

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Кваліфікаційна наукова
праця на правах рукопису

ПОПИК ЮРІЙ ІГОРОВИЧ

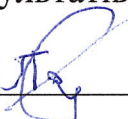
УДК 004.89:004.75

ДИСЕРТАЦІЯ

**ЗНАННЯ-ОРІЄНТОВАНА ПРОГРАМНА СИСТЕМА УПРАВЛІННЯ
КОРПОРАТИВНИМИ КОМП'ЮТЕРНИМИ МЕРЕЖАМИ**

Спеціальність 121 – Інженерія програмного забезпечення
Галузь знань 12 – Інформаційні технології

Подається на здобуття ступеня доктора філософії.
Дисертація містить результати власних досліджень. Використання ідей,
результатів і текстів інших авторів мають посилання на відповідне джерело

/  / Ю.І. Попик

Науковий керівник: Мельник Андрій Миколайович, доктор технічних наук,
професор

Тернопіль – 2026

АНОТАЦІЯ

Попик Ю.І. Знання-орієнтована програмна система управління корпоративними комп'ютерними мережами. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття ступеня доктора філософії за спеціальністю 121 «Інженерія програмного забезпечення» – Західноукраїнський національний університет, Тернопіль, 2026.

Підготовка здійснювалась на кафедрі комп'ютерних наук Західноукраїнського національного університету Міністерства освіти і науки України.

Дисертаційна робота присвячена розв'язуванню актуального науково-технічного завдання підвищення ефективності управління корпоративними комп'ютерними мережами шляхом розроблення моделей і методів на основі знання-орієнтованого підходу, що враховують структуру мережі, її ресурси та множину задач, які ними розв'язуються.

Сучасні корпоративні комп'ютерні мережі є складними розподіленими інформаційними середовищами, які об'єднують значну кількість користувачів, серверів, мережевих сервісів, інформаційних ресурсів та прикладних задач. Постійне зростання обсягів інформації, підвищення вимог до доступності сервісів, необхідність забезпечення кібербезпеки та ефективного використання ресурсів призводять до ускладнення процесів адміністрування мережевої інфраструктури. Традиційні системи управління корпоративними мережами переважно орієнтовані на моніторинг технічних параметрів обладнання та автоматизацію окремих адміністративних операцій, проте недостатньо враховують знання про взаємозв'язки між ресурсами, сервісами, користувачами та множиною задач, що розв'язуються в мережевому середовищі. Це обмежує можливості автоматизованого прийняття обґрунтованих конфігураційних рішень та знижує ефективність використання мережевих ресурсів.

Проведений аналіз сучасних методів управління корпоративними

комп'ютерними мережами показав, що існуючі програмні платформи моніторингу та адміністрування, зокрема Cisco DNA Center, SolarWinds Network Performance Monitor, ManageEngine OpManager, PRTG Network Monitor та Microsoft System Center, забезпечують високий рівень збору та візуалізації інформації про стан мережі, однак не підтримують повноцінного онтологічного представлення знань про предметну область і механізмів логічного виведення, необхідних для реалізації інтелектуального управління мережевою інфраструктурою.

У роботі обґрунтовано доцільність використання знання-орієнтованого підходу для управління корпоративними комп'ютерними мережами. На відміну від традиційних підходів, запропонована концепція базується на формалізації знань про структуру мережі, ресурси, сервіси, користувачів, політики безпеки та множину задач у межах єдиного семантичного простору, що створює основу для автоматизованого логічного виведення та підтримки прийняття рішень.

У першому розділі проведено аналіз особливостей функціонування корпоративних комп'ютерних мереж, сучасних методів їх управління, моніторингу, конфігурування та балансування навантаження. Досліджено знання-орієнтовані підходи до побудови інтелектуальних інформаційних систем, проаналізовано методи формалізації знань, включаючи логічні моделі, продукційні системи, семантичні мережі, фреймові структури та онтології. Також розглянуто можливості використання нечіткої логіки, машинного навчання та штучних нейронних мереж для підтримки процесів управління мережевою інфраструктурою. На основі проведеного аналізу сформульовано науково-технічну проблему та обґрунтовано необхідність застосування онтологічного підходу.

У другому розділі розроблено математичне забезпечення знання-орієнтованої системи управління корпоративними комп'ютерними мережами. Запропоновано онтологічну модель корпоративної мережі, яка формалізує знання про мережеві ресурси, сервіси, користувачів та множину задач у вигляді концептів, відношень, аксіом та правил логічного виведення. Розроблено метод онтологічно-керованого конфігурування корпоративної мережі, який забезпечує автоматизоване формування

конфігураційних рішень на основі семантичного аналізу знань про предметну область та механізмів логічного виведення. Удосконалено математичну модель розподілу навантаження між центрами термінальних кластерів на основі багатокритеріальної оптимізації та онтологічного подання знань про ресурси корпоративної комп'ютерної мережі й множину задач. Запропонована модель дозволяє враховувати не лише поточний стан ресурсів, але й семантичні взаємозв'язки між задачами, сервісами та мережевою інфраструктурою під час формування оптимальних конфігурацій розподілу навантаження.

У третьому розділі розроблено архітектуру знання-орієнтованої системи управління корпоративними комп'ютерними мережами. Запропоновано багаторівневу програмну архітектуру, яка інтегрує підсистеми моніторингу, управління онтологічною базою знань, логічного виведення, конфігурування та оптимізації. Реалізовано онтологічну базу знань на основі технології OWL та механізм логічного виведення, який забезпечує автоматичне формування нових знань на основі фактів, отриманих від підсистеми моніторингу. Розроблено програмну реалізацію математичної моделі розподілу навантаження між центрами термінальних кластерів та забезпечено її інтеграцію з онтологічною базою знань і механізмом логічного виведення. Обґрунтовано вибір технологій програмної реалізації та описано особливості інтеграції програмних компонентів у межах єдиної системи.

У четвертому розділі наведено результати програмної реалізації та експериментальних досліджень розробленої системи. Реалізовано веборієнтоване програмне забезпечення з використанням сучасних технологій розроблення програмних систем. Проведено оцінювання ефективності запропонованого методу онтологічно-керованого конфігурування та математичної моделі розподілу навантаження між центрами термінальних кластерів. Результати експериментів підтвердили можливість автоматизованого формування конфігураційних рішень, зменшення рівня нерівномірності використання ресурсів мережі та підвищення ефективності функціонування мережевої інфраструктури. Виконано апробацію

запропонованих моделей і методів на прикладі корпоративної мережі Західноукраїнського національного університету та мережі компанії «Колумбус». Отримані результати показали зменшення середньої затримки доступу до сервісів на 29–36 %, зниження завантаження серверних ресурсів на 20–24 %, скорочення кількості порушень політик доступу та покращення інтегральних показників якості функціонування мережі майже на 50 %.

Практичне значення одержаних результатів полягає у створенні знання-орієнтованої програмної системи управління корпоративними комп'ютерними мережами, яка забезпечує автоматизацію процесів моніторингу, конфігурування ресурсів, розподілу навантаження та підтримки прийняття рішень на основі онтологічного представлення знань. Результати дослідження можуть бути використані під час проектування, модернізації та експлуатації корпоративних комп'ютерних мереж підприємств, установ та організацій різного профілю діяльності.

У висновках сформульовано основні результати дисертаційної роботи.

Ключові слова: програмно-апаратні засоби (апаратно-програмні засоби); інтелектуалізована система; комп'ютерна мережа; архітектура програмного забезпечення; онтологія (онтологія предметної області); база знань; інформаційні технології; нечітка система (нечітка логіка); штучний інтелект; прийняття рішень; нейронні мережі; математична модель (математичне моделювання); методи оптимізації (оптимізація); комп'ютерна система; машинне навчання.

ПЕРЕЛІК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Наукові праці, в яких опубліковано основні наукові результати дисертації

1. Dyvak, M.; Tsapiv, Y.; Pukas, A.; Petrovskiy, Y.; Melnyk, A.; Dyvak, A.; Banasik, A.; Czupryna-Nowak, A.; Pikiwicz, P.; Popyk, Y.; Dzyha, Y. AR Technology for Restoring Upper-Limb Joint Mobility in Patients. Appl. Sci. 2026, 16, 5878. (3,6 д.а. / 0,5 д.а.; особистий внесок: програмний модуль налаштування параметрів системи та оцінка його ефективності).

DOI: <https://doi.org/10.3390/app16125878> (Scopus, Q2).

2. Melnyk A., Popyk Y. An Ontology-Driven Knowledge-Based Approach to Complex Systems Management. *Computer Systems and Information Technologies*. 2026. № 2. P. 86-99. (1,4 д.а. / 0,8 д.а.; особистий внесок: онтологічна модель корпоративної комп'ютерної мережі, метод онтологічно-керованого конфігурування корпоративної комп'ютерної мережі).

DOI: <https://doi.org/10.31891/csit-2026-2-9>

3. Мельник А. М., Попик Ю. І. Програмна архітектура онтологічно-керованої системи для управління складними об'єктами. *Вісник Національного університету водного господарства та природокористування. Серія «Технічні науки»*. 2026. Вип. 1(125). С. 276-302. (2,7 д.а. / 2,5 д.а.; особистий внесок: модель динамічного розподілу навантаження центрів термінальних кластерів, архітектура програмної онтологічно-керованої системи управління корпоративними комп'ютерними мережами).

DOI: <https://doi.org/10.31713/vt1202620>

4. Popyk Y., Ples M. Optimization of Energy Consumption in Data Centers Based on an Ontological Approach. *Opto-Electronic Devices and Components in Laser and Energy Technologies*. 2025, № 50 (2), 282-292. (1,2 д.а. / 0,8 д.а.; особистий внесок: апробація та підхід до налаштування параметрів онтологічно-керованого конфігурування корпоративної комп'ютерної мережі).

DOI: <https://doi.org/10.31649/1681-7893-2025-50-2-282-292>

5. M. Pylypchuk, N. Porplytsya, I. Stasiv, R. Monko, Y. Popyk, T. Yurchyshyn, Parallelization and Optimization of a Method for Web Content Analysis and Accessibility Evaluation for Visually Impaired Users. 2025 15th International Conference on Advanced Computer Information Technologies (ACIT), Sibenik, Croatia, 2025, pp. 879-883. (0,5 д.а. / 0,1 д.а.; особистий внесок: метод оцінки доступності інформаційних ресурсів та його вплив на продуктивність роботи системи).

DOI: <https://doi.org/10.1109/ACIT65614.2025.11185892> (Scopus)

6. A. Melnyk, I. Romanets, R. Shevchuk, O. Kvasnytsia, Y. Popyk, V. Ivanyna.

An Intelligent System for Detecting Malicious Messages in IP Telephony within a Corporate Network. 2025 IEEE 13th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Gliwice, Poland, 2025, pp. 1-5. (0,5 д.а. / 0,1 д.а.; особистий внесок: алгоритм налаштування корпоративної мережі при аналізі достовірності контенту).

DOI: <https://doi.org/10.1109/IDAACS68557.2025.11322321> (Scopus).

