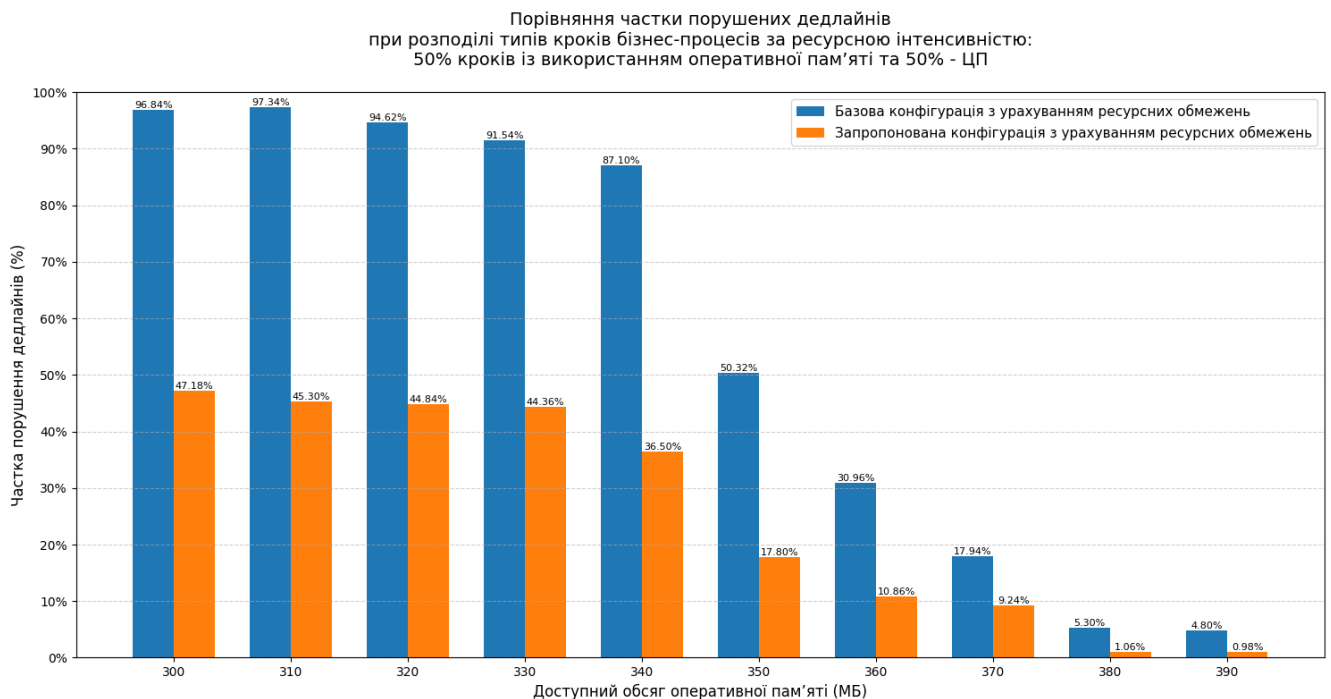


Рис. 4.11. Порівняння частки порушених дедлайнів при розподілі типів кроків бізнес-процесів за ресурсами 50% кроків із використанням оперативної пам'яті та 50% - ЦП

У другому сценарії (Рис. 4.11.), де частка ОП-інтенсивних кроків збільшена до 50%, рівень порушень істотно зростає для обох конфігурацій у зоні малих значень доступної пам'яті, що відображає посилення конкуренції за оперативну пам'ять. Для базової конфігурації при 300–330 МБ зафіксовано 96.84% (300 МБ), 97.34% (310 МБ), 94.62% (320 МБ) та 91.54% (330 МБ). При 340 МБ показник становить 87.10%. Тобто в інтервалі 300–340 МБ переважна більшість кроків не вкладається в задані часові обмеження.

Запропонована конфігурація в тому самому інтервалі демонструє нижчі значення частки порушень: 47.18% (300 МБ), 45.30% (310 МБ), 44.84% (320 МБ), 44.36% (330 МБ) та 36.50% (340 МБ). Зафіксована різниця між конфігураціями в зоні 300–340 МБ є суттєвою: вона становить десятки відсоткових пунктів і відображає зменшення частоти порушень дедлайнів за тих самих лімітів пам'яті.

Починаючи з 350 МБ, спостерігається помітне зменшення частки порушень у двох конфігураціях, однак із різними абсолютними рівнями. Для базового варіанта при 350 МБ частка порушень становить 50.32%, при 360 МБ - 30.96%, при 370 МБ - 17.94%, при 380 МБ - 5.30%, при 390 МБ - 4.80%. Для запропонованого варіанта відповідні значення дорівнюють 17.80%, 10.86%, 9.24%, 1.06% та 0.98%. Отже, у зоні 350–370 МБ запропонована конфігурація підтримує показник на рівні приблизно 9–18%, тоді як базова конфігурація в цьому самому інтервалі зберігає суттєво вищі значення (приблизно 18–50%). На верхній межі діапазону (380–390 МБ) запропонована конфігурація наближається до рівня близько 1%, тоді як базова залишається в діапазоні близько 4.8–5.3%.

Табл. 4.1.

Порівняння середніх часток порушення дедлайнів для базової та запропонованої конфігурацій при різній інтенсивності кроків, що використовують ОП

Показник	20% ОП-інтенсивних кроків	50% ОП-інтенсивних кроків

Середнє значення частки порушення дедлайнів для базової конфігурації	14.56	67.88
Середнє значення частки порушення дедлайнів для запропонованої конфігурації	5.48	25.81
Покращення	2.66	2.63

Таким чином, графіки (Рис. 4.10., Рис. 4.11.) демонструють узгоджену тенденцію: за однакових обмежень доступної оперативної пам'яті запропонована конфігурація забезпечує меншу частку порушення дедлайнів порівняно з базовою. Ефект є найбільш вираженим у зоні жорсткого дефіциту пам'яті (300–340 МБ) та в сценарії з підвищеною часткою ОП-інтенсивних кроків, де навантаження на пам'ять виступає визначальним фактором. Одночасно зі збільшенням доступного обсягу оперативної пам'яті обидві конфігурації демонструють зниження частки порушень, однак запропонована конфігурація зберігає перевагу в усьому дослідженому діапазоні та забезпечує менші значення як у термінах абсолютної частки порушень, так і за інтегральними (середніми) оцінками, наведеними в Табл. 4.1.

4.5. Розробка рекомендацій по впровадженню запропонованих моделей та методів

Розроблені у межах дослідження методи, моделі та архітектурні рішення спрямовані на підвищення ефективності процесів проєктування, управління та виконання бізнес-сервісів у ПНОП та гібридних корпоративних системах. Їх упровадження у практичну діяльність потребує не лише технічної реалізації, а й адаптації до існуючих бізнес-процесів, інформаційних інфраструктур та організаційних моделей управління. Тому на даному етапі формуються узагальнені

рекомендації щодо впровадження, стандартизації та поступового розгортання запропонованих рішень у реальних умовах експлуатації.

Основою впровадження є концепція ресурсно-орієнтованого управління виконанням бізнес-процесів, яка базується на поєднанні двох підходів - адаптивного планування (через політики диспетчеризації) та реактивного моніторингу (через безперервний аналіз стану ресурсів). Реалізація цього принципу передбачає інтеграцію таких елементів: підсистеми моніторингу ресурсів, підсистеми управління виконанням бізнес-процесів, підсистеми моделювання процесів і підсистеми інтеграції. Рекомендовано впроваджувати дану архітектуру у три основні фази: пілотне розгортання, функціональне масштабування та організаційна інтеграція.

На етапі пілотного розгортання доцільно створити ізольоване тестове середовище, у якому реалізуються основні компоненти запропонованої платформи: модуль диспетчеризації завдань, модуль моніторингу ресурсів, репозиторій профілів задач і модуль аналітичного моніторингу. Це дозволить відпрацювати базову взаємодію між компонентами та провести калібрування параметрів політик диспетчеризації. Пілотний запуск має проводитися з обмеженою кількістю бізнес-процесів (наприклад, до 10 типових моделей різного навантаження), налаштувати оптимізацію можна з допомогою модуля проектування БП, що дозволяє отримати об'єктивні дані щодо продуктивності системи, часу реакції на події та ефективності використання ресурсів. На основі отриманих результатів формується конфігураційний профіль системи - набір параметрів, які визначають поведінку алгоритмів планування, частоту моніторингу, пріоритети задач і обмеження використання ресурсів.

На етапі функціонального масштабування система поступово інтегрується з існуючими корпоративними середовищами. Це включає підключення зовнішніх джерел даних (через API або шину подій), підключення сервісів журналювання та аналітики. Особливу увагу слід приділити механізмам синхронізації між підсистемою зберігання та підсистемою виконання, оскільки саме узгодженість даних є критичною умовою достовірності результатів. Рекомендується

застосовувати покращений метод логуювання [5]. Така модель знижує навантаження на СУБД і підвищує швидкість реакції системи при високому рівні паралельності запитів.

З технічного погляду, впровадження запропонованих моделей вимагає стандартизації процесів моделювання, виконання й моніторингу. Для цього рекомендовано створити реєстр процесних моделей, який міститиме їхні версії, стан узгодження, атрибути ресурсоемності та статистику виконання. Такий реєстр виступає не лише сховищем даних, а й основою для формування аналітичних показників і розроблення політик оптимізації. Кожна модель у реєстрі має бути описана у вигляді метаданих, що містять тип процесу, середній час виконання, пріоритет, ресурсний профіль і зв'язки із зовнішніми системами. Це дозволяє формувати аналітичні залежності між типом процесу та рівнем споживання ресурсів, що в подальшому слугує базою для адаптивного планування.