Наукові праці, які засвідчують апробацію матеріалів дисертації

7. Попик Ю.І. Знання-орієнтована програмна система управління корпоративними комп'ютерними мережами. Матеріали школи-семінару молодих вчених і студентів «Комп'ютерні інформаційні технології» (СІТ'2026). Тернопіль. 2026, ст. 190-191 (0,1 д.а.).

8. Segin A. I., Popyk Y. I. System of Automatic Translation of Verbal Audio Streams in Real Time. Materials of the 17th International Symposium of Croatian Metallurgical Society SHMD'2024 (published in: Metalurgija 63 (2024) 2). P. 317-320. (0,4 д.а. / 0,1 д.а.; особистий внесок: програмне забезпечення для побудови кореляційних моделей у полярній системі координат, реалізації алгоритмів обробки даних та аналізу точності отриманих моделей на тестових наборах даних).

URL: <https://hrcak.srce.hr/en/file/451123>

9. Сегін А. І., Заставний О. М., Попик Ю. І. Модель передбачення географічного місцеположення і часу виникнення землетрусів. Комп'ютерне моделювання та програмне забезпечення інформаційних систем і технологій (КМПЗ-2024): збірник наукових праць (тези доповідей та вибрані статті) IV Міжнародної науково-практичної конференції, Чернівці, 30 травня – 01 червня 2024 р. Львів: ЛНУ імені Івана Франка, 2024. С. 137-141. (0,5 д.а. / 0,1 д.а.; особистий внесок: аналіз вимог до програмних систем прогнозування). URL: https://bukuniver.edu.ua/wp-content/themes/bukuniver/docs/tezy_kmpz_2024.pdf

10. Сегін А. І., Попик Ю.І., Когут І. Р., Когут Ю., Павловський А. О., Возняк В. С. Кореляційні моделі в полярній системі координат // Збірник матеріалів

проблемно-наукової міжгалузевої конференції «Інформаційні проблеми комп'ютерних систем, юриспруденції, енергетики, моделювання та управління» (ICSM-2023). Надвірна, 2023. С. 188-191. (0,4 д.а. / 0,1 д.а.; особистий внесок: результати формалізації методу кореляційного аналізу знакових функцій).

URL:

https://drive.google.com/file/d/1eLKEeeVfgQ1lxU8AfwdO_FF5ywHBgdch/view

ANNOTATION

Popyk Yu.I. Knowledge-Oriented Software System for Corporate Computer Network Management. – Scientific work on the rights of the manuscript.

Thesis for the degree of Doctor of Philosophy in the specialty 121 "Software Engineering" - West Ukrainian National University, Ternopil, 2026.

The research was carried out at the Department of Computer Science of the West Ukrainian National University of the Ministry of Education and Science of Ukraine.

The dissertation is devoted to solving the relevant scientific and technical problem of improving the efficiency of corporate computer network management through the development of models and methods based on a knowledge-oriented approach that takes into account the network structure, its resources, and the set of tasks performed within it.

Modern corporate computer networks are complex distributed information environments that integrate a large number of users, servers, network services, information resources, and application tasks. The continuous growth of information volumes, increasing requirements for service availability, the need to ensure cybersecurity, and the efficient utilization of resources significantly complicate network infrastructure administration. Traditional corporate network management systems are primarily focused on monitoring technical equipment parameters and automating individual administrative operations; however, they insufficiently consider knowledge about the relationships among resources, services, users, and the set of tasks performed within the network environment. This limitation reduces the ability to make automated and well-grounded configuration decisions and decreases the efficiency of network resource utilization.

The analysis of modern corporate computer network management methods demonstrated that existing monitoring and administration platforms, including Cisco DNA Center, SolarWinds Network Performance Monitor, ManageEngine OpManager, PRTG Network Monitor, and Microsoft System Center, provide a high level of network

status monitoring and visualization. However, they do not support comprehensive ontological knowledge representation of the domain and logical reasoning mechanisms required for intelligent network infrastructure management.

The dissertation substantiates the feasibility of applying a knowledge-oriented approach to corporate computer network management. Unlike traditional approaches, the proposed concept is based on the formalization of knowledge about network structure, resources, services, users, security policies, and task sets within a unified semantic space. This creates the foundation for automated logical reasoning and decision support.

The first chapter analyzes the operational features of corporate computer networks, modern methods of their management, monitoring, configuration, and load balancing. Knowledge-oriented approaches to intelligent information system development are investigated, and methods of knowledge formalization, including logical models, production systems, semantic networks, frame structures, and ontologies, are analyzed. The possibilities of applying fuzzy logic, machine learning, and artificial neural networks to support network infrastructure management processes are also examined. Based on the conducted analysis, the scientific and technical problem is formulated, and the necessity of applying an ontological approach is substantiated.

The second chapter develops the mathematical foundations of a knowledge-oriented corporate computer network management system. An ontological model of a corporate network is proposed, formalizing knowledge about network resources, services, users, and task sets in the form of concepts, relationships, axioms, and logical inference rules. An ontology-driven network configuration method is developed that enables the automated generation of configuration decisions based on semantic analysis of domain knowledge and logical reasoning mechanisms. A mathematical model for workload distribution among terminal cluster centers is improved using multicriteria optimization and ontological representation of knowledge about corporate network resources and task sets. The proposed model makes it possible to consider not only the current state of resources but also semantic relationships among tasks, services, and

network infrastructure when generating optimal workload distribution configurations.

The third chapter presents the architecture of a knowledge-oriented corporate computer network management system. A multilayer software architecture integrating monitoring, ontology-based knowledge management, logical reasoning, configuration, and optimization subsystems is proposed. An ontology-based knowledge base was implemented using OWL technology, together with a logical inference mechanism that automatically generates new knowledge from facts obtained by the monitoring subsystem. A software implementation of the workload distribution model for terminal cluster centers was developed and integrated with the ontology knowledge base and reasoning engine. The choice of software implementation technologies is justified, and the integration features of software components within a unified system are described.

The fourth chapter presents the results of software implementation and experimental studies of the developed system. A web-oriented software solution was implemented using modern software engineering technologies. The effectiveness of the proposed ontology-driven configuration method and the mathematical model of workload distribution among terminal cluster centers was evaluated. Experimental results confirmed the possibility of automated generation of configuration decisions, reduction of resource utilization imbalance, and improvement of network infrastructure performance. The proposed models and methods were validated using the corporate network of West Ukrainian National University and the network infrastructure of Columbus Company. The obtained results demonstrated a reduction in average service access latency by 29–36%, a decrease in server resource utilization by 20–24%, a reduction in access policy violations, and an improvement in integrated network performance indicators of nearly 50%.

The practical significance of the obtained results lies in the development of a knowledge-oriented software system for corporate computer network management that automates monitoring, resource configuration, workload distribution, and decision-support processes based on ontological knowledge representation. The research results can be applied in the design, modernization, and operation of corporate computer

networks in enterprises, institutions, and organizations of various domains.

The conclusions summarize the principal scientific and practical results of the dissertation.

Keywords: hardware-software systems; intelligent system; computer network; software architecture; ontology (domain ontology); knowledge base; information technologies; fuzzy system(fuzzy logic); artificial intelligence; decision making; neural networks; mathematical model (mathematical modeling); optimization methods (optimization); computer system; machine learning.

LIST OF PUBLISHED PAPERS BY THE TOPIC OF THESIS

SCIENTIFIC PAPERS IN WHICH THE MAIN SCIENTIFIC RESULTS OF THE DISSERTATION WERE PUBLISHED:

1. Dyvak, M.; Tsapiv, Y.; Pukas, A.; Petrovskiy, Y.; Melnyk, A.; Dyvak, A.; Banasik, A.; Czupryna-Nowak, A.; Pikiwicz, P.; Popyk, Y.; Dzyha, Y. AR Technology for Restoring Upper-Limb Joint Mobility in Patients. *Appl. Sci.* 2026, 16, 5878. (3.6 author's sheets / 0.5 author's sheets; personal contribution: development of a software module for configuring system parameters and evaluation of its effectiveness).

DOI: <https://doi.org/10.3390/app16125878> (Scopus, Q2).

2. Melnyk A., Popyk Y. An Ontology-Driven Knowledge-Based Approach to Complex Systems Management. *Computer Systems and Information Technologies.* 2026. № 2. P. 86-99. (1.4 author's sheets / 0.8 author's sheets; personal contribution: ontological model of a corporate computer network and a method for ontology-driven configuration of a corporate computer network).

DOI: <https://doi.org/10.31891/csit-2026-2-9>

3. Melnyk, A.M.; Popyk, Yu.I. Software Architecture of an Ontology-Driven System for Complex Object Management. *Bulletin of the National University of Water and Environmental Engineering. Technical Sciences Series.* 2026, Issue 1(125), pp. 276–302. (2.7 author's sheets / 2.5 author's sheets; personal contribution: development of a

dynamic workload distribution model for terminal cluster centers and the architecture of an ontology-driven software system for corporate computer network management).

DOI: <https://doi.org/10.31713/vt1202620>

4. Popyk Y., Ples M. Optimization of Energy Consumption in Data Centers Based on an Ontological Approach. *Opto-Electronic Devices and Components in Laser and Energy Technologies*. 2025, № 50 (2), 282-292. (1.2 author's sheets / 0.8 author's sheets; personal contribution: validation and development of an approach for tuning the parameters of ontology-driven configuration of a corporate computer network)..

DOI: <https://doi.org/10.31649/1681-7893-2025-50-2-282-292>

5. M. Pylypchuk, N. Porplytsya, I. Stasiv, R. Monko, Y. Popyk, T. Yurchyshyn, Parallelization and Optimization of a Method for Web Content Analysis and Accessibility Evaluation for Visually Impaired Users. 2025 15th International Conference on Advanced Computer Information Technologies (ACIT), Sibenik, Croatia, 2025, pp. 879-883. (0.5 author's sheets / 0.1 author's sheets; personal contribution: method for evaluating the accessibility of information resources and analysis of its impact on system performance).

DOI: <https://doi.org/10.1109/ACIT65614.2025.11185892> (Scopus)

6. A. Melnyk, I. Romanets, R. Shevchuk, O. Kvasnytsia, Y. Popyk, V. Ivanyna. An Intelligent System for Detecting Malicious Messages in IP Telephony within a Corporate Network. 2025 IEEE 13th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Gliwice, Poland, 2025, pp. 1-5. (0.5 author's sheets / 0.1 author's sheets; personal contribution: development of a corporate network configuration algorithm for content credibility analysis).

DOI: <https://doi.org/10.1109/IDAACS68557.2025.11322321> (Scopus).

SCIENTIFIC WORKS CERTIFYING THE APPROVAL OF THE DISSERTATION MATERIALS:

7. Popyk, Yu.I. Knowledge-Oriented Software System for Corporate Computer

Network Management. In: Proceedings of the School-Seminar of Young Scientists and Students “Computer Information Technologies” (CIT’2026), Ternopil, Ukraine, 2026, pp. 190–191. (0.1 author's sheets).