Для ефективного впровадження запропонованого гібридного методу планування задач доцільно використовувати механізми поступової еволюції політик. Це означає, що спочатку в системі може застосовуватись перехід від класичного планування за принципом черги FCFS до модифікованого принципу черги FCFS із перевітками на доступність ресурсів, а після накопичення достатньої кількості статистичних даних активується адаптивна компонента, яка враховує дедлайни, вагу завдань і поточне використання ресурсів. Такий поступовий підхід знижує ризики дестабілізації системи у період впровадження і дає змогу здійснювати постійне порівняння продуктивності між класичними та новими алгоритмами. Для забезпечення прозорості цього процесу необхідно налаштувати модуль аналітичного моніторингу Grafana з панелями порівняння показників: середнього часу обробки, кількості порушених дедлайнів, використанням ОП та навантаження на ЦП.

З організаційної точки зору, важливою умовою успішного впровадження є наявність процесуального адміністрування. Це передбачає створення ролі “адміністратора бізнес-процесів”, який відповідає за затвердження нових моделей, контроль виконання, аналіз статистики та оптимізацію ресурсного розподілу.

Також слід передбачити роль “аналітика процесів”, який на основі даних підсистеми моніторингу визначатиме неефективні ділянки бізнес-логіки. Формування таких ролей підвищує керованість системи, забезпечуючи баланс між технічною автоматизацією та людським контролем.

Іншим важливим аспектом практичного впровадження є адаптація до обчислювального середовища. Для промислових інсталяцій доцільно реалізувати багаторівневу інфраструктуру з виділенням середовищ dev, staging і production. У середовищі розроблення (dev) тестуються нові моделі процесів і перевіряються політики диспетчеризації. У проміжному середовищі (staging) здійснюється тестування інтеграцій із зовнішніми системами, а в продуктивному (production) середовищі - контрольна експлуатація з реальними навантаженнями. Такий підхід забезпечує керованість змінами й дає змогу виявляти проблеми на ранніх етапах без впливу на роботу основного сервісу.

Серед технічних рекомендацій особливо важливими є:

Використання контейнеризації (Docker/Kubernetes) для ізоляції компонентів і забезпечення масштабованості.

Впровадження механізму самодіагностики підсистем моніторингу, який дозволяє відновлювати зв'язки при втраті доступу до вузлів.

Періодична калібровка параметрів політик диспетчеризації на основі статистики реального виконання для забезпечення стабільності планувальника.

Автоматизація оновлення моделей через інтеграцію з репозиторіями процесів і системами контролю версій (наприклад, GitLab).

Таким чином, рекомендації щодо впровадження запропонованих методів охоплюють як технічні, так і організаційні аспекти, включаючи стандартизацію моделей, інтеграцію із зовнішніми системами, створення процесних ролей і впровадження процедур моніторингу та аудиту. Реалізація зазначених підходів забезпечує перехід від традиційного реактивного управління бізнес-процесами до інтегрованої системи адаптивного керування, здатної самостійно оптимізувати використання ресурсів, знижувати навантаження та підвищувати стабільність виконання. У підсумку це створює технологічну основу для формування

інтелектуального середовища автоматизованого проєктування та виконання бізнес-сервісів, орієнтованого на самонавчання, прогнозування та безперервне вдосконалення.

Висновки до розділу 4

Представлено узагальнену інформаційну модель функціонування середовища проєктування та виконання бізнес-сервісів, яка охоплює взаємодію рівнів даних, контролю та розробки. Модель формалізує структури, що беруть участь у виконанні бізнес-процесів, зокрема профілі задач, моделі процесів, політики диспетчеризації та конфігураційні параметри, а також забезпечує задання правил генерування навантаження, контролю ресурсів і збору телеметрії для проведення відтворюваних експериментів.

Побудовано структурну та функціональну схеми системи управління і перевірки середовища проєктування та виконання бізнес-сервісів. Структурна схема відображає багаторівневу архітектуру з модулями диспетчеризації задач, алокації ресурсів, моніторингу, виявлення некоректної роботи, редагування моделей, формування звітів і відповідними репозиторіями. Функціональна схема деталізує взаємодію між рівнями розробки, даних і контролю та реалізує цикл “аналіз - виконання - контроль - корекція”.

На основі поєднання інформаційної моделі, структурної та функціональної схем сформовано цілісну архітектуру, що підтримує гнучке планування відповідно до політик, моніторинг ресурсів, керування дедлайнами та накопичення телеметричних даних. Розроблений фреймворк охоплює ініціалізацію середовища, виконання кроків бізнес-процесів із застосуванням різних політик диспетчеризації, збір телеметрії та формування аналітичних результатів. Центральним елементом системи є планувальник, який взаємодіє з політиками розподілу задач, механізмами алокації ресурсів і модулями моніторингу. Використання інверсії залежностей та уніфікованих інтерфейсів забезпечує розширюваність і адаптивність системи до змін навантаження.

У межах розділу також реалізовано графічний інтерфейс для керування експериментами, конфігурування моделей та візуалізації результатів, що дає змогу відстежувати динаміку виконання задач і аналізувати ефективність роботи системи за різних параметрів.

Запропонована система управління включає підсистеми контролю ресурсів і ЦПРС, обробки та підготовки даних, інтеграції та зберігання інформації. Її реалізація забезпечує повний життєвий цикл бізнес-процесів - від створення моделей і задання параметрів до моніторингу стану ресурсів та аналізу виконання. Підсистема моніторингу ЦПРС забезпечує виявлення відхилень від допустимих меж і ініціювання коригувальних дій, що сприяє підвищенню стійкості та передбачуваності роботи системи в умовах змінного навантаження.

Експериментальна оцінка показала, що запропонована конфігурація знижує частку порушень дедлайнів порівняно з базовою. За 20% ОП-інтенсивних кроків середня частка порушень зменшується з 14,56% до 5,48%, а за 50% ОП-інтенсивних кроків - з 67,88% до 25,81%, що відповідає покращенню у 2,66 та 2,63 рази відповідно. Отримані результати свідчать про зниження впливу дефіциту оперативної пам'яті та підвищення стабільності виконання кроків бізнес-процесів.

ВИСНОВКИ

В роботі вирішена задача щодо підвищення ефективності виконання кроків бізнес-процесів у платформах автоматизації проєктування та виконання бізнес-сервісів в умовах змінного навантаження. Наукові результати дослідження є внеском у розвиток методів, моделей та інформаційних технологій автоматизації проєктування та виконання бізнес-процесів у ресурсно-обмежених динамічних середовищах, зокрема в частині дедлайн-орієнтованого планування задач, багаторесурсного контролю запуску, моніторингу виконання та підтримання стабільності бізнес-сервісів за змінного навантаження.

На основі проведеного аналізу сучасних моделей і методів, що застосовуються у засобах автоматизації проєктування та виконання бізнес-сервісів, встановлено розрив між потребою стабільного й передбачуваного функціонування бізнес-сервісів та можливостями наявних методів до керованого виконання бізнес-процесів за змінного навантаження. Огляд наукових праць засвідчив недостатню опрацьованість питань моніторингу виконання кроків процесів, своєчасного виявлення погіршення показників функціонування та узгодженого планування задач з урахуванням поточного стану обчислювальних ресурсів у динамічних середовищах. Це ускладнює забезпечення прогнозованої продуктивності та дотримання заданих параметрів виконання. В дисертаційній роботі сформульовано наукове завдання розроблення моделі та методів керованого виконання бізнес-процесів, які поєднують засоби моделювання, моніторингу використання ресурсів і адаптивного планування задач. Таким чином, сформульоване в дисертації наукове завдання є актуальним і обґрунтованим з огляду на сучасний стан і тенденції розвитку засобів автоматизації проєктування та виконання бізнес-сервісів.

На основі результатів аналізу встановлено, що вперше розроблений метод реактивного мульти-токенового планування задач за дедлайном старту забезпечує зменшення частки порушень дедлайнів старту (ЦПРС) з 8,31% до 0,71%. При цьому середня продуктивність зберігається на рівні 8 задач/с, а механізм контрольованого допуску та резервування ресурсів знижує ризик перевантаження середовища, що

підтверджується меншим використанням ОП (середнє значення з 330,81 до 297,49 МБ, 95-й перцентиль: з 509,69 до 453,47 МБ) та більш керованою поведінкою системи під час піків (максимум виконаних задач за секунду: з 24 до 14 як наслідок обмеження запуску недопустимих задач).

На основі проведеного аналізу результатів експериментів удосконаленого методу планування задач за правилом "перший прийшов - перший оброблений" зазначений метод підвищує функціональну стійкість: у режимах дефіциту оперативної пам'яті він не допускає старт задач, що можуть спричинити перевищення ліміту ОП, і тим самим запобігає аварійному завершенню системи. Водночас виявлено недолік: за браку ОП виникає блокування голови черги, через що інші задачі не запускаються навіть за наявності вільних токенів ЦП, що призводить до простою ресурсів і зростання порушень термінів старту.

На основі аналізу результатів порівняльного експериментального дослідження встановлено, що модель моніторингу та виконання кроків бізнес-процесів у ресурсно-обмеженому середовищі дозволяє порівнювати методи впорядкування задач у межах єдиного механізму багаторесурсного допуску, оцінюючи вплив обраного методу на частку порушень дедлайнів старту за незмінного механізму перевірки можливості запуску задач за ресурсними обмеженнями.

У результаті порівняння базової та запропонованої конфігурації на промисловому середовищі показало зменшення частки порушених дедлайнів старту у сценаріях із 5000 задач. У першому сценарії цей показник знизився з 14,56% до 5,48%, тобто на 9,08%, у другому - з 67,88% до 25,81%, тобто на 42,07%. Найбільше зменшення частки порушень спостерігається за обмеженого обсягу оперативної пам'яті. Зі збільшенням доступної пам'яті рівень порушень дедлайнів зменшується в обох конфігураціях, однак запропонована конфігурація забезпечує нижчі значення частки порушених дедлайнів, що підтверджує доцільність використання запропонованої конфігурації.

Результати досліджень прийняті до впровадження у ТОВ "Інтехфорвард" та Конструкторському бюро інформаційних систем КПІ ім. Ігоря Сікорського.

Набула подальшого розвитку інформаційна технологія автоматизації проєктування та виконання бізнес-процесів у частині інтеграції засобів моделювання, виконання, моніторингу, аналітики та керування в єдиний замкнений контур підтримання стабільності за змінного навантаження. Практичним втіленням цього підходу є програмне забезпечення.

Розроблене програмне забезпечення дозволяє пропозити експериментальні дослідження виконання бізнес-процесів в умовах обмежених ресурсів. Реалізований функціонал охоплює моделювання різних сценаріїв навантаження, оцінювання стану ресурсів у процесі виконання задач, визначення частки порушених дедлайнів старту та порівняння методів планування. Інформаційна система підтримує збирання телеметрії, формування звітності та аналіз результатів, завдяки чому може бути використана як засіб тестування, діагностики й налаштування параметрів платформ автоматизації бізнес-процесів.

Мету дисертаційного дослідження, що полягає у підвищенні ефективності виконання кроків бізнес-процесів у платформах автоматизації проєктування та виконання бізнес-сервісів в умовах змінного навантаження шляхом розроблення та реалізації моделі моніторингу й ресурсно-орієнтованого планування, а також відповідних методів керування запуском задач, досягнуто; усі поставлені часткові наукові завдання вирішено в повному обсязі. Отримані результати становлять внесок у розвиток інформаційних технологій для системного аналізу та проєктування програмних платформ та інформаційних систем, зокрема в частині моніторингу, контролю ресурсів і планування виконання.

Перспективними напрямками подальших досліджень є удосконалення методів контролю ресурсів, розроблення засобів адаптивного планування кроків процесів та розширення моделей оцінювання ефективності виконання бізнес-процесів у змінних технічних та експлуатаційних умовах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Корнага Я. І., Лемешко В. А. Порівняння алгоритмів Modified First-Come-First-Served та Reactive Multi-Resource Token–Earliest Deadline First за умов обмеженого доступу до оперативної пам’яті // Телекомунікаційні та інформаційні технології. № 3 (2025). С. 26–33. ISSN 2412-4338. DOI: 10.31673/2412-4338.2025.038704.
2. Лемешко В. А. Формалізація процесу планування за методом RMRT-EDF з урахуванням обмежених ресурсів // Адаптивні системи автоматичного управління. Т. 2, № 47 (2025). С. 101–107. ISSN 1560-8956. DOI: 10.20535/1560-8956.47.2025.340183.
3. Пархомей І., Бойко Ю., Лемешко В. Математична модель та адаптивне управління алгоритмом обслуговування даних журналу в умовах змінного серверного навантаження // Вісник Хмельницького національного університету. Серія «Технічні науки». Т. 347, № 1 (2025). С.168–174. ISSN 2307-5732. DOI: 10.31891/2307-5732-2025-347-23.
4. Пархомей І., Бойко Ю., Лемешко В., Єрмоменко О. Оптимізація логування в інформаційних системах на основі математичної моделі управління буфером у шаблоні Producer–Consumer // Вимірювальна та обчислювальна техніка в технологічних процесах. № 2(2025). С. 218–225. ISSN 2219-9365. DOI: 10.31891/2219-9365-2025-82-30.
5. Пархомей І., Бойко Ю., Лемешко В. Алгоритм налаштування кількості потоків для виконання фонових задач // Вимірювальна та обчислювальна техніка в технологічних процесах. № 4 (2024). С. 162–173. ISSN 2219-9365. DOI: 10.31891/2219-9365-2024-80-20.
6. Rokis K., Kirikova M. Exploring Low-Code Development: A Comprehensive Literature Review // Complex Systems Informatics and Modeling Quarterly. 2023. No. 36. P. 68–86. DOI: 10.7250/csimq.2023-36.04.
7. Domański R., Wojciechowski H., Lewandowicz J., Hadaś Ł. Digitalization of Management Processes in Small and Medium-Sized Enterprises-An Overview of Low-Code and No-Code Platforms // Applied Sciences. 2023. Vol. 13, No. 24. Art. 13078.

DOI: 10.3390/app132413078.

8. Cigref. New Low Code/No Code development practices: Unlocking value by controlling risks [Electronic resource]. Paris : Cigref, 2023. URL: <https://www.cigref.fr/new-low-code-no-code-development-practices-unlocking-value-by-controlling-risks>.
9. KPMG International. Shaping digital transformation with low-code platforms [Electronic resource]. 2022. URL: https://assets.kpmg.com/content/dam/kpmg/cy/pdf/KPMG_Shaping%20digital%20transformation%20with%20low-code%20platforms_BF_sec_cy.pdf.
10. Aldridge V., David A., Shekhar S. Low-Code and No-Code BPM Tools Empowering Citizen Developers in Digital Transformation [Electronic resource]. 2023. URL: https://www.researchgate.net/publication/390175157_Low-Code_and_No-Code_BPM_Tools_Empowering_Citizen_Developers_in_Digital_Transformation.
11. Ouyang C., Dumas M., van der Aalst W. M. P., ter Hofstede A. H. M., Mendling J. From business process models to process-oriented software systems // ACM Transactions on Software Engineering and Methodology. 2009. Vol. 19, No. 1. Art. 2. P. 1–37. DOI: 10.1145/1555392.1555395.
12. Subbiah S. Low Code Development Platforms as the Future of Business Process Management [Electronic resource]. 2024. URL: https://www.researchgate.net/publication/378874963_LOW_CODE_DEVELOPMENT_PLATFORMS_AS_THE_FUTURE_OF_BUSINESS_PROCESS_MANAGEMENT.
13. Kokala A. Low-Code Platforms in BPM: How Workflow and Rules Engines Enable Citizen Development // International Journal of Science and Research Archive. 2022. Vol. 6, No. 1. P. 343–346. DOI: 10.30574/ijrsra.2022.6.1.0145.
14. Sivanarasu S. Empowering Citizen Developers: Evaluating the Usability of Low-Code BPM Tools in Process Design // European Journal of Computer Science and Information Technology. 2025. Vol. 13, No. 48. P. 220–233. DOI: 10.37745/ejcsit.2013/vol13n48220233.
15. Kalluri K. Revolutionizing BPM: The Role of Low-Code/No-Code Platforms in Accelerating Business Process Automation // International Journal for Multidisciplinary

- Research. 2024. Vol. 6, No. 6. DOI: 10.36948/ijfmr.2024.v06i06.23466.
16. Okeke H. E., Akinbolajo O. D. Integrating AI and automation into low-code development: Opportunities and challenges // *International Journal of Science and Research Archive*. 2023. Vol. 8, No. 1. P. 1094–1109. DOI: 10.30574/ijsra.2023.8.1.0077.
17. Desmond M., Duesterwald E., Isahagian V., Muthusamy V. A No-Code Low-Code Paradigm for Authoring Business Automations Using Natural Language [Electronic resource]. arXiv. 2022. arXiv:2207.10648. DOI: 10.48550/arXiv.2207.10648.
18. Duesterwald E., Huo S., Isahagian V., Jayaram K. R., Kumar R., Muthusamy V., Oum P., Saha D., Thomas G., Venkateswaran P. FLOW-BENCH: Towards Conversational Generation of Enterprise Workflows // *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*. Suzhou (China) : Association for Computational Linguistics, 2025. P. 1426–1436. DOI: 10.18653/v1/2025.emnlp-industry.100.
19. Alfonso I., Conrardy A., Cabot J. Towards the interoperability of low-code platforms [Electronic resource]. arXiv. 2024. arXiv:2412.05075. DOI: 10.48550/arXiv.2412.05075.
20. Bock A. C., Frank U. Low-Code Platform // *Business & Information Systems Engineering*. 2021. Vol. 63, No. 6. P. 733–740. DOI: 10.1007/s12599-021-00726-8.
21. Weerakkody V., Janssen M., Dwivedi Y. K. Transformational change and business process reengineering (BPR): Lessons from the British and Dutch public sector // *Government Information Quarterly*. 2011. Vol. 28, No. 3. P. 320–328. DOI: 10.1016/j.giq.2010.07.010.
22. Saldanha D. M. F., Dias C. N., Guillaumon S. Transparency and accountability in digital public services: Learning from the Brazilian cases // *Government Information Quarterly*. 2022. Vol. 39, No. 2. Art. 101680. DOI: 10.1016/j.giq.2022.101680.
23. Ziekow J. Administrative Procedures and Processes // *Public Administration in Germany* / eds. S. Kuhlmann, I. Proeller, D. Schimanke, J. Ziekow. Cham : Palgrave Macmillan, 2021. P. 163–183. DOI: 10.1007/978-3-030-53697-8_11.
24. Ajimati M. O., Carroll N., Maher M. Adoption of low-code and no-code