8. Segin, A.I.; Popyk, Y.I. System of Automatic Translation of Verbal Audio Streams in Real Time. Materials of the 17th International Symposium of the Croatian Metallurgical Society SHMD'2024 (published in: Metalurgija 63 (2024), No. 2), pp. 317–320. (0.4 author's sheets / 0.1 author's sheets; personal contribution: software development for constructing correlation models in a polar coordinate system, implementation of data processing algorithms, and accuracy analysis of the obtained models using test datasets).

URL: <https://hrcak.srce.hr/en/file/451123>

9. Segin, A.I.; Zastavnyi, O.M.; Popyk, Y.I. A Model for Predicting the Geographic Location and Time of Earthquake Occurrence. In: Computer Modeling and Software of Information Systems and Technologies (CMPSIST-2024): Proceedings (Abstracts and Selected Papers) of the IV International Scientific and Practical Conference, Chernivtsi, Ukraine, May 30 – June 1, 2024. Lviv: Ivan Franko National University of Lviv, 2024, pp. 137–141. (0.5 author's sheets / 0.1 author's sheets; personal contribution: analysis of requirements for forecasting software systems).

URL: https://bukuniver.edu.ua/wp-content/themes/bukuniver/docs/tezy_kmpz_2024.pdf

10. Segin, A.I.; Popyk, Y.I. ; Kohut, I.R.; Kohut, Yu.; Pavlovskyi, A.O.; Vozniak, V.S. Correlation Models in a Polar Coordinate System. In: Proceedings of the Interdisciplinary Scientific Conference “Information Problems of Computer Systems, Law, Energy, Modeling and Management” (ICSM-2023). Nadvirna, Ukraine, 2023, pp. 188–191. (0.4 author's sheets / 0.1 author's sheets; personal contribution: results of formalizing the method of correlation analysis of sign functions).

URL: https://drive.google.com/file/d/1eLKEeeVfgQ1lxU8AfwdO_FF5ywhBgdch/view

ЗМІСТ

АНОТАЦІЯ.....	2
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	17
ВСТУП.....	18
РОЗДІЛ 1 АНАЛІЗ МЕТОДІВ ТА ПРОГРАМНИХ ЗАСОБІВ ДЛЯ ПОБУДОВИ ЗНАННЯ-ОРІЄНТОВАНИХ ПРОГРАМНИХ СИСТЕМ	27
1.1 Особливості функціонування та управління корпоративними комп'ютерними мережами	28
1.2 Аналіз сучасних методів управління корпоративними комп'ютерними мережами	32
1.3 Аналіз сучасних програмних засобів управління корпоративними мережами	38
1.4 Аналіз знання-орієнтованих підходів та онтологічних моделей.....	41
1.5 Постановка задачі дослідження	47
Висновки до розділу 1	50
РОЗДІЛ 2 ОНТОЛОГІЧНЕ МОДЕЛЮВАННЯ ПРОЦЕСІВ У КОРПОРАТИВНИХ КОМП'ЮТЕРНИХ МЕРЕЖАХ	53
2.1 Теоретичне підґрунтя побудови онтології корпоративної мережі	54
2.2 Формалізація онтології корпоративної мережі.....	64
2.3 Формалізація зв'язку між онтологічними аксіомами та основними операціями алгебри кортежів	73
2.4 Метод онтологічно-керованої конфігурації корпоративної мережі на основі множини задач.....	77
Висновки до розділу 2	86
РОЗДІЛ 3 АРХІТЕКТУРА ЗНАННЯ-ОРІЄНТОВАНОЇ СИСТЕМИ УПРАВЛІННЯ КОРПОРАТИВНИМИ КОМП'ЮТЕРНИМИ МЕРЕЖАМИ	88
3.1 Проектування архітектури знання-орієнтованої системи управління корпоративними комп'ютерними мережами.....	89
3.2 Реалізація онтологічної бази знань та механізму логічного виведення	97

3.3 Математичну модель розподілу навантаження між центрами термінальних кластерів	101
Висновки до розділу 3	116
РОЗДІЛ 4 ЗНАННЯ-ОРІЄНТОВАНА ПРОГРАМНА СИСТЕМА ДЛЯ УПРАВЛІННЯ КОРПОРАТИВНИМИ КОМП'ЮТЕРНИМИ МЕРЕЖАМИ	118
4.1 Організація користувацького інтерфейсу програмної системи управління корпоративними комп'ютерними мережами.....	119
4.2 Експериментальні дослідження методу онтологічно-керованої конфігурації корпоративної мережі	134
Висновки до розділу 4	150
ВИСНОВКИ	153
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	156
ДОДАТОК А ФРАГМЕНТ OWL-ОНТОЛОГІЇ КОРПОРАТИВНОЇ КОМП'ЮТЕРНОЇ МЕРЕЖІ.....	169
ДОДАТОК Б ЛІСТИНГ КОДУ РЕАЛІЗАЦІЇ МЕТОДУ ОНТОЛОГІЧНО-КЕРОВАНОГО КОНФІГУРУВАННЯ КОРПОРАТИВНОЇ МЕРЕЖІ.....	175
ДОДАТОК В ЛІСТИНГ КОДУ ОСНОВНИХ МОДУЛІВ СИСТЕМИ.....	183
ДОДАТОК Г СПИСОК ПУБЛІКАЦІЙ ЗА ТЕМОЮ ДИСЕРТАЦІЇ.....	188
ДОДАТОК Д АКТИ ВПРОВАДЖЕНЬ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОГО ДОСЛІДЖЕННЯ	190

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

DDL	Data Definition Language
DML	Data Manipulation Language
DSS	Decision Support System
MVC	Model View Controller
АК	алгебра кортежів
БД	база даних
КМ	комп'ютерна мережа
ММ	математична модель
ПЗ	програмне забезпечення
СМ	система моніторингу
СУБД	система управління базою даних

ВСТУП

Актуальність теми. Сучасні корпоративні комп'ютерні мережі є складними розподіленими інформаційно-комунікаційними системами, які забезпечують функціонування інформаційних ресурсів, корпоративних сервісів, систем зберігання даних, засобів електронної комунікації та прикладних інформаційних систем. Постійне зростання кількості користувачів, мережевих сервісів, обсягів переданих даних та складності інформаційних процесів призводить до суттєвого ускладнення процесів управління мережевою інфраструктурою. У таких умовах особливого значення набувають питання ефективного конфігурування мережевих ресурсів, забезпечення необхідного рівня продуктивності, надійності та якості обслуговування користувачів [8, 9, 11, 54, 60, 61].

Незважаючи на значну кількість сучасних програмних засобів управління корпоративними мережами, таких як Cisco Prime Infrastructure, SolarWinds Network Performance Monitor, Zabbix, Nagios, OpenDaylight, IBM Netcool та інших, більшість із них орієнтовані переважно на моніторинг технічних параметрів мережевої інфраструктури та використання набору заздалегідь визначених правил адміністрування [96, 97, 111, 112, 114]. При цьому процеси прийняття рішень щодо конфігурування мережевих ресурсів, перерозподілу навантаження та адаптації мережі до зміни умов функціонування значною мірою залишаються залежними від експертних знань адміністратора [20, 30].

Однією з основних проблем існуючих систем управління корпоративними мережами є відсутність формалізованого подання знань про структуру мережі, її ресурси, сервіси, користувачів та множину прикладних задач, що розв'язуються в мережевому середовищі. Як наслідок, більшість систем не забезпечують можливості автоматизованого аналізу семантичних зв'язків між компонентами мережі та не підтримують механізми логічного виведення, необхідні для інтелектуалізації процесів управління [74, 82, 92].

Додатковою проблемою є те, що існуючі методи балансування навантаження

здебільшого базуються на локальних показниках використання ресурсів, таких як завантаження процесора, оперативної пам'яті або мережевих каналів. Водночас вони не враховують змістовні взаємозв'язки між мережевими ресурсами, сервісами та задачами, що розв'язуються в корпоративному середовищі. Це призводить до зниження ефективності прийнятих рішень, особливо в умовах динамічної зміни структури навантаження, появи нових сервісів або зміни пріоритетів користувачів. Недоліком існуючих підходів є також відсутність єдиного механізму інтеграції процесів моніторингу, конфігурування, розподілу навантаження та підтримки прийняття рішень. У більшості випадків зазначені функції реалізуються окремими програмними компонентами, що ускладнює автоматизацію процесів управління та знижує адаптивність корпоративної мережі до змін умов її функціонування.

Перспективним напрямом розв'язання зазначених проблем є застосування знання-орієнтованих технологій, зокрема онтологічного представлення знань [35, 37, 57, 58, 63]. Використання онтологій дозволяє формалізувати знання про структуру корпоративної мережі, її ресурси, сервіси та множину задач, що розв'язуються в мережевому середовищі, забезпечити підтримку механізмів логічного виведення та автоматизувати процеси прийняття рішень. Поєднання онтологічного моделювання, алгебри кортежів, методів логічного виведення та багатокритеріальної оптимізації створює передумови для побудови інтелектуалізованих систем управління корпоративними комп'ютерними мережами нового покоління.

Отже, актуальним є науково-технічне завдання розроблення методів, моделей та програмних засобів знання-орієнтованого управління корпоративними комп'ютерними мережами на основі онтологічного представлення знань про ресурси мережі, сервіси та множину задач, що розв'язуються в мережевому середовищі. Розв'язання цього завдання дозволить автоматизувати процеси конфігурування мережевих ресурсів, підтримки прийняття рішень та розподілу навантаження між центрами термінальних кластерів, підвищити ефективність використання ресурсів корпоративної мережі, покращити якість обслуговування

користувачів та забезпечити адаптивність функціонування мережевої інфраструктури в умовах динамічної зміни навантажень і структури прикладних задач.

Мета і завдання дослідження. Метою дисертаційного дослідження є підвищення ефективності управління корпоративними комп'ютерними мережами шляхом розроблення моделей і методів на основі знання-орієнтованого підходу, що враховують структуру мережі, її ресурси та множину задач, які ними розв'язуються.

Для досягнення цієї мети необхідно виконати наступні завдання:

- провести аналіз сучасних технологій, методів і програмних засобів управління корпоративними комп'ютерними мережами, включаючи системи моніторингу, конфігурації та балансування навантаження;
- дослідити можливості застосування знання-орієнтованих, зокрема, онтологічних підходів для підвищення ефективності управління мережевою інфраструктурою;
- розробити онтологічну модель корпоративної комп'ютерної мережі, що забезпечує формалізацію знань про структуру мережі, її ресурси, сервіси та множину задач, які розв'язуються в мережевому середовищі;
- розробити метод онтологічно-керованого конфігурування корпоративної комп'ютерної мережі, який базується на семантичному поданні знань про мережеві ресурси, сервіси та множину задач і забезпечує автоматизоване прийняття конфігураційних рішень на основі механізмів логічного виведення;
- розробити архітектуру програмної онтологічно-керованої системи управління корпоративними комп'ютерними мережами, яка забезпечує інтеграцію онтологічного представлення знань про структуру мережі, її ресурси та множину задач із механізмами логічного виведення, адаптивного прийняття рішень і динамічного розподілу навантаження;
- удосконалити математичну модель розподілу навантаження між центрами термінальних кластерів на основі багатокритеріальної оптимізації та

онтологічного подання знань про ресурси корпоративної комп'ютерної мережі й множину задач;

- провести апробацію розробленої, знання-орієнтованої програмної системи на прикладі діючих корпоративних компютерних мереж, які орієнтовані на розв'язання прикладних задач.