- development: A systematic literature review and future research agenda // *Journal of Systems and Software*. 2025. Vol. 222. Art. 112300. DOI: 10.1016/j.jss.2024.112300.
25. Weinzierl S., Zilker S., Dunzer S., Matzner M. Machine Learning in Business Process Management: A Systematic Literature Review // *Expert Systems with Applications*. 2024. Vol. 253. Art. 124181. DOI: 10.1016/j.eswa.2024.124181.
26. Ritter D. Compilation of BPMN-based Integration Flows // *Services and their Composition: Proceedings of the 7th Central European Workshop on Services and their Composition, ZEUS 2015, Jena, Germany, February 19–20, 2015* / eds. T. S. Heinze, T. M. Prinz. CEUR Workshop Proceedings. 2015. Vol. 1360. P. 1–9.
27. Kirikkayis Y., Gallik F., Winter M., Reichert M. BPMNE4IoT: A Framework for Modeling, Executing and Monitoring IoT-Driven Processes // *Future Internet*. 2023. Vol. 15, No. 3. Art. 90. DOI: 10.3390/fi15030090.
28. Zeisler G., Braunauer T. T., Fleischmann A., Singer R. A Step Towards a Universal Method for Modeling and Implementing Cross-Organizational Business Processes [Electronic resource]. arXiv. 2024. arXiv:2406.12302. DOI: 10.48550/arXiv.2406.12302.
29. Valderas P., Torres V., Pelochano V. Supporting a Hybrid Composition of Microservices. The EUCalipTool Platform // *Journal of Software Engineering Research and Development*. 2020. Vol. 8. P. 1:1–1:14. DOI: 10.5753/jserd.2020.457.
30. Castro L. F. S., Rigo S. Relating Edge Computing and Microservices by Means of Architecture Approaches and Features, Orchestration, Choreography, and Offloading: A Systematic Literature Review [Electronic resource]. arXiv. 2023. arXiv:2301.07803. DOI: 10.48550/arXiv.2301.07803.
31. Ortiz J., Torres V., Valderas P. Microservice compositions based on the choreography of BPMN fragments: facing evolution issues // *Computing*. 2023. Vol. 105, No. 2. P. 375–416. DOI: 10.1007/s00607-022-01128-8.
32. de Vries S., Blaauw F., Andrikopoulos V. Cost-Profiling Microservice Applications Using an APM Stack // *Future Internet*. 2023. Vol. 15, No. 1. Art. 37. DOI: 10.3390/fi15010037.
33. Tupsakhare P. Ensuring Scalability and Stability in Low-Code Platforms: Best

- Practices and Challenges // Journal of Scientific and Engineering Research. 2022. Vol. 9, No. 4. P. 209–214. DOI: 10.5281/zenodo.13918722.
34. Tadi S. R. C. C. T. Scalability and Performance Benchmarking of Low-Code Platforms vs. Traditional Development in Large-Scale Enterprise Applications // Journal of Scientific and Engineering Research. 2021. Vol. 8, No. 8. P. 262–273. DOI: 10.5281/zenodo.15044772.
35. Aveiro D., Freitas V., Cunha E., Quintal F., Almeida Y. Traditional vs. low-code development: comparing needed effort and system complexity in the NexusBRaNT experiment // 2023 IEEE 25th Conference on Business Informatics (CBI). 2023. P. 1–10. DOI: 10.1109/CBI58679.2023.10187470.
36. Sadovnikov K., Sweijen R., van der Werf J. M., van de Weerd I. A Framework to Assess the Suitability of Low-Code for BPM // ECIS 2023 Research Papers. 2023. Paper 433.
37. del-Río-Ortega A., Gutiérrez A. M., Durán A., Resinas M., Ruiz-Cortés A. Modelling Service Level Agreements for Business Process Outsourcing Services // Advanced Information Systems Engineering : 27th International Conference, CAiSE 2015, Stockholm, Sweden, June 8–12, 2015. Proceedings / ed. by J. Zdravkovic, M. Kirikova, P. Johannesson. Cham : Springer, 2015. P. 485–500. (Lecture Notes in Computer Science; vol. 9097). DOI: 10.1007/978-3-319-19069-3_30.
38. Farshidi S., Jansen S., Fortuin S. Model-driven development platform selection: four industry case studies // Software and Systems Modeling. 2021. Vol. 20, No. 5. P. 1525–1551. DOI: 10.1007/s10270-020-00855-w.
39. Ouyang C., Wynn M. T., Fidge C., ter Hofstede A. H. M., Kuhr J.-C. Modelling Complex Resource Requirements in Business Process Management Systems // ACIS 2010 Proceedings : 21st Australasian Conference on Information Systems / ed. by M. Rosemann, P. Green, F. Rohde. Brisbane : AIS Library, 2010. Paper 60.
40. Vasilecas O., Laureckas E., Rima A. Analysis of Using Resources in Business Process Modeling and Simulation // Applied Computer Systems. 2014. Vol. 16, No. 1. P. 19–25. DOI: 10.1515/acss-2014-0009.
41. Comuzzi M., Spanoudakis G. A Framework for Hierarchical and Recursive

- Monitoring of Service Based Systems // Fourth International Conference on Internet and Web Applications and Services (ICIW 2009) / eds. H. Sasaki, G. O. Bellot, M. Ehmann, O. Dini. Venice/Mestre, Italy, 24–28 May 2009. P. 383–388. IEEE, 2009. DOI: 10.1109/ICIW.2009.63.
42. Ihde S., Pufahl L., Völker M., Goel A., Weske M. A framework for modeling and executing task-Specific resource allocations in business processes // *Computing*. 2022. Vol. 104. P. 2405–2429. DOI: 10.1007/s00607-022-01093-2.
43. Нікіфорова Л., Дьогтева І., Платонов О., Шиян А. Модель бізнес-процесу для цифрової трансформації державної атестації наукових установ та закладів вищої освіти // *Відкрита наука та інновації*. 2024. Т. 1, № 2. С. 64–88. DOI: 10.62405/osi.2024.02.06.
44. Рожко В. І., Хлистун А. А. Моделювання бізнес-процесів у сучасному бізнес-середовищі // *Проблеми економіки*. 2023. № 4. С. 159–164. DOI: 10.32983/2222-0712-2023-4-159-164.
45. Юрчук Н. П., Міронова Ю. В. Розробка інформаційних систем для цифрового менеджменту: виклики та перспективи // *Ефективна економіка*. 2024. № 12. DOI: 10.32702/2307-2105.2024.12.51.
46. Kedziora D., Aunimo L., Kortessalmi H. Between No-Code, Low-Code, Custom-Code, and AI. Hyperautomation Strategies at Nordic Enterprises // *Digital Interaction and Machine Intelligence. MIDI 2023* / eds. C. Biele, J. Kacprzyk, W. Kopeć, J. Możaryn, J. W. Owsiniński, A. Romanowski, M. Sikorski. Cham : Springer, 2024. P. 172–179. (Lecture Notes in Networks and Systems; vol. 1076). DOI: 10.1007/978-3-031-66594-3_18.
47. Suresh V., Ezhilchelvan P., Watson P. Scalable and Responsive Event Processing in the Cloud // *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*. 2013. Vol. 371, No. 1983. Art. 20120095. DOI: 10.1098/rsta.2012.0095.
48. Bejarano J., Barón D., González-Rojas O., Camargo M. Discovering Optimal Resource Allocations for What-If Scenarios Using Data-Driven Simulation // *Frontiers in Computer Science*. 2023. Vol. 5. Art. 1279800. DOI: 10.3389/fcomp.2023.1279800.

49. Xu J., Liu C., Zhao X. Resource Allocation vs. Business Process Improvement: How They Impact on Each Other // Business Process Management. BPM 2008 / eds. M. Dumas, M. Reichert, M.-C. Shan. Berlin; Heidelberg : Springer, 2008. P. 228–243. (Lecture Notes in Computer Science; vol. 5240). DOI: 10.1007/978-3-540-85758-7_18.
50. Stroppi L. J. R., Chiotti O., Villarreal P. D. Defining the resource perspective in the development of processes-aware information systems // Information and Software Technology. 2015. Vol. 59. P. 86–108. DOI: 10.1016/j.infsof.2014.10.006.
51. Hoenisch P., Schulte S., Dustdar S., Venugopal S. Self-Adaptive Resource Allocation for Elastic Process Execution // 2013 IEEE Sixth International Conference on Cloud Computing (CLOUD). 2013. P. 220–227. DOI: 10.1109/CLOUD.2013.126.
52. Euting S., Janiesch C., Fischer R., Tai S., Weber I. Scalable Business Process Execution in the Cloud // 2014 IEEE International Conference on Cloud Engineering (IC2E). Boston, MA, USA, 2014. P. 175–184. DOI: 10.1109/IC2E.2014.13.
53. Kritikos K., Zeginis C., Iranzo J., Gonzalez R. S., Seybold D., Griesinger F., Domaschka J. Multi-cloud provisioning of business processes // Journal of Cloud Computing. 2019. Vol. 8. Article 18. DOI: 10.1186/s13677-019-0143-x.
54. Razian M., Fathian M., Bahsoon R., Toosi A. N., Buyya R. Service composition in dynamic environments: A systematic review and future directions // Journal of Systems and Software. 2022. Vol. 188. Art. 111290. DOI: 10.1016/j.jss.2022.111290.
55. Schulte S., Janiesch C., Venugopal S., Weber I., Hoenisch P. Elastic Business Process Management: State of the art and open challenges for BPM in the cloud // Future Generation Computer Systems. 2015. Vol. 46. P. 36–50. DOI: 10.1016/j.future.2014.09.005.
56. Jakoubi S., Goluch G., Tjoa S., Quirchmayr G. Deriving Resource Requirements Applying Risk-Aware Business Process Modeling and Simulation // 16th European Conference on Information Systems (ECIS 2008). Galway, Ireland, 2008. P. 1542–1554.
57. Borges E. S., Fantinato M., Aksu Ü., Reijers H. A., Thom L. H. Monitoring of Non-functional Requirements of Business Processes based on Quality of Service Attributes of Web Services // Proceedings of the 21st International Conference on Enterprise

- Information Systems (ICEIS 2019), Heraklion, Crete, Greece, May 3–5, 2019. Setúbal : SciTePress, 2019. P. 588–595. DOI: 10.5220/0007700605880595.
58. van der Aalst W. M. P. Business Process Management: A Comprehensive Survey // ISRN Software Engineering. 2013. Vol. 2013. Art. 507984. 37 p. DOI: 10.1155/2013/507984.
59. Leymann F., Karastoyanova D., Papazoglou M. P. Business Process Management Standards // Handbook on Business Process Management 1: Introduction, Methods, and Information Systems / eds. J. vom Brocke, M. Rosemann. Berlin; Heidelberg : Springer, 2010. P. 513–542. DOI: 10.1007/978-3-642-00416-2_24.
60. Represa J. G., Larrinaga F., Varga P., Ochoa W., Perez A., Kozma D., Delsing J. Investigation of Microservice-Based Workflow Management Solutions for Industrial Automation // Applied Sciences. 2023. Vol. 13, No. 3. Art. 1835. DOI: 10.3390/app13031835.
61. Rudrabhatla C. K. Comparison of Event Choreography and Orchestration Techniques in Microservice Architecture // International Journal of Advanced Computer Science and Applications. 2018. Vol. 9, No. 8. P. 18–22. DOI: 10.14569/IJACSA.2018.090804.
62. Laigner R., Almeida A. C., Assunção W. K. G., Zhou Y. An Empirical Study on Challenges of Event Management in Microservice Architectures // ACM Transactions on Software Engineering and Methodology. 2025. DOI: 10.1145/3776581.
63. Hens P., Snoeck M., De Backer M., Poels G. Transforming Standard Process Models to Decentralized Autonomous Entities // CEUR Workshop Proceedings. 2010. Vol. 662. P. 97–106.
64. Valderas P., Torres V., Serral E. Modelling and executing IoT-enhanced business processes through BPMN and microservices // Journal of Systems and Software. 2022. Vol. 184. Art. 111139. DOI: 10.1016/j.jss.2021.111139.
65. Drave I., Michael J., Müller E., Rumpe B., Varga S. Model-Driven Engineering of Process-Aware Information Systems // SN Computer Science. 2022. Vol. 3, No. 6. Art. 479. DOI: 10.1007/s42979-022-01334-3.
66. Bousdekis A., Kerasiotis A., Kotsias S., Theodoropoulou G., Miaoulis G.,

- Ghazanfarpour D. Modelling and Predictive Monitoring of Business Processes under Uncertainty with Reinforcement Learning // *Sensors*. 2023. Vol. 23, No. 15. Art. 6931. DOI: 10.3390/s23156931.
67. Zhang Z., Guo M., Guo T. A Scientific Workflow Engine Design and Implementation // *Proceedings of the 4th International Conference on Public Management and Intelligent Society, PMIS 2024*, 15–17 March 2024, Changsha, China. 2024. DOI: 10.4108/eai.15-3-2024.2346547.
68. Nadeem A., Malik M. Z. A Case for Microservices Orchestration Using Workflow Engines // *Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER '22)*. 2022. P. 6–10. DOI: 10.1145/3510455.3512777.
69. Silva M., Santos L., Soares M., Rocha F. Guidelines for the Application of Event Driven Architecture in Micro Services with High Volume of Data // *Proceedings of the 27th International Conference on Enterprise Information Systems - Volume 2: ICEIS*. 2025. P. 859–866. DOI: 10.5220/0013348600003929.
70. Blal R., Leshob A., Mili H., Benzarti I., Hadaya P., Rab R. SOA Services Identification and Design Methods from Business Models: A Systematic Literature Review // *IEEE Access*. 2025. Vol. 13. P. 9879–9901. DOI: 10.1109/ACCESS.2025.3528297.
71. Calinescu R., Weyns D., Gerasimou S., Iftikhar M. U., Habli I., Kelly T. Engineering Trustworthy Self-Adaptive Software with Dynamic Assurance Cases // *IEEE Transactions on Software Engineering*. 2018. Vol. 44, No. 11. P. 1039–1069. DOI: 10.1109/TSE.2017.2738640.
72. Pourmajidi W., Zhang L., Steinbacher J., Erwin T., Miranskyy A. A Reference Architecture for Governance of Cloud Native Applications [Electronic resource]. arXiv. 2023. arXiv:2302.11617. DOI: 10.48550/arXiv.2302.11617.
73. El-Gharib N. M., Amyot D. Data Preprocessing Method and API for Mining Processes from Cloud-Based Application Event Logs // *Algorithms*. 2022. Vol. 15, No. 6. Art. 180. DOI: 10.3390/a15060180.
74. Usman M., Ferlin S., Brunstrom A., Taheri J. A Survey on Observability of

Distributed Edge & Container-Based Microservices // IEEE Access. 2022. Vol. 10. P. 86904–86919. DOI: 10.1109/ACCESS.2022.3193102.

75. Solano G. F., Caldas R. D., Rodrigues G. N., Vogel T., Pelliccione P. Taming Uncertainty in the Assurance Process of Self-Adaptive Systems: a Goal-Oriented Approach // 2019 IEEE/ACM 14th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS). 2019. P. 89–99. DOI: 10.1109/SEAMS.2019.00020.

76. Cloudera. Managing Apache Impala. Cloudera Runtime 7.2.16. Admission Control and Query Queuing [Electronic resource]. 2020. URL: <https://docs.cloudera.com/runtime/7.2.16/impala-manage/impala-manage.pdf>.

77. Martinez Suñé A. E., Lopez Pombo C. G. Probabilistic Quality of Service aware Service Selection [Electronic resource]. arXiv. 2022. arXiv:2205.09877. DOI: 10.48550/arXiv.2205.09877.

78. Pufahl L., Stiehle F., Ihde S., Weske M., Weber I. Resource allocation in business process executions—A systematic literature study // Information Systems. 2025. Vol. 132. Art. 102541. DOI: 10.1016/j.is.2025.102541.

79. Kratsch W., Manderscheid J., Röglinger M., Seyfried J. Machine Learning in Business Process Monitoring: A Comparison of Deep Learning and Classical Approaches Used for Outcome Prediction // Business & Information Systems Engineering. 2021. Vol. 63, No. 3. P. 261–276. DOI: 10.1007/s12599-020-00645-0.

80. Dumas M., Fournier F., Limonad L. et al. AI-Augmented Business Process Management Systems: A Research Manifesto // ACM Transactions on Management Information Systems. 2023. Vol. 14, No. 1. Art. 11. P. 1–19. DOI: 10.1145/3576047.

81. Saha A., Agarwal P., Ghosh S., Gantayat N., Sindhgatta R. Towards Business Process Observability // Proceedings of the 7th Joint International Conference on Data Science & Management of Data (11th ACM IKDD CODS and 29th COMAD). Bangalore, India, 2024. P. 257–265. DOI: 10.1145/3632410.3632435.

82. Middelhuis J., Lo Bianco R., Sherzer E., Bukhsh Z., Adan I., Dijkman R. Learning policies for resource allocation in business processes // Information Systems. 2025. Vol. 128. Art. 102492. DOI: 10.1016/j.is.2024.102492.

83. Keller A., Ludwig H. The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services // *Journal of Network and Systems Management*. 2003. Vol. 11. P. 57–81. DOI: 10.1023/A:1022445108617.
84. Bucaioni A., Cicchetti A., Ciccozzi F. Modelling in low-code development: a multi-vocal systematic review // *Software and Systems Modeling*. 2022. Vol. 21, No. 5. P. 1959–1981. DOI: 10.1007/s10270-021-00964-0.
85. Pautasso C. A Flexible System for Visual Service Composition : doctoral thesis. Zürich : ETH Zurich, 2004. DOI: 10.3929/ethz-a-004860667.
86. Viljoen A., Stelzl B., Yang M., Nguyen J., Hein A., Elshan E., Krcmar H. Navigating Flexibility and Standardisation in Low-Code/No-Code Development // *Information Systems Journal*. 2026. Vol. 36, No. 1. P. 95–109. DOI: 10.1111/isj.70001.
87. Naqvi B., Kedziora D., Zhang L., Oyedeki S. Quality of Low-Code/No-Code Development Platforms Through the Lens of ISO 25010:2023 // *Computer*. 2025. Vol. 58, No. 3. P. 30–40. DOI: 10.1109/MC.2024.3493192.
88. de Vrieze P., Xu L. Resilience analysis of service-oriented collaboration process management systems // *Service Oriented Computing and Applications*. 2018. Vol. 12, No. 1. P. 25–39. DOI: 10.1007/s11761-018-0233-5.
89. Comuzzi M., Vonk J., Grefen P. W. P. J. Measures and mechanisms for process monitoring in evolving business networks // *Data & Knowledge Engineering*. 2012. Vol. 71, No. 1. P. 1–28. DOI: 10.1016/j.datak.2011.07.004.
90. Oletsky O., Moholivskyi V. On Supervising and Coordinating Microservices Within Web Applications on the Basis of State Machines // *Information Technology and Implementation (IT&I 2024): Selected Papers of the XI International Scientific Conference, Kyiv, Ukraine, November 20–21, 2024. CEUR Workshop Proceedings*. 2025. Vol. 3909. P. 442–454.
91. Tzanettis I., Androna C.-M., Zafeiropoulos A., Fotopoulou E., Papavassiliou S. Data Fusion of Observability Signals for Assisting Orchestration of Distributed Applications // *Sensors*. 2022. Vol. 22, No. 5. Art. 2061. DOI: 10.3390/s22052061.
92. Mendonça N. C., Jamshidi P., Garlan D., Pahl C. Developing Self-Adaptive Microservice Systems: Challenges and Directions // *IEEE Software*. 2021. Vol. 38, No.