Об'єкт дослідження – процеси знання-орієнтованого управління корпоративними комп'ютерними мережами.

Предмет дослідження – моделі, методи та програмні засоби знання-орієнтованого управління корпоративними комп'ютерними мережами.

Методи дослідження. Для формалізації знань про структуру корпоративної мережі, її ресурси, сервіси та множину задач застосовано методи онтологічного моделювання, теорії множин, алгебри кортежів, математичної логіки та знання-орієнтованого представлення знань. Алгебру кортежів використано для формального опису взаємозв'язків між ресурсами, сервісами, користувачами та задачами мережі. Під час розроблення методу онтологічно-керованої конфігурації мережі використано методи логічного виведення, продукційних правил, семантичного аналізу та алгебри кортежів. Для побудови математичної моделі розподілу навантаження між центрами термінальних кластерів застосовано методи математичного моделювання, багатокритеріальної оптимізації, теорії прийняття рішень та дослідження операцій. Під час проектування архітектури системи використано методи системного проектування, об'єктно-орієнтованого аналізу та моделювання інформаційних систем. Для реалізації та перевірки запропонованих моделей і методів застосовано комп'ютерне моделювання, тестування програмного забезпечення та статистичний аналіз результатів експериментів.

Наукова новизна отриманих результатів. У межах дисертаційної роботи *вперше*:

- розроблено онтологічну модель корпоративної комп'ютерної мережі, яка, на відміну від існуючих підходів до опису мережевої інфраструктури, враховує семантичні зв'язки між ресурсами мережі, сервісами та множиною задач, що ними

розв'язуються, забезпечуючи формалізоване подання знань для підтримки процесів конфігурації та управління корпоративною мережею, що дозволило підвищити обґрунтованість прийняття управлінських рішень і скоротити час налаштування мережевих ресурсів;

- запропоновано метод онтологічно-керованого конфігурування корпоративної комп'ютерної мережі, який ґрунтується на семантичному поданні знань про мережеві ресурси, сервіси та множину задач, що розв'язуються в мережі, та, на відміну від традиційних підходів до конфігурування, забезпечує автоматизоване формування конфігурацій на основі механізмів логічного виведення;

- запропоновано та обґрунтовано архітектуру програмної онтологічно-керованої системи управління корпоративними комп'ютерними мережами, яка забезпечує інтеграцію формалізованих знань про структуру мережі, її ресурси та множину задач із моделями динамічного розподілу навантаження, механізмами логічного виведення та адаптивного прийняття рішень, що створює теоретичну та програмну основу для побудови самоналаштовуваних систем управління корпоративними мережами;

набула подальшого розвитку:

- математична модель розподілу навантаження між центрами термінальних кластерів, яка, на відміну від існуючих моделей на основі локальних показників завантаженості ресурсів, базується на розв'язанні багатокритеріальної задачі оптимізації з використанням онтологічного подання знань про ресурси мережі та множину задач, що дозволяє здійснювати пошук субоптимальних рішень щодо розподілу навантаження за критеріями продуктивності, збалансованості використання ресурсів та якості обслуговування користувачів.

Особистий внесок здобувача. Основні наукові результати, наведені в дисертаційній роботі, отримані автором особисто. Усі теоретичні положення, математичні моделі, методи, алгоритми та програмні засоби розроблені автором самостійно. У наукових працях, опублікованих у співавторстві, автору належать:

[1] програмний модуль налаштування параметрів системи та оцінка його ефективності;

[2] онтологічна модель корпоративної комп'ютерної мережі, метод онтологічно-керованого конфігурування корпоративної комп'ютерної мережі;

[3] модель динамічного розподілу навантаження центрів термінальних кластерів, архітектура програмної онтологічно-керованої системи управління корпоративними комп'ютерними мережами;

[4] апробація та підхід до налаштування параметрів онтологічно-керованого конфігурування корпоративної комп'ютерної мережі;

[5] метод оцінки доступності інформаційних ресурсів та його вплив на продуктивність роботи системи;

[6] алгоритм налаштування корпоративної мережі при аналізі достовірності контенту;

[8] програмне забезпечення для побудови кореляційних моделей у полярній системі координат, реалізації алгоритмів обробки даних та аналізу точності отриманих моделей на тестових наборах даних;

[9] аналізу вимог до програмних систем прогнозування;

[10] результати формалізації методу кореляційного аналізу знакових функцій;

Основні результати дисертаційного дослідження, його наукові положення, методи, моделі та практичні розробки опубліковані у наукових працях автора та викладені в них у повному обсязі. Опубліковані праці відображають зміст дисертаційної роботи, її наукову новизну, теоретичне та практичне значення.

Апробація результатів дисертації. Основні положення і результати дисертаційної роботи презентовано на 5 конференціях, зокрема:

- 13th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications Gliwice, Poland 2025;
- 2025 15th International Conference on Advanced Computer Information Technologies (ACIT'2025). Šibenik, Croatia, 2025;

- 7th International Conference on Computer Science, Engineering and Education Applications (ICCSEEA2024);
- 17th International Symposium of Croatian Metallurgical Society - SHMD'2024;
- Проблемно-наукова міжгалузева конференція “Інформаційні проблеми комп’ютерних систем, юриспруденції, енергетики, моделювання та управління” (ICSM-2023);
- Всеукраїнська школа-семінар молодих вчених і студентів «Комп’ютерні інформаційні технології» (СІТ-2026).

Публікації. За результатами дисертаційних досліджень опубліковано 10 наукових праць (Додаток Г) загальним обсягом 134 сторінки, зокрема 6 статей у фахових наукових виданнях [1, 2 3, 4, 5, 6], 3 з яких входить до міжнародної наукометричної бази Scopus та Web of Science [5, 6] та 1 відповідно до класифікації SCImago Journal and Country Rank віднесено до квартилю Q2 [1], 4 публікацій у матеріалах конференцій [7, 8, 9, 10].

Структура та обсяг роботи. Дисертаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел із 119 найменувань та 5 додатків. Загальний обсяг роботи складає 194 сторінки друкарського тексту, з них 155 сторінок основного тексту. Робота містить 41 рисунок і 18 таблиць.

Зв'язок роботи з науковими програмами, планами, темами.

Дисертаційна робота виконувалася в рамках пріоритетного напрямку «Інформаційні та комунікаційні технології» відповідно до закону України «Про пріоритетні напрями розвитку науки і техніки» та пов’язана з науковими дослідженнями, які виконувалися за держбюджетною тематикою кафедри комп’ютерних наук Західноукраїнського національного університету. Основні результати дисертаційного дослідження отримано в межах виконання наступних тем:

- держбюджетного прикладного дослідження на тему: «Інформаційна технологія забезпечення ефективної акумуляції електроенергії гібридною

електростанцією» (державний реєстраційний номер 0126U002209), у якому автором запропоновано та обґрунтовано метод онтологічно-керованого конфігурування корпоративної комп'ютерної мережі, який ґрунтується на семантичному поданні знань про мережеві ресурси, сервіси та множину задач, що розв'язуються в мережі, та, на відміну від традиційних підходів до конфігурування, забезпечує автоматизоване формування конфігурацій на основі механізмів логічного виведення.

Виконання роботи здійснювалося за безпосередньої участі автора та з його особистим внеском у отриманні результати.

Практичне значення отриманих результатів полягає у розробленні знання-орієнтованої програмної системи управління корпоративними комп'ютерними мережами, яка забезпечує автоматизацію процесів моніторингу, конфігурування мережевих ресурсів, розподілу навантаження та підтримки прийняття рішень на основі онтологічного представлення знань про структуру мережі, її ресурси, сервіси та множину задач, що розв'язуються в мережевому середовищі.

Теоретичні та прикладні результати дисертаційної роботи використано:

- у ПП «Колумбус» для конфігурування корпоративної комп'ютерної мережі (акт про впровадження результатів дисертаційної роботи від 12 травня 2026 р.);
- в Центрі інформаційних технологій Західноукраїнського національного університету в процесі реалізації програмних систем, які включають інтеграцію знання-орієнтованого підходу з використанням онтологій (акт про впровадження результатів дисертаційної роботи від 11 травня 2026 р.);
- в НДЧ ЗУНУ при виконанні теми:
 - «Інформаційна технологія забезпечення ефективної акумуляції електроенергії гібридною електростанцією» (державний реєстраційний номер 0126U002209);
- при виконанні міжнародних проєктів:

- «TALKING HEADS / Спілкування з мистецтвом минулого» № 101174331-CREA-CULT-2024-COOP-2;
- «Мультифункціональне Фермерство / Multifunctionality Farming (MyFarm) 101140288 ERASMUS-EDU-2023-PI-ALL-INNO-EDU-ENTERP;
- в освітньому процесі Західноукраїнського національного університету на кафедрі комп'ютерних наук при викладанні дисциплін: «Web-онтології», «Архітектура та проектування програмного забезпечення» (акт про впровадження в освітній процес від 10 червня 2026 р.)

У додатку Д наведено акти впровадження та використання результатів дисертаційного дослідження.

Декларація про використання ШІ. Під час підготовки роботи автор використовував Gemini 3 Pro та OpenAI GPT-5 для пошуку та первинного опрацювання літературних джерел, оформлення ілюстрації, а також виявлення та виправлення граматичних, орфографічних та стилістичних помилок. Після застосування цих інструментів автор провів ретельний перегляд і вніс необхідні зміни, беручи на себе повну відповідальність за остаточний зміст дисертаційної роботи.

РОЗДІЛ 1

АНАЛІЗ МЕТОДІВ ТА ПРОГРАМНИХ ЗАСОБІВ ДЛЯ ПОБУДОВИ ЗНАННЯ-ОРІЄНТОВАНИХ ПРОГРАМНИХ СИСТЕМ

Ефективне управління сучасними корпоративними комп'ютерними мережами є одним із ключових чинників забезпечення надійного функціонування інформаційної інфраструктури організацій. Постійне зростання кількості мережевих ресурсів, сервісів, користувачів та прикладних задач призводить до ускладнення процесів адміністрування, конфігурування та підтримки працездатності мережевого середовища. У таких умовах традиційні підходи до управління мережами, які базуються переважно на аналізі окремих параметрів функціонування обладнання та ручному прийнятті рішень, дедалі частіше виявляються недостатньо ефективними.

Сучасні тенденції розвитку інформаційних технологій свідчать про поступовий перехід від управління окремими ресурсами до управління знаннями про структуру мережі, її сервіси, політики безпеки та множину задач, що розв'язуються у корпоративному середовищі. Особливого значення при цьому набувають знання-орієнтовані підходи, які дозволяють формалізувати предметну область, автоматизувати процеси прийняття рішень та забезпечити адаптацію мережевої інфраструктури до змін умов функціонування.

У цьому розділі проведено аналіз особливостей функціонування корпоративних комп'ютерних мереж, досліджено сучасні методи та програмні засоби їх управління, розглянуто основні підходи до формалізації знань, а також проаналізовано можливості використання онтологічних моделей, методів штучного інтелекту, машинного навчання та логічного виведення для підтримки процесів управління мережевою інфраструктурою. На основі виконаного аналізу сформульовано науково-технічну проблему дослідження, обґрунтовано необхідність застосування онтологічного підходу та визначено мету, задачі й основні напрями подальших досліджень. Основні результати цього розділу опубліковані в працях [27, 65, 71, 72, 88, 94, 116, 117].

1.1 Особливості функціонування та управління корпоративними комп'ютерними мережами

Сучасний етап цифрової трансформації організацій супроводжується стрімким зростанням обсягів інформації, кількості мережевих сервісів та рівня інтеграції програмних систем. У таких умовах корпоративні комп'ютерні мережі перетворюються із засобу забезпечення обміну даними на складні розподілені інформаційні середовища, від ефективності функціонування яких безпосередньо залежить стабільність бізнес-процесів, якість обслуговування користувачів та конкурентоспроможність організацій. Корпоративна мережа фактично виступає цифровою інфраструктурою підприємства, забезпечуючи взаємодію між користувачами, прикладними сервісами, базами даних, засобами інформаційної безпеки та зовнішніми інформаційними ресурсами [18-20, 62, 70, 73].

На відміну від традиційних локальних мереж, сучасні корпоративні комп'ютерні мережі характеризуються значною територіальною розподіленістю, багаторівневою структурою управління та високим ступенем динамічності. Такі мережі об'єднують тисячі користувачів, сотні серверів та десятки програмних сервісів, які функціонують у режимі безперервної взаємодії. З огляду на це, процес управління корпоративною мережею поступово переходить від адміністрування окремих мережевих пристроїв до управління складною сукупністю взаємопов'язаних ресурсів, сервісів та задач [94, 100].

Основу корпоративної мережі формують мережеві ресурси, до яких належать серверне обладнання, комунікаційні пристрої, канали зв'язку, системи зберігання даних та програмні сервіси (рисунок 1.1.). Ефективність функціонування такої інфраструктури визначається не лише характеристиками окремих компонентів, але й якістю їхньої взаємодії. У сучасних умовах навіть незначне перевантаження одного з вузлів може призвести до каскадного погіршення роботи великої кількості сервісів, що особливо актуально для корпоративних інформаційних систем критичного призначення.

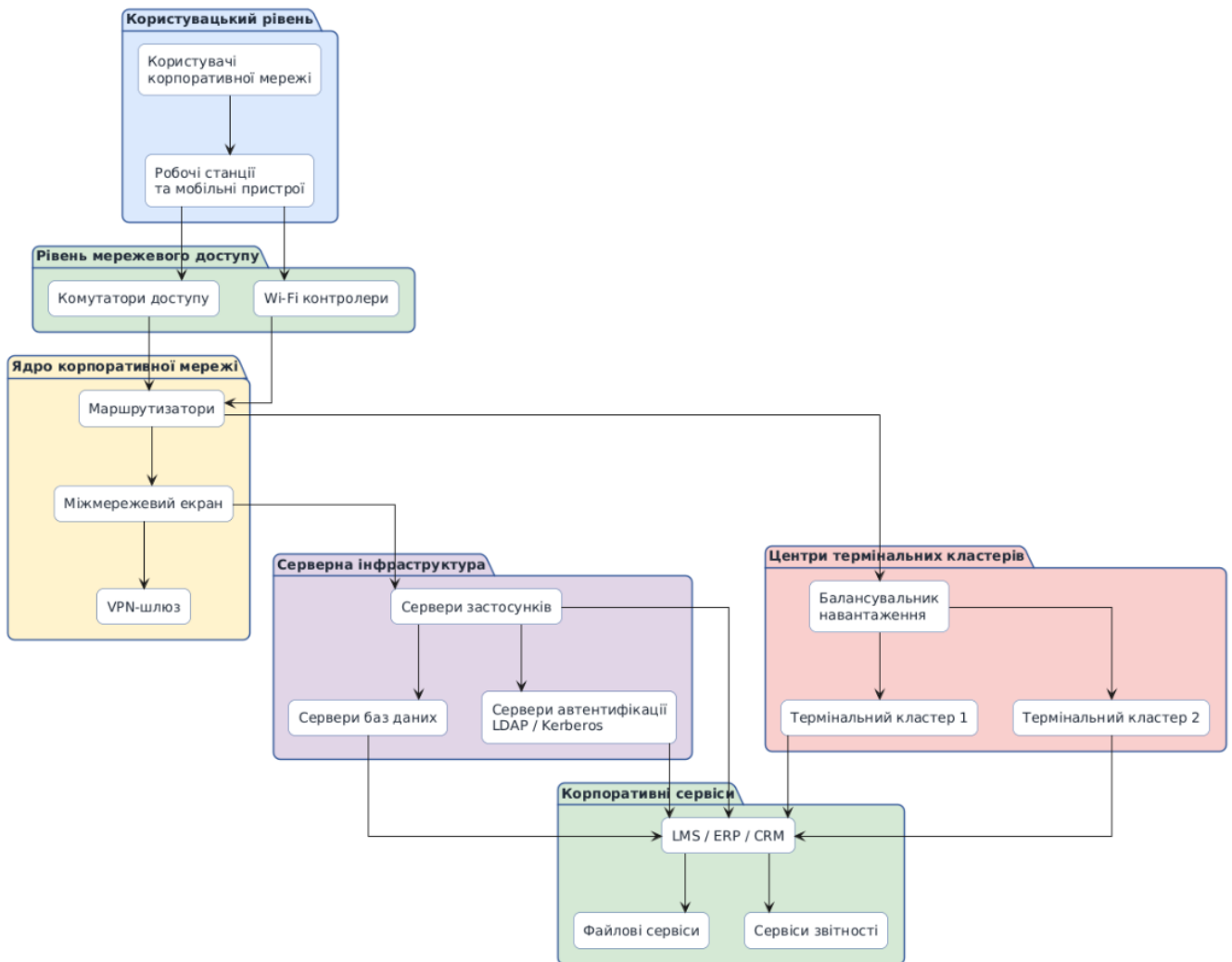


Рис. 1.1. Узагальнена структура корпоративної комп'ютерної мережі

Важливу роль у функціонуванні корпоративних мереж відіграють центри обробки даних та термінальні кластери. Саме вони забезпечують централізоване виконання обчислювальних задач, обслуговування користувачських запитів та підтримку корпоративних сервісів. У класичній архітектурі корпоративної мережі центр обробки даних виконує функції зберігання інформації, виконання прикладних програм та адміністрування ресурсів, тоді як термінальні кластери забезпечують масштабованість і балансування навантаження між обчислювальними вузлами [32, 36].

Проте зі збільшенням кількості користувачів та сервісів виникає проблема нерівномірного використання ресурсів. На практиці окремі сервери можуть функціонувати на межі своїх можливостей, тоді як інші ресурси залишаються

недостатньо завантаженими. Це призводить до збільшення часу відгуку сервісів, погіршення якості обслуговування користувачів та зниження загальної ефективності функціонування мережевої інфраструктури.

Одним із ключових завдань адміністрування корпоративних комп'ютерних мереж є забезпечення раціонального використання ресурсів та підтримання необхідного рівня доступності сервісів. Традиційно до основних задач адміністрування належать моніторинг стану мережі, управління конфігураціями обладнання, контроль доступу користувачів, забезпечення інформаційної безпеки, балансування навантаження та відновлення працездатності після відмов. Проте сучасні тенденції розвитку інформаційних технологій свідчать про необхідність переходу від реактивного адміністрування до проактивного управління, яке базується на прогнозуванні станів мережі та автоматизованому прийнятті рішень, рисунок 1.2.



Рис. 1.2. Основні проблеми управління корпоративною мережею та напрям їх розв'язання

Особливої актуальності набуває проблема масштабованості корпоративних мереж. З одного боку, зростання кількості користувачів та сервісів вимагає постійного розширення інфраструктури. З іншого боку, збільшення кількості мережевих компонентів суттєво ускладнює процеси управління та підвищує ризик виникнення помилок конфігурації. У результаті виникає суперечність між необхідністю забезпечення гнучкості мережі та складністю її адміністрування.

Додаткові труднощі створює динамічний характер навантаження. Упродовж функціонування корпоративної мережі інтенсивність використання ресурсів постійно змінюється під впливом різних факторів: активності користувачів, запуску нових сервісів, виконання ресурсоємних задач або аварійних ситуацій. Тому статичні схеми розподілу ресурсів втрачають ефективність і потребують заміни адаптивними механізмами управління [107, 109].

Аналіз сучасних програмних засобів управління корпоративними мережами свідчить, що більшість існуючих рішень орієнтована переважно на моніторинг технічних характеристик мережевого обладнання та збір статистичних показників функціонування, рисунок 1.3. Незважаючи на значний рівень автоматизації, такі системи здебільшого не враховують семантичні взаємозв'язки між ресурсами, сервісами та задачами, що розв'язуються в мережевому середовищі. У результаті процес прийняття управлінських рішень значною мірою залишається залежним від досвіду та кваліфікації адміністратора [20, 30, 32].