2. P. 70–79. DOI: 10.1109/MS.2019.2955937.
93. Qazi F., Kwak D., Khan F. G., Ali F., Khan S. U. Service Level Agreement in Cloud Computing: Taxonomy, Prospects, and Challenges // *Internet of Things*. 2024. Vol. 25. Art. 101126. DOI: 10.1016/j.iot.2024.101126.
94. Sharma Y., Bhamare D., Sastry N., Javadi B., Buyya R. SLA Management in Intent-Driven Service Management Systems: A Taxonomy and Future Directions // *ACM Computing Surveys*. 2023. Vol. 55, No. 13s. Art. 292. P. 1–38. DOI: 10.1145/3589339.
95. Márquez-Chamorro A. E., Resinas M., Ruiz-Cortés A. Predictive Monitoring of Business Processes: A Survey // *IEEE Transactions on Services Computing*. 2018. Vol. 11, No. 6. P. 962–977. DOI: 10.1109/TSC.2017.2772256.
96. Janetzky F., Wallin M. Low-code Performance Evaluation and its Impact on Software Development : bachelor thesis. KTH Royal Institute of Technology, 2024. 36 p. (TRITA-EECS-EX; 2024:14). URL: <https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-345165>.
97. Dustdar S., Guo Y., Satzger B., Truong H. L. Principles of Elastic Processes // *IEEE Internet Computing*. 2011. Vol. 15, No. 5. P. 66–71. DOI: 10.1109/MIC.2011.121.
98. Wang Z., Li H., Sun L., Rosenkrantz S., Che H., Jiang H. A Tail Latency SLO Guaranteed Task Scheduling Scheme for User-Facing Services // *IEEE Transactions on Parallel and Distributed Systems*. 2025. Vol. 36, No. 4. P. 759–774. DOI: 10.1109/TPDS.2025.3542638.
99. Welsh M., Culler D. Adaptive Overload Control for Busy Internet Servers // *Proceedings of the 4th USENIX Symposium on Internet Technologies and Systems (USITS '03)*. Seattle, WA, USA, 2003. P. 43–57.
100. Delimitrou C., Kozyrakis C. Quasar: Resource-Efficient and QoS-Aware Cluster Management // *Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '14)*. 2014. P. 127–144. DOI: 10.1145/2541940.2541941.
101. Zhang L., Zhang Y., Jamshidi P., Xu L., Pahl C. Service workload patterns for QoS-driven cloud resource management // *Journal of Cloud Computing*. 2015. Vol. 4. Art. 23. DOI: 10.1186/s13677-015-0048-2.

102. Lim N., Majumdar S., Ashwood-Smith P. A resource management technique for processing deadline-constrained multi-stage workflows // *Journal of Cloud Computing: Advances, Systems and Applications*. 2017. Vol. 6. Art. 21. DOI: 10.1186/s13677-017-0091-2.
103. Hummer W., Rosenberg F., Oliveira F., Eilam T. Testing Idempotence for Infrastructure as Code // *Middleware 2013. Lecture Notes in Computer Science*. Vol. 8275. Springer, Berlin, Heidelberg, 2013. P. 368–388. DOI: 10.1007/978-3-642-45065-5_19.
104. Zhou H., Huang T., Deng H. et al. WeChat Overload: An Overload Control System for Microservices // *Proceedings of the 2018 ACM Symposium on Cloud Computing (SoCC '18)*. 2018. P. 149–161. DOI: 10.1145/3267809.3267810.
105. Zhang Y., Zhou Z., Elnikety S. et al. Ursa: Analytically-Driven Resource Management for Cloud-Native Microservices // *2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. 2024. P. 954–969. DOI: 10.1109/HPCA57654.2024.00077.
106. Sachidananda V., Sivaraman A. Collective Autoscaling for Cloud Microservices [Electronic resource]. arXiv. 2021. arXiv:2112.14845. DOI: 10.48550/arXiv.2112.14845.
107. Hansen M., Hasselbring W. Instrumentation of Software Systems with OpenTelemetry for Software Visualization // *Softwaretechnik-Trends*. 2025. Vol. 45, No. 1. P. 17–19.
108. Chen Z., Jiang Z., Su Y., Lyu M. R., Zheng Z. TraceMesh: Scalable and Streaming Sampling for Distributed Traces // *Proceedings - 2024 IEEE 17th International Conference on Cloud Computing (CLOUD 2024)*. 2024. P. 54–65. DOI: 10.1109/CLOUD62652.2024.00016.
109. Chen Z., Pu J., Zheng Z. Tracezip: Efficient Distributed Tracing via Trace Compression // *Proceedings of the ACM on Software Engineering*. 2025. Vol. 2, No. ISSTA. Art. ISSTA019. P. 411–433. DOI: 10.1145/3728888.
110. Kephart J. O., Chess D. M. The Vision of Autonomic Computing // *Computer*. 2003. Vol. 36, No. 1. P. 41–50. DOI: 10.1109/MC.2003.1160055.

111. Kephart J. O. Research Challenges of Autonomic Computing // Proceedings of the 27th International Conference on Software Engineering (ICSE '05). 2005. P. 15–22. DOI: 10.1145/1062455.1062464.
112. Xing T., Xiong C., Ye C., Wei Q., Picorel J., Barbalace A. Maximizing VMs' IO Performance on Overcommitted CPUs with Fairness // Proceedings of the 2023 ACM Symposium on Cloud Computing. 2023. P. 93–108. DOI: 10.1145/3620678.3624649.
113. Rabenstein B., Volz J. Prometheus: A Next-Generation Monitoring System [Electronic resource] : presentation. SREcon15 Europe, Dublin, Ireland, 14 May 2015. URL:
<https://www.usenix.org/conference/srecon15europe/program/presentation/rabenstein>.
114. Alqahtani A., Li Y., Patel P., Solaiman E., Ranjan R. End-to-End Service Level Agreement Specification for IoT Applications // 2018 International Conference on High Performance Computing & Simulation (HPCS). 2018. P. 926–935. DOI: 10.1109/HPCS.2018.00147.
115. Punniyamoorthy V., Kumar B., Saha S., Butra L., Palanigounder M., Agarwal A. K., Kannan K. An SLO Driven and Cost-Aware Autoscaling Framework for Kubernetes [Electronic resource]. arXiv. 2025. arXiv:2512.23415. DOI: 10.48550/arXiv.2512.23415.
116. Kristianto H., Zahra A. Performance Analysis of Choreography and Orchestration in Microservices Architecture // Journal of Theoretical and Applied Information Technology. 2021. Vol. 99, No. 18. P. 4220–4230. URL:
<https://www.jatit.org/volumes/Vol99No18/4Vol99No18.pdf>.
117. Zehra S., Syed H. J., Faseeha U. FedMon: Federated eBPF Monitoring for Distributed Anomaly Detection in Multi-Cluster Cloud Environments [Electronic resource]. arXiv. 2025. arXiv:2510.10126. 7 p. DOI: 10.48550/arXiv.2510.10126.
118. Guidi C. Towards Self-Architecting Autonomic Microservices // Joint Post-proceedings of the Third and Fourth International Conference on Microservices (Microservices 2020/2022). Open Access Series in Informatics (OASICs). 2023. Vol. 111. P. 7:1–7:14. DOI: 10.4230/OASICs.Microservices.2020-2022.7.
119. Mathew A., Andrikopoulos V., Blaauw F. J., Karastoyanova D. Pattern-based

- serverless data processing pipelines for Function-as-a-Service orchestration systems // Future Generation Computer Systems. 2024. Vol. 154. P. 87–100. DOI: 10.1016/j.future.2023.12.026.
120. Maharjan R., Chy M. S. H., Arju M. A., Cerny T. Benchmarking Message Queues // Telecom. 2023. Vol. 4, No. 2. P. 298–312. DOI: 10.3390/telecom4020018.
121. Jani Y. Unified Monitoring for Microservices: Implementing Prometheus and Grafana for Scalable Solutions // Journal of Artificial Intelligence, Machine Learning and Data Science. 2024. Vol. 2, No. 1. P. 848–852. DOI: 10.51219/JAIMLD/yash-jani/206.
122. Shkuro Y., Renard B., Singh A. Positional Paper: Schema-First Application Telemetry // ACM SIGOPS Operating Systems Review. 2022. Vol. 56, No. 1. P. 8–17. DOI: 10.1145/3544497.3544500.
123. Kleinrock L. Queueing Systems. Volume I: Theory. New York : Wiley, 1975. 448 p. ISBN 0471491101.
124. Cai Z., Blackburn S. M., Bond M. D., Maas M. Distilling the Real Cost of Production Garbage Collectors // 2022 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS). 2022. P. 46–57. DOI: 10.1109/ISPASS55109.2022.00005.
125. Microsoft. Garbage collection and performance [Electronic resource] // Microsoft Learn. 2022. URL: <https://learn.microsoft.com/en-us/dotnet/standard/garbage-collection/performance>.
126. Hakimzadeh K., Dowling J. Ops-Scale: Scalable and Elastic Cloud Operations by a Functional Abstraction and Feedback Loops // 2019 IEEE 13th International Conference on Self-Adaptive and Self-Organizing Systems (SASO). 2019. P. 62–71. DOI: 10.1109/SASO.2019.00017.
127. Maggi F. M., Di Francescomarino C., Dumas M., Ghidini C. Predictive Monitoring of Business Processes [Electronic resource]. arXiv. 2013. arXiv:1312.4874. DOI: 10.48550/arXiv.1312.4874.
128. Hoenisch P., Schulte S., Dustdar S. Workflow Scheduling and Resource Allocation for Cloud-Based Execution of Elastic Processes // 2013 IEEE 6th International