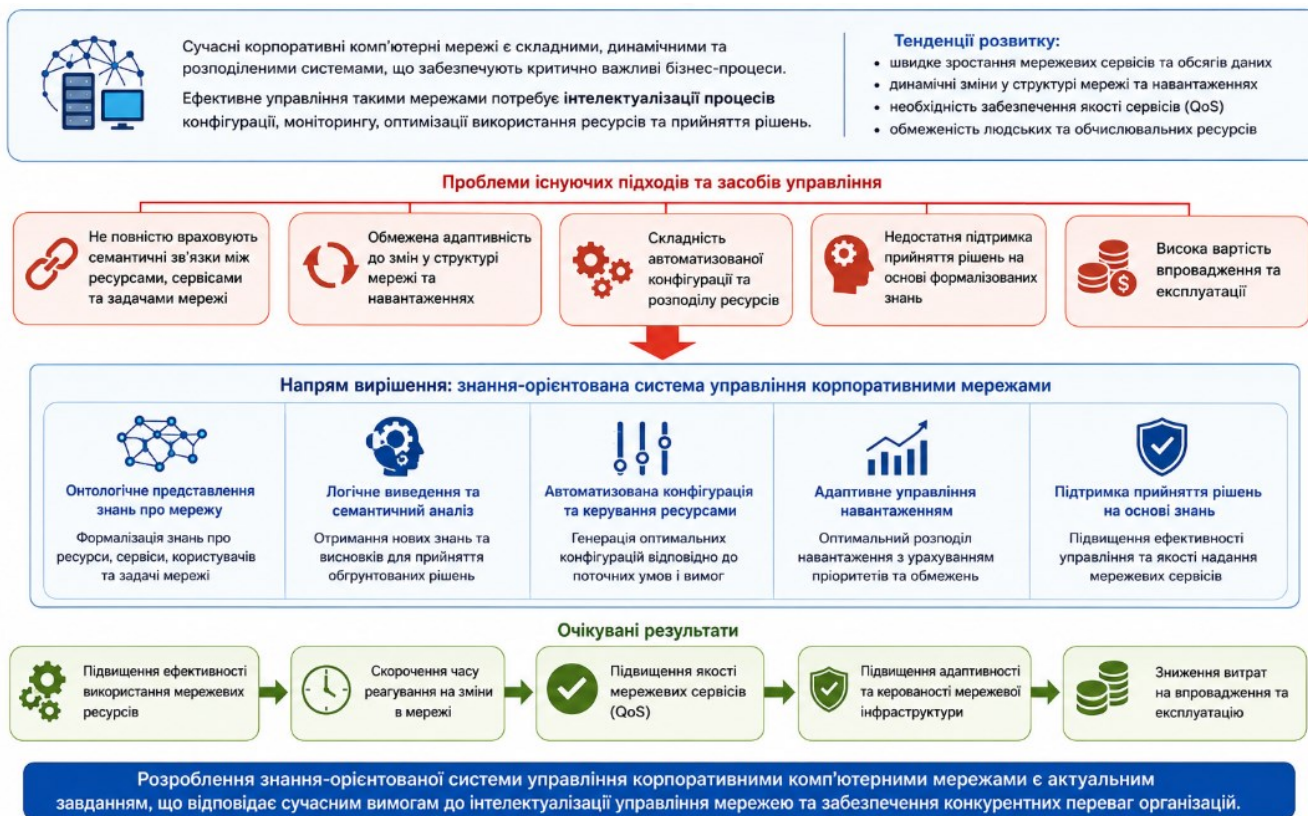


Рис. 1.3. Проблеми відомих підходів та засобів управління

У контексті зазначених проблем особливого значення набуває використання знання-орієнтованих підходів до управління корпоративними комп'ютерними мережами. Їх головна ідея полягає у формалізації знань про структуру мережі, її ресурси, сервіси, політики безпеки та множину задач у вигляді семантичних моделей, придатних для автоматизованої обробки. На відміну від традиційних інформаційних систем, знання-орієнтовані системи дозволяють не лише накопичувати інформацію про стан мережі, але й виконувати логічне виведення, формувати рекомендації та підтримувати процеси прийняття рішень.

Таким чином, сучасні корпоративні комп'ютерні мережі є складними динамічними системами, ефективне функціонування яких потребує інтеграції засобів моніторингу, конфігурування, оптимізації та підтримки прийняття рішень. Проведений аналіз свідчить про доцільність використання знання-орієнтованих підходів, які забезпечують можливість формалізації знань про мережеву інфраструктуру та створюють передумови для автоматизації процесів управління ресурсами, сервісами та задачами корпоративного середовища.

1.2 Аналіз сучасних методів управління корпоративними комп'ютерними мережами

Ускладнення архітектури корпоративних комп'ютерних мереж, збільшення кількості користувачів, сервісів та інформаційних ресурсів призводить до постійного зростання вимог до ефективності процесів управління мережевою інфраструктурою. На відміну від локальних інформаційних систем, сучасні корпоративні мережі функціонують як складні розподілені динамічні середовища, у яких одночасно відбуваються процеси маршрутизації даних, балансування навантаження, забезпечення інформаційної безпеки, підтримки сервісів та взаємодії великої кількості користувачів. У таких умовах управління мережею перетворюється на багатокритеріальну задачу, розв'язання якої потребує комплексного врахування технічних, функціональних та організаційних факторів

[36, 54, 60, 61, 62, 107, 109]. На рисунку 1.4 представлено методи управління корпоративними мережами.

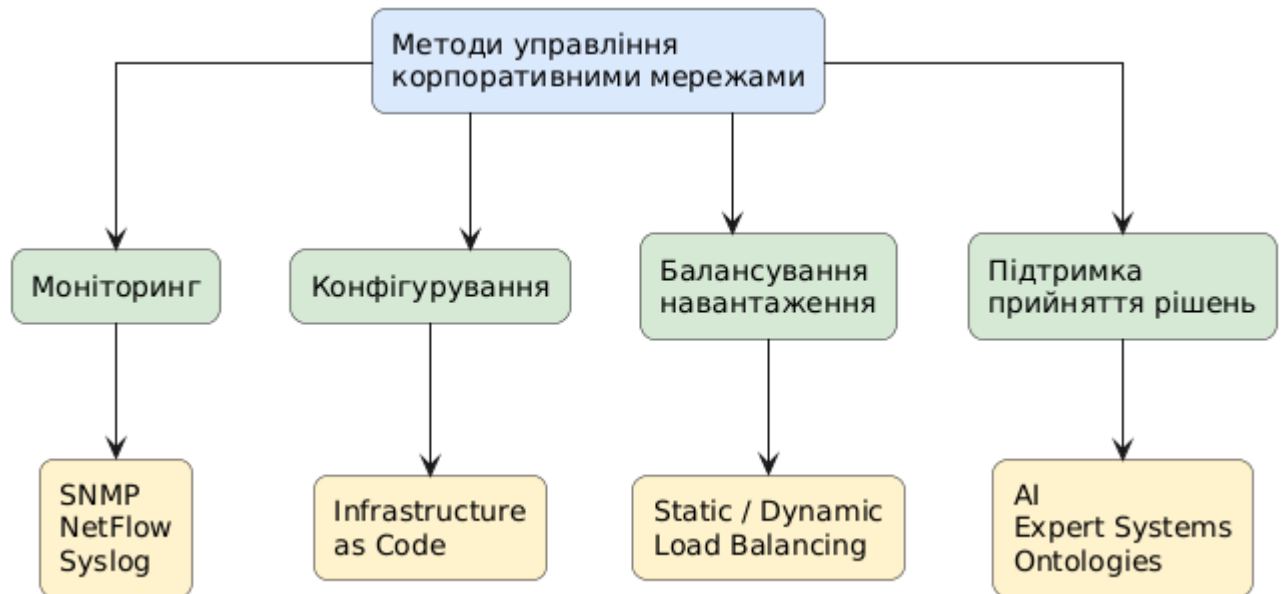


Рис. 1.4. Методи управління корпоративними мережами

Централізовані та децентралізовані підходи до управління. Історично найбільш поширеним підходом до управління корпоративними мережами був централізований підхід, відповідно до якого всі функції моніторингу, конфігурування та прийняття рішень виконуються єдиним центром управління. Перевагами такого підходу є наявність цілісного бачення стану мережі, спрощення адміністрування та можливість реалізації єдиних політик безпеки [18, 19, 67, 69].

Разом з тим централізовані системи мають низку суттєвих обмежень. Зі збільшенням масштабів мережі зростає обсяг інформації, яку необхідно обробляти центральному вузлу управління, що може призводити до перевантаження керуючого центру та збільшення часу прийняття рішень. Крім того, виникає проблема єдиної точки відмови, яка може негативно впливати на надійність функціонування всієї мережевої інфраструктури.

Альтернативою є децентралізовані та розподілені підходи до управління, які передбачають делегування частини функцій окремим сегментам мережі або

локальним вузлам. Такі рішення забезпечують кращу масштабованість та підвищують відмовостійкість системи. Проте їх застосування ускладнює координацію між компонентами мережі та створює додаткові вимоги до узгодженості прийнятих рішень.

У сучасних корпоративних мережах переважно використовуються гібридні архітектури, які поєднують переваги централізованого стратегічного управління та децентралізованого оперативного реагування на зміни стану ресурсів [74, 82, 92].

Методи моніторингу мережевих ресурсів. Моніторинг є базовою складовою будь-якої системи управління мережею, оскільки забезпечує отримання інформації про поточний стан ресурсів та сервісів. Основною метою моніторингу є своєчасне виявлення відхилень від нормального режиму функціонування та формування інформаційної основи для подальшого прийняття управлінських рішень.

Традиційні методи моніторингу базуються на використанні протоколів SNMP, NetFlow, Syslog та ICMP. Вони забезпечують збір статистичної інформації про навантаження процесорів, використання пам'яті, пропускну здатність каналів зв'язку, кількість активних з'єднань та інші показники функціонування мережевої інфраструктури [30, 32, 36, 54, 60, 61, 62, 70].

Незважаючи на значну поширеність таких підходів, вони мають низку обмежень. Насамперед більшість існуючих систем оперують окремими числовими показниками та не враховують контекст функціонування мережі. У результаті адміністратор отримує значний обсяг даних, але змушений самостійно інтерпретувати їх та визначати причинно-наслідкові зв'язки між подіями. У зв'язку з цим останніми роками активно розвиваються інтелектуальні системи моніторингу, які доповнюють традиційний збір даних механізмами аналізу залежностей між ресурсами, сервісами та користувацькими задачами, рисунок 1.5.

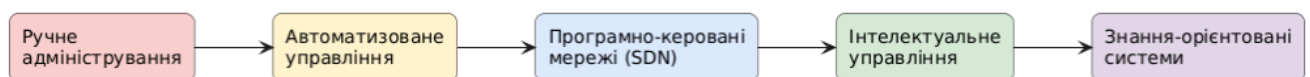


Рис. 1.5. Розвиток підходів до управління корпоративними мережами

Методи конфігурування мережевої інфраструктури. Одним із найскладніших аспектів управління корпоративною мережею є конфігурування її компонентів. Конфігурація визначає параметри функціонування мережевих пристроїв, серверів, сервісів, політик безпеки та механізмів взаємодії між ними. Традиційно процес конфігурування здійснювався вручну адміністраторами мережі. Такий підхід є достатньо гнучким, проте характеризується високою ймовірністю помилок та значними часовими витратами. Зі збільшенням складності корпоративних мереж ручне конфігурування стає дедалі менш ефективним [60, 84, 99, 100].

Подальший розвиток отримали автоматизовані засоби конфігурування, зокрема системи Ansible, Puppet, Chef та SaltStack, які реалізують принцип Infrastructure as Code. Використання таких технологій дозволяє автоматизувати розгортання та підтримку мережевої інфраструктури, забезпечуючи відтворюваність конфігурацій.

Разом з тим більшість сучасних засобів автоматизації не здійснює аналізу семантичної сумісності ресурсів та задач. Вони автоматизують виконання команд, але не забезпечують інтелектуальної підтримки прийняття конфігураційних рішень. Саме ця проблема створює передумови для використання онтологічного подання знань та механізмів логічного виведення.

Методи балансування навантаження та розподілу ресурсів. Зростання інтенсивності використання корпоративних сервісів призводить до необхідності ефективного розподілу навантаження між обчислювальними ресурсами. Основною метою балансування є забезпечення рівномірного використання ресурсів та запобігання перевантаженню окремих вузлів мережі [103, 104].

Найпростішими є статичні методи балансування, які використовують наперед визначені правила розподілу запитів. До таких методів належать Round Robin, Weighted Round Robin та Random Distribution. Перевагою цих алгоритмів є простота реалізації, однак вони не враховують поточний стан ресурсів та особливості виконуваних задач. Більш ефективними є динамічні методи балансування, які використовують інформацію про завантаження серверів,

інтенсивність потоків даних та доступні обчислювальні ресурси. До цієї групи належать Least Connections, Least Response Time та Resource-Based Load Balancing. Однак навіть сучасні динамічні методи орієнтовані переважно на технічні характеристики ресурсів. Вони практично не враховують зміст задач, політики безпеки та взаємозв'язки між сервісами. Як наслідок, оптимальне з точки зору навантаження рішення може виявитися неефективним з позиції бізнес-процесів або вимог інформаційної безпеки [2, 59, 54, 60].