- Conference on Service-Oriented Computing and Applications. 2013. DOI: 10.1109/SOCA.2013.44.
129. Park G., Song M. Optimizing Resource Allocation Based on Predictive Process Monitoring // IEEE Access. 2023. Vol. 11. P. 38309–38323. DOI: 10.1109/ACCESS.2023.3267538.
130. Object Management Group. Business Process Model and Notation (BPMN). Version 2.0 [Electronic resource]. OMG Document Number: formal/11-01-03. 2010. URL: <https://www.omg.org/spec/BPMN/2.0/PDF>.
131. Liu C. L., Layland J. W. Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment // Journal of the ACM. 1973. Vol. 20, No. 1. P. 46–61. DOI: 10.1145/321738.321743.
132. Holenderski M. J., Bril R. J., Lukkien J. J. Parallel-Task Scheduling on Multiple Resources // Proceedings of the 24th Euromicro Conference on Real-Time Systems (ECRTS 2012). 2012. P. 233–244. DOI: 10.1109/ECRTS.2012.33.
133. Inam R., Mahmud N., Behnam M., Nolte T., Sjödin M. The Multi-Resource Server for predictable execution on multi-core platforms // Proceedings of the 20th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2014). 2014. P. 1–12. DOI: 10.1109/RTAS.2014.6925986.
134. Arias M., Saavedra R., Marques M. R., Muñoz Gama J., Sepúlveda M. Human resource allocation in business process management and process mining: a systematic mapping study // Management Decision. 2018. Vol. 56, No. 2. P. 376–405. DOI: 10.1108/MD-05-2017-0476.
135. Ihde S., Pufahl L., Goel A., Weske M. Towards Dynamic Resource Management in Business Processes // Proceedings of the 11th ZEUS Workshop (ZEUS 2019). CEUR Workshop Proceedings. 2019. Vol. 2339. P. 17–23. URL: <https://ceur-ws.org/Vol-2339/paper4.pdf>.
136. Senkul P., Kifer M., Toroslu I. H. A Logical Framework for Scheduling Workflows under Resource Allocation Constraints // Proceedings of the 28th International Conference on Very Large Data Bases (VLDB 2002). 2002. P. 694–705. DOI: 10.1016/B978-155860869-6/50067-6.

137. Liu T., Cheng Y., Ni Z. Mining Event Logs to Support Workflow Resource Allocation // *Knowledge-Based Systems*. 2012. Vol. 35. P. 320–331. DOI: 10.1016/j.knosys.2012.05.010.
138. Ghodsi A., Zaharia M., Hindman B., Konwinski A., Shenker S., Stoica I. Dominant Resource Fairness: Fair Allocation of Multiple Resource Types // *Proceedings of the 8th USENIX Symposium on Networked Systems Design and Implementation (NSDI 11)*. 2011. P. 323–336.
139. Grandl R., Ananthanarayanan G., Kandula S., Rao S., Akella A. Multi-Resource Packing for Cluster Schedulers // *Proceedings of the 2014 ACM Conference on SIGCOMM*. 2014. P. 455–466. DOI: 10.1145/2619239.2626334.
140. Rosenthal K., Ternes B., Strecker S. Business Process Simulation on Procedural Graphical Process Models // *Business & Information Systems Engineering*. 2021. Vol. 63, No. 5. P. 569–602. DOI: 10.1007/s12599-021-00690-3.
141. Hlupic V., de Vreede G. J. Business process modelling using discrete-event simulation: current opportunities and future challenges // *International Journal of Simulation and Process Modelling*. 2005. Vol. 1, No. 1–2. P. 72–81. DOI: 10.1504/IJSPM.2005.007115.
142. Onggo S., Proudlove N., D’Ambrogio A., Calabrese A., Bisogno S., Levialdi Ghiron N. A BPMN extension to support discrete-event simulation for healthcare applications: an explicit representation of queues, attributes and data-driven decision points // *Journal of the Operational Research Society*. 2018. Vol. 69, No. 5. P. 788–802. DOI: 10.1057/s41274-017-0267-7.
143. El Kassis M., Troussel F., Zacharewicz G., Daclin N. Bridging the gap between business process and simulation: transformation from BPMN to DEVS // *IFAC-PapersOnLine*. 2023. Vol. 56, No. 2. P. 11888–11893. DOI: 10.1016/j.ifacol.2023.10.601.
144. Afifi N., Awad A., Abdelsalam H. RBPSim: A Resource-aware Extension of BPSim Using Workflow Resource Patterns // *Proceedings of the 10th Central European Workshop on Services and their Composition (ZEUS 2018)*. Dresden, Germany, February 8–9, 2018. *CEUR Workshop Proceedings*. 2018. Vol. 2072. P. 32–39. URL:

<https://ceur-ws.org/Vol-2072/paper6.pdf>.

145. Havur G., Cabanillas C., Mendling J., Polleres A. Resource Allocation with Dependencies in Business Process Management Systems // Business Process Management Forum. 2016. Lecture Notes in Business Information Processing. Vol. 260. P. 3–19. DOI: 10.1007/978-3-319-45468-9_1.
146. Cabanillas C., García J. M., Resinas M., Ruiz D., Mendling J., Ruiz-Cortés A. Priority-Based Human Resource Allocation in Business Processes // Service-Oriented Computing. ICSOC 2013. Lecture Notes in Computer Science. 2013. Vol. 8274. P. 374–388. DOI: 10.1007/978-3-642-45005-1_26.
147. Audsley N. C., Burns A., Richardson M. F., Wellings A. J. Hard real-time scheduling: the deadline-monotonic approach // IFAC Proceedings Volumes. 1991. Vol. 24, No. 2. P. 127–132. DOI: 10.1016/S1474-6670(17)51283-5.
148. Bonifaci V., Chan H.-L., Marchetti-Spaccamela A., Megow N. Algorithms and complexity for periodic real-time scheduling // ACM Transactions on Algorithms. 2012. Vol. 9, No. 1. Art. 6. P. 1–19. DOI: 10.1145/2390176.2390182.
149. Rahimikhanghah A., Tajkey M., Rezazadeh B., Rahmani A. M. Resource scheduling methods in cloud and fog computing environments: a systematic literature review // Cluster Computing. 2022. Vol. 25, No. 2. P. 911–945. DOI: 10.1007/s10586-021-03467-1.
150. Saxena V., Kumar A., Danwa V. A Review of Cloud Computing CPU Resource Optimization: Methods, Difficulties, and Prospects // ICIMMI '24: Proceedings of the 6th International Conference on Information Management & Machine Intelligence. New York, NY, USA : Association for Computing Machinery, 2025. Art. 26. P. 1–5. DOI: 10.1145/3745812.3745839.
151. Mutichiro B., Tran M.-N., Kim Y.-H. QoS-Based Service-Time Scheduling in the IoT-Edge Cloud // Sensors. 2021. Vol. 21, No. 17. Art. 5797. DOI: 10.3390/s21175797.
152. Al-Arasi R. A., Saif A. Task Scheduling in Cloud Computing Based on Meta-Heuristic Techniques: A Review Paper // EAI Endorsed Transactions on Cloud Systems. 2020. Vol. 6, No. 17. Art. e4. DOI: 10.4108/eai.13-7-2018.162829.
153. Минайленко Р. М. Аналіз алгоритмів планування ресурсами в розподіленому

- обчислювальному середовищі // Конструювання, виробництво та експлуатація сільськогосподарських машин. 2020. Вип. 50. С. 229–235. DOI: 10.32515/2414-3820.2020.50.229-235.
154. Savvas I. K., Kechadi M. Tahar. Dynamic task scheduling in computing cluster environments // Third International Symposium on Parallel and Distributed Computing / Third International Workshop on Algorithms, Models and Tools for Parallel Computing on Heterogeneous Networks. 2004. P. 372–379. DOI: 10.1109/ISPDC.2004.21.
155. Abduljabbar M., Eljammaly M., Pericas M. Mitigating inefficient task mappings with an Adaptive Resource-Moldable Scheduler (ARMS) [Electronic resource]. arXiv. 2021. arXiv:2112.09509 [cs.DC]. DOI: 10.48550/arXiv.2112.09509.
156. Kopanski J., Rządca K. Plan-Based Job Scheduling for Supercomputers with Shared Burst Buffers [Electronic resource]. arXiv. 2021. arXiv:2109.00082 [cs.DC]. DOI: 10.48550/arXiv.2109.00082.
157. Materwala H., Ismail L. Performance and energy-aware bi-objective tasks scheduling for cloud data centers // Procedia Computer Science. 2022. Vol. 197. P. 238–246. DOI: 10.1016/j.procs.2021.12.137.
158. Singh S., Chana I. A Survey on Resource Scheduling in Cloud Computing: Issues and Challenges // Journal of Grid Computing. 2016. Vol. 14, No. 2. P. 217–264. DOI: 10.1007/s10723-015-9359-2.
159. You Q., Tang B. Efficient task offloading using particle swarm optimization algorithm in edge computing for industrial internet of things // Journal of Cloud Computing: Advances, Systems and Applications. 2021. Vol. 10. Art. 41. DOI: 10.1186/s13677-021-00256-4.
160. Mohammadjafari A., Khajouie P. Optimizing Task Scheduling in Heterogeneous Computing Environments: A Comparative Analysis of CPU, GPU, and ASIC Platforms Using E2C Simulator [Electronic resource]. arXiv. 2024. arXiv:2405.08187 [cs.DC]. DOI: 10.48550/arXiv.2405.08187.

ДОДАТОК А

ЗАТВЕРДЖУЮ

Директор КБ ІС

КПІ ім. Ігоря Сікорського

Олесь КОВТУНЕЦЬ

«23» 10 2025 року

АКТ**про впровадження результатів дисертаційної роботи
Лемешка В'ячеслава Анатолійовича**

Комісія в складі: начальника відділу розробки та підтримки інформаційних систем навчального процесу конструкторського бюро інформаційних систем Національного технічного університету України “Київський політехнічний інститут імені Ігоря Сікорського”, Ромашкевич Я.О. - голова і членів: провідного інженера відділу розробки та підтримки інформаційних систем навчального процесу Фіногенова О.Д., провідного інженера відділу адміністрування загальноуніверситетських систем та забезпечення доступу до Європейських мереж Савінова Н.М.. розглянула матеріали звітів та результати роботи під час розробки та впровадження у 2023-2025 роках інформаційних систем автоматизації проектування та виконання бізнес-сервісів встановила наступне.

У процесі розробки й практичного застосування зазначених інформаційних систем впроваджені такі результати дисертаційної роботи дисертаційної роботи Лемешка В.А. “Методи та платформа автоматизації проектування та виконання бізнес-сервісів”, отримані ним особисто:

- досліджено та реалізовано засоби моніторингу виконання кроків бізнес-процесів у платформі автоматизації проектування та виконання бізнес-сервісів;

- розроблено та впроваджено метод планування задач, який передбачає перевірку наявності обчислювальних ресурсів перед запуском кроку бізнес-процесу з метою запобігання перевантаженню системи;

- розроблено та впроваджено метод ресурсно-орієнтованого планування кроків бізнес-процесів, спрямований на мінімізацію часток порушень показника рівня сервісу в умовах динамічного навантаження

шляхом врахування ресурсних обмежень та дедлайн-орієнтовану пріоритезацію за правилом “найраніший термін”.

Тестові випробування та практичне застосування інформаційних систем підтвердили коректність і ефективність запропонованих проєктних рішень та достовірність положень дисертаційної роботи в частині моніторингу виконання кроків бізнес-процесів, контролю ресурсних обмежень і застосування методів планування для підвищення стійкості та ефективності функціонування засобів автоматизації проєктування та виконання бізнес-сервісів.

Даний акт не є підставою для фінансових взаєморозрахунків.

Голова комісії



Яна РОМАШКЕВИЧ

Члени комісії:



Олексій ФІНОГЕНОВ



Наталія САВІНОВА

АКТ

про впровадження результатів дисертаційної роботи
Лемешка В'ячеслава Анатолійовича

Результати, отримані в дисертаційному дослідженні аспіранта Національного технічного університету «Київський політехнічний інститут імені Ігоря Сікорського» Лемешка В'ячеслава Анатолійовича, реалізовані при впровадженні засобів управління та перевірки автоматизації проєктування та виконання бізнес-сервісів. Тема дисертації «Методи та платформа автоматизації проєктування та виконання бізнес-сервісів».

Прийняті до впровадження такі результати дослідження: модель планування кроків бізнес-процесів, яка відрізняється від існуючих тим, що розглядає кроки як задачі із прогнозованими профілями мультиресурсного споживання та одночасно враховує терміни старту, багатовимірні ресурсні обмеження і динамічну зміну навантаження середовища виконання.

Зазначена модель дозволяє, у випадку динамічного навантаження середовища виконання, зменшувати кількість аварійних відмов через перевищення доступних ресурсів і покращувати ефективність виконання кроків бізнес-процесів.

Даний акт не є підставою для фінансових взаєморозрахунків.

Директор ТОВ Інтехфорвард



Є.В. Крилов

ДОДАТОК Б

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА

Наукові праці, в яких опубліковані основні наукові результати дисертації:

1. Корнага Я.І., Лемешко В.А. Порівняння алгоритмів Modified First-Come-First-Served та Reactive Multi-Resource Token–Earliest Deadline First за умов обмеженого доступу до оперативної пам'яті // Телекомунікаційні та інформаційні технології. № 3 (2025). С. 26–33. ISSN 2412-4338.
<https://doi.org/10.31673/2412-4338.2025.038704>

Корнага Я.І. брав участь у постановці наукового завдання порівняльного дослідження методів в умовах обмеженого доступу до оперативної пам'яті, а також в узагальненні одержаних результатів. Лемешко В.А. розробив алгоритмічний опис удосконаленого методу, реалізував експериментальне середовище та програмну реалізацію зазначених методів, виконав аналіз показників продуктивності, порушень цільового показника сервісу та використання ресурсів, що дозволило вперше обґрунтувати удосконалений метод реактивного мульти-токенового планування задач за правилом "перший прийшов - перший оброблений" та обґрунтувати метод реактивного мульти-токенового планування задач за дедлайном старту в ресурсно-обмеженому середовищі.

2. Лемешко В.А. Формалізація процесу планування за методом RMRT-EDF з урахуванням обмежених ресурсів // Адаптивні системи автоматичного управління. Т. 2, № 47 (2025). С. 101–107. ISSN 1560-8956.
<https://doi.org/10.20535/1560-8956.47.2025.340183>

Лемешко В.А. виконав аналіз наукових публікацій за темою статті, запропонував математичну модель процесу планування, що дозволило удосконалити модель моніторингу та виконання кроків бізнес-процесів у ресурсно-обмеженому середовищі.

3. Пархомей І., Бойко Ю., Лемешко В. Математична модель та адаптивне управління алгоритмом обслуговування даних журналу в умовах змінного серверного навантаження // Вісник Хмельницького національного університету. Серія «Технічні науки». Т. 347, № 1 (2025). С.168–174. ISSN 2307-5732. <https://doi.org/10.31891/2307-5732-2025-347-23>

Пархомей І. запропонував критерій ефективності, взяв участь в узагальненні результатів роботи. Бойко Ю. здійснив постановку наукового завдання дослідження, обрахував результати, подані на графіках "Показник sql IOPS", "Показник sql processes blocked", "Показник sql disk queue". Лемешко В. розробив опис параметрів налаштування й алгоритм обслуговування даних журналу бізнес-процесів, що забезпечило розвиток інформаційної технології автоматизації проєктування та виконання бізнес-процесів через реалізацію механізму адаптивного обслуговування й оцінювання ефективності за показниками продуктивності серверного середовища.

4. Пархомей І., Бойко Ю., Лемешко В., Єрмоєнко О. Оптимізація логування в інформаційних системах на основі математичної моделі управління буфером у шаблоні Producer–Consumer // Вимірювальна та обчислювальна техніка в технологічних процесах. № 2(2025). С. 218–225. ISSN 2219-9365. <https://doi.org/10.31891/2219-9365-2025-82-30>

Пархомей І. визначив науково-практичне завдання дослідження. Бойко Ю. формалізував умову ефективного функціонування буфера та виконав розрахунки параметрів моделі. Єрмоєнко О. виконав обрахунок і аналіз експериментальних результатів. Лемешко В. розробив загальні схеми базового і запропонованого підходів для підсистеми логування в системі виконання бізнес-процесів, здійснив параметричне удосконалення підсистеми логування, обґрунтував застосування буфера з агрегацією повідомлень і використання даних Query Store для оцінювання ефективності запропонованого підходу, що забезпечило подальший розвиток інформаційної технології автоматизації проєктування та виконання бізнес-процесів на основі даних моніторингу та оцінювання поточного стану середовища.

5. Пархомей І., Бойко Ю., Лемешко В. Алгоритм налаштування кількості потоків для виконання фонових задач // Вимірювальна та обчислювальна техніка в технологічних процесах. № 4 (2024). С. 162–173. ISSN 2219-9365. <https://doi.org/10.31891/2219-9365-2024-80-20>

Пархомей І. поставив наукову задачу та брав участь в узагальненні отриманих результатів. Бойко Ю. брав участь у визначенні ресурсних параметрів, що враховуються в алгоритмі, та в обґрунтуванні відповідних розрахунків. Лемешко В. розробив загальну схему підсистеми обробки задач, метод визначення кількості фонових обробників та запропонував конфігурацію кількості одночасно виконуваних фонових операцій для всіх підсистем. Загальна схема підсистеми обробки задач відображає виконання кроків бізнес-процесів у спільному ресурсно-обмеженому середовищі та враховує взаємодію підсистем, які використовують спільні обчислювальні ресурси, що дозволило удосконалити модель моніторингу та виконання кроків бізнес-процесів в умовах змінного навантаження. Алгоритм визначення кількості фонових обробників ґрунтується на аналізі стану системи, графіків навантаження, журналів подій і зворотного зв'язку щодо реакції середовища на зміну конфігурації, що забезпечило подальший розвиток інформаційна технологія автоматизації проектування та виконання бізнес-процесів на основі даних моніторингу та оцінювання поточного стану середовища.

Наукові праці, які засвідчують апробацію матеріалів дисертації:

1. Корнага Я. І., Лемешко В.А. RMRT-EDF: реактивний мультиресурсний токен-планувальник для гарантованого запуску задач у реальному часі // XIII Міжнародна науково-практична конференція з інформаційних систем та технологій. 21-22 травня 2025 року. Матеріали конференції, ст. 35-38; Форма участі – дистанційно.

Корнага Я.І. запропонував методологію дослідження реактивного мультиресурсного токенового планування задач за дедлайном старту та брав участь в узагальненні отриманих результатів. Лемешко В.А. розробив алгоритмічне та

архітектурне представлення методу, виконав його імітаційну перевірку та аналіз показників ефективності, що дозволило вперше обґрунтувати метод реактивного мульти-токенового планування задач за дедлайном старту в ресурсно-обмеженому середовищі.

Праці, які додатково відображають результати дисертації:

Відсутні.