Інтелектуальні методи підтримки прийняття рішень у мережевому адмініструванні. Останніми роками спостерігається активне впровадження методів штучного інтелекту та аналізу знань у сферу мережевого адміністрування. Основною метою таких підходів є автоматизація процесів прийняття рішень та зменшення залежності ефективності управління від людського фактора. Серед найбільш перспективних напрямів можна виділити експертні системи, машинне навчання, нечітку логіку, багатокритеріальну оптимізацію та онтологічне моделювання. Використання цих підходів дозволяє не лише аналізувати поточний стан мережі, але й прогнозувати майбутні зміни, виявляти приховані закономірності та формувати рекомендації щодо управління ресурсами [12, 17, 26, 93, 94].

Особливий інтерес становлять знання-орієнтовані системи, які базуються на формалізації знань про структуру мережі, сервіси, ресурси та множину задач. На відміну від класичних систем моніторингу, такі рішення дозволяють виконувати логічне виведення, автоматично перевіряти відповідність конфігурацій вимогам задач та формувати пояснювані рекомендації щодо управління мережею.

Проведений аналіз свідчить, що існуючі методи управління корпоративними комп'ютерними мережами забезпечують високий рівень автоматизації окремих функцій, проте переважно орієнтовані на роботу з локальними показниками стану ресурсів, таблиця 1.1. Порівняльний аналіз показує, що традиційні централізовані та децентралізовані підходи забезпечують базові можливості управління мережевою інфраструктурою, однак не підтримують формалізоване представлення

знань про мережеве середовище та не забезпечують пояснюваність процесу прийняття рішень. Інтелектуальні системи частково усувають ці недоліки за рахунок використання алгоритмів штучного інтелекту, проте переважно працюють як «чорна скринька». Найбільш перспективними є знання-орієнтовані системи, які поєднують механізми логічного виведення, онтологічне представлення знань та багатокритеріальну оптимізацію. Саме цей підхід покладено в основу даного дисертаційного дослідження, оскільки він дозволяє враховувати взаємозв'язки між ресурсами, сервісами та множиною задач, що розв'язуються у корпоративній комп'ютерній мережі.

Таблиця 1.1. Порівняльний аналіз підходів до управління корпоративними комп'ютерними мережами

Характеристика	Централізовані системи	Децентралізовані системи	Інтелектуальні системи	Знання-орієнтовані системи
Масштабованість	Середня	Висока	Висока	Висока
Відмовостійкість	Низька	Висока	Висока	Висока
Автоматизація конфігурування	Часткова	Часткова	Висока	Висока
Врахування контексту задач	Ні	Ні	Частково	Так
Використання знань про мережу	Ні	Ні	Частково	Так
Логічне виведення	Ні	Ні	Обмежене	Так
Пояснюваність рішень	Низька	Низька	Середня	Висока
Підтримка багатокритеріальної оптимізації	Низька	Середня	Висока	Висока
Адаптація до змін середовища	Низька	Середня	Висока	Висока

Недостатньо дослідженими залишаються питання інтеграції формалізованих знань про структуру мережі, сервіси та множину задач із процесами конфігурування, оптимізації та підтримки прийняття рішень. Це обумовлює доцільність розроблення знання-орієнтованої програмної системи управління корпоративними комп'ютерними мережами, яка використовує онтологічне

представлення знань та механізми логічного виведення для підвищення ефективності управління мережевою інфраструктурою.

1.3 Аналіз сучасних програмних засобів управління корпоративними мережами

Розвиток корпоративних комп'ютерних мереж супроводжується постійним збільшенням кількості мережевих пристроїв, серверів, сервісів та користувачів. За таких умов ефективне управління мережевою інфраструктурою стає неможливим без використання спеціалізованих програмних платформ, які забезпечують моніторинг ресурсів, автоматизацію конфігурування, підтримку інформаційної безпеки та аналіз продуктивності мережі [96, 97, 62, 70, 73].

Сучасний ринок програмних засобів управління корпоративними мережами представлений великою кількістю рішень, серед яких особливого поширення набули Cisco DNA Center, SolarWinds Network Performance Monitor, ManageEngine OpManager, PRTG Network Monitor та Microsoft System Center, рисунок 1.6. Незважаючи на високий рівень функціональності, більшість із цих систем орієнтована переважно на управління технічними параметрами мережевої інфраструктури та практично не використовує формалізоване подання знань про структуру мережі, сервіси та множину задач.

Cisco DNA Center. Одним із найбільш функціонально розвинених рішень для управління корпоративними мережами є Cisco DNA Center. Дана платформа реалізує концепцію програмно-керованих мереж (Software Defined Networking) та забезпечує централізоване управління мережевою інфраструктурою підприємства. Основними функціональними можливостями Cisco DNA Center є автоматизоване розгортання мережевих пристроїв, централізоване конфігурування обладнання, моніторинг продуктивності, управління політиками доступу та аналітика функціонування мережі. Платформа активно використовує механізми машинного навчання для виявлення аномалій та прогнозування потенційних проблем [74, 82].

Разом із тим Cisco DNA Center орієнтована переважно на обладнання Cisco

та використовує закриту модель представлення знань про мережеве середовище. Це обмежує можливості інтеграції з неоднорідними мережевими інфраструктурами та ускладнює реалізацію універсальних механізмів логічного виведення.

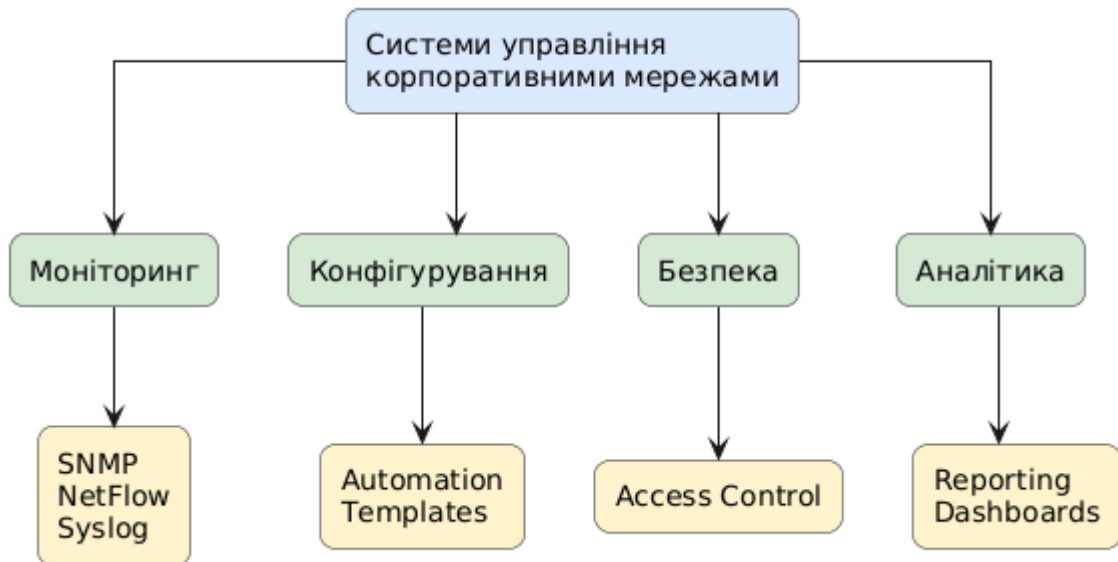


Рис. 1.6. Основні функціональні компоненти сучасних систем управління корпоративними мережами

SolarWinds Network Performance Monitor. Система SolarWinds Network Performance Monitor належить до класу спеціалізованих платформ моніторингу корпоративних мереж та орієнтована на збір і аналіз показників функціонування мережевих ресурсів. Перевагою даного рішення є розвинуті засоби візуалізації мережевої інфраструктури, автоматичне виявлення пристроїв, підтримка протоколів SNMP, NetFlow та Syslog, а також потужні механізми формування звітів. Система забезпечує оперативне виявлення перевантажень та збоїв у роботі мережевого обладнання [90, 91, 96, 97].

Основним недоліком є відсутність механізмів семантичного аналізу структури мережі. Прийняття управлінських рішень значною мірою залишається залежним від адміністратора, а інформація про взаємозв'язки між ресурсами, сервісами та задачами не використовується під час аналізу ситуацій.

ManageEngine OpManager. Платформа ManageEngine OpManager призначена

для моніторингу та адміністрування мережевої інфраструктури середніх і великих підприємств. До переваг системи належать підтримка багатовиробничого обладнання, автоматичне виявлення пристроїв, контроль доступності сервісів, моніторинг серверів та віртуалізованих середовищ. Крім того, система підтримує механізми кореляції подій та формування аналітичних звітів. Разом із тим можливості автоматичного прийняття рішень залишаються обмеженими. Система не підтримує онтологічне представлення знань і не забезпечує побудову логічних висновків на основі структури мережі та бізнес-процесів організації [36, 54].

PRTG Network Monitor. Система PRTG Network Monitor є універсальним засобом моніторингу мережевих ресурсів, серверів, сервісів та прикладних програм. Особливістю PRTG є використання концепції сенсорів, кожен з яких відповідає за контроль певного параметра мережевого середовища. Такий підхід забезпечує високу гнучкість моніторингу та можливість інтеграції з різноманітними інформаційними системами. Недоліком є орієнтація переважно на контроль технічних параметрів функціонування інфраструктури. Система не забезпечує семантичного аналізу мережевого середовища та не підтримує автоматизоване конфігурування на основі знань про структуру мережі [34].

Microsoft System Center. Комплекс Microsoft System Center являє собою набір програмних засобів для централізованого управління корпоративною ІТ-інфраструктурою. Система забезпечує моніторинг серверів, управління конфігураціями, автоматизацію розгортання програмного забезпечення, адміністрування користувачів та підтримку віртуалізованих середовищ. Особливу увагу приділено інтеграції з корпоративною екосистемою Microsoft. Водночас функціональність платформи значною мірою орієнтована на інфраструктури, побудовані на продуктах Microsoft. Засоби логічного виведення та семантичного аналізу мережевого середовища реалізовані лише частково. У таблиці 1.2 представлено аналіз підходів до управління корпоративними комп'ютерними мережами [96, 97].

Таблиця 1.2. Порівняльний аналіз підходів до управління корпоративними комп'ютерними мережами

Функціональність	Cisco DNA Center	SolarWinds NPM	OpManager	PRTG	System Center	Запропонована система
Моніторинг ресурсів	+	+	+	+	+	+
Автоматизоване конфігурування	+	–	±	–	+	+
Балансування навантаження	+	±	±	–	±	+
Централізоване управління	+	+	+	+	+	+
Інтеграція з політиками безпеки	+	±	+	±	+	+
Логічне виведення	–	–	–	–	–	+
Онтологічне представлення знань	–	–	–	–	–	+
Підтримка множини задач	–	–	–	–	–	+
Пояснюваність рішень	–	–	–	–	–	+
Багатокритеріальна оптимізація	–	–	–	–	–	+

Проведений аналіз показує, що існуючі програмні платформи забезпечують високий рівень автоматизації процесів моніторингу та адміністрування мережевої інфраструктури. Проте жодна з розглянутих систем не використовує онтологічне представлення знань про структуру мережі, ресурси, сервіси та множину задач, а також не підтримує механізми логічного виведення для формування пояснюваних управлінських рішень. Саме усунення цих обмежень є основною метою розроблення знання-орієнтованої програмної системи управління корпоративними комп'ютерними мережами, запропонованої в даній дисертаційній роботі.

1.4 Аналіз знання-орієнтованих підходів та онтологічних моделей

Зростання складності корпоративних комп'ютерних мереж, збільшення кількості інформаційних ресурсів, сервісів та користувачів призводить до необхідності переходу від традиційних методів адміністрування до

інтелектуальних систем управління, здатних враховувати не лише поточні параметри функціонування мережевої інфраструктури, але й знання про її структуру, закономірності функціонування та взаємозв'язки між ресурсами, сервісами та задачами. У таких умовах особливого значення набувають знання-орієнтовані підходи, які забезпечують формалізацію знань про предметну область та підтримують автоматизоване прийняття рішень [12, 17, 26, 29, 33].

На відміну від традиційних інформаційних систем, що оперують переважно даними, знання-орієнтовані системи працюють із формалізованими знаннями, які містять не лише факти, але й правила, обмеження, причинно-наслідкові зв'язки та закономірності функціонування досліджуваної предметної області. Це дозволяє виконувати логічне виведення нових знань та підтримувати процеси прийняття рішень у складних динамічних середовищах. У загальному випадку знання-орієнтовану систему можна представити як

$$KBS = \langle KB, WM, IE, UI \rangle, \quad (1.1)$$

де *KB* - база знань, *WM* - робоча пам'ять, *IE* - механізм логічного виведення, *UI* - користувацький інтерфейс.

Підходи до формалізації знань. Ключовою проблемою побудови знання-орієнтованих систем є вибір способу формалізації знань. Від обраної моделі залежить можливість накопичення, аналізу та використання знань у процесі прийняття рішень. У сучасних інформаційних системах використовуються декілька основних підходів до представлення знань, рисунок 1.7:

- логічні моделі;
- продукційні системи;
- семантичні мережі;
- фреймові структури;
- онтологічні моделі.

Загальну структуру знань можна подати у вигляді

$$K = \langle F, R, C \rangle, \quad (1.2)$$

де F - множина фактів, R - множина відношень між фактами, C - множина обмежень та правил предметної області.

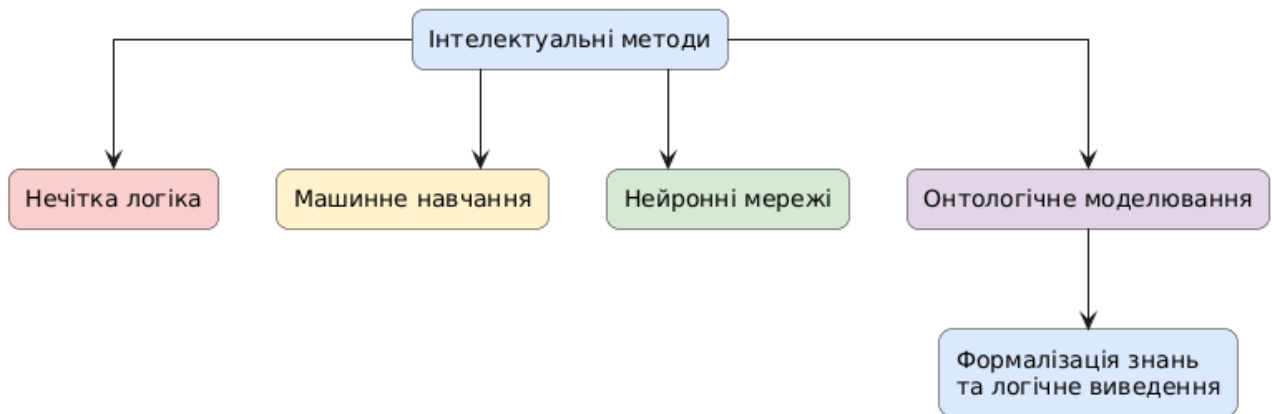


Рис. 1.7. Місце онтологічного підходу серед інтелектуальних методів

Логічні моделі. Логічні моделі базуються на апараті математичної логіки та використовують предикати для опису властивостей об'єктів. Для мережевого середовища правило може бути подане у вигляді

$$\text{Server}(x) \wedge \text{HighLoad}(x) \rightarrow \text{Reconfigure}(x). \quad (1.3)$$

Така модель забезпечує високий рівень формалізації, однак зі збільшенням кількості об'єктів та правил складність підтримки бази знань суттєво зростає.

Продукційні системи. У продукційних системах знання представляються правилами типу, рисунок 1.8.

```

IF CPU_Load > 90%
AND Memory_Load > 85%
THEN Redistribute_Load
  
```

Рис. 1.8. Приклад продукційних правил

Такі моделі широко використовуються в експертних системах, однак недостатньо ефективно відображають складні взаємозв'язки між мережевими об'єктами.

Семантичні мережі. Семантичні мережі представляють знання у вигляді графових структур

$$SN = \langle V, E \rangle, \quad (1.4)$$

де V - множина вершин, а E - множина зв'язків між ними. Перевагою такого підходу є наочність представлення взаємозв'язків між ресурсами, сервісами та користувачами.

Фреймові моделі. Фреймова модель розглядає кожен об'єкт як сукупність слотів:

$$\text{Frame} = \langle S_1, S_2, \dots, S_n \rangle. \quad (1.5)$$

Наприклад:

$$\text{Server} = \{CPU, \text{Memory}, \text{OS}, \text{Services}, \text{Location} \}. \quad (1.6)$$

Такі моделі добре підходять для структурованого опису складних об'єктів, однак мають обмежені можливості щодо логічного виведення.

Інтелектуальні підходи до представлення знань [45, 52, 55, 73]. Поряд із класичними моделями формалізації знань останніми роками активно розвиваються інтелектуальні підходи, які використовують методи штучного інтелекту для аналізу даних та підтримки прийняття рішень.

Нечітка логіка [52, 55, 75, 78, 79]. Нечітка логіка дозволяє працювати з неповною та невизначеною інформацією. Основою такого підходу є функція належності

$$\mu_A(x): X \rightarrow [0,1]. \quad (1.7)$$

Наприклад, завантаження сервера може характеризуватися не лише категоріями «низьке» або «високе», а й проміжними значеннями. Перевагою нечітких систем є робота в умовах невизначеності. Проте вони не забезпечують формалізованого опису структури предметної області та складних семантичних залежностей.

Машинне навчання [86, 90, 95, 102, 110, 113]. Методи машинного навчання використовуються для автоматичного пошуку закономірностей у даних та побудови прогностичних моделей. У загальному випадку задача машинного навчання описується як пошук функції

$$f: X \rightarrow Y, \quad (1.8)$$

де X - множина ознак, Y - множина результатів.

У корпоративних мережах такі методи застосовуються для:

- прогнозування навантаження;
- виявлення аномалій;
- класифікації інцидентів;
- прогнозування відмов обладнання.

Проте моделі машинного навчання не містять формалізованих знань про структуру мережі та зазвичай не пояснюють причини отриманих результатів.

Штучні нейронні мережі [6, 7, 13, 14, 25, 43]. Особливого поширення набули штучні нейронні мережі, які демонструють високу ефективність у задачах прогнозування та аналізу великих обсягів мережевих даних. Математично функціонування нейрона визначається виразом

$$y = \varphi(\sum_{i=1}^n w_i x_i + b), \quad (1.8)$$

де x_i - вхідні сигнали, w_i - вагові коефіцієнти, b - зміщення, φ - функція активації.

Нейронні мережі успішно використовуються для виявлення аномалій мережевого трафіку, прогнозування навантаження серверів та класифікації мережевих подій.

Водночас вони функціонують як моделі типу «чорна скринька», що суттєво обмежує можливість пояснення прийнятих рішень і створює труднощі при їх використанні в системах управління критичною інфраструктурою.

Онтологічні моделі представлення знань [1, 3, 4, 5, 10, 15, 16, 22, 23]. Найбільш перспективним підходом до побудови знання-орієнтованих систем управління складними інформаційними середовищами є онтологічне моделювання. Онтологія забезпечує одночасне представлення:

- концептів предметної області;
- властивостей об'єктів;
- відношень між об'єктами;
- аксіом та обмежень;
- правил логічного виведення.

Формально онтологія задається у вигляді

$$O = \langle C, R, A, I \rangle, \quad (1.9)$$

де C - множина концептів; R - множина відношень; A - множина аксіом; I - множина екземплярів.

На відміну від нейронних мереж та алгоритмів машинного навчання, онтології забезпечують не лише накопичення знань, але й можливість пояснюваного логічного виведення [93, 96, 97, 98, 101, 102]. Крім того, вони дозволяють інтегрувати знання про структуру мережі, ресурси, сервіси, політики безпеки та задачі в єдиному інформаційному просторі. У таблиці 1.3. наведено порівняльний аналіз підходів до формалізації знань

Таблиця 1.3. Порівняльний аналіз підходів до формалізації знань

Підхід	Формалізація знань	Логічне виведення	Пояснюваність	Робота з невизначеністю	Масштабованість
Логічні моделі	Висока	Висока	Висока	Низька	Низька
Продукційні системи	Середня	Висока	Висока	Низька	Середня
Семантичні мережі	Висока	Низька	Висока	Низька	Середня
Фрейми	Середня	Середня	Висока	Низька	Середня
Нечітка логіка	Середня	Середня	Середня	Висока	Висока
Машинне навчання	Низька	Низька	Низька	Висока	Висока
Нейронні мережі	Низька	Низька	Низька	Висока	Дуже висока
Онтології	Висока	Висока	Висока	Середня	Висока

Проведений аналіз свідчить, що хоча методи машинного навчання, нейронні мережі та нечітка логіка ефективно розв'язують задачі прогнозування, класифікації та роботи з невизначеністю, вони не забезпечують повноцінної формалізації знань про структуру корпоративної мережі та не підтримують пояснюване прийняття рішень. Саме тому для задач управління корпоративними комп'ютерними мережами найбільш доцільним є використання онтологічного підходу, який дозволяє інтегрувати знання про ресурси, сервіси, користувачів та множину задач у межах єдиної формальної моделі та забезпечує основу для реалізації онтологічно-керованого конфігурування і багатокритеріальної оптимізації мережевої інфраструктури.

1.5 Постановка задачі дослідження

Сучасні корпоративні комп'ютерні мережі є складними розподіленими інформаційно-комунікаційними середовищами, що забезпечують функціонування великої кількості інформаційних ресурсів, сервісів та прикладних задач. Вони об'єднують користувачів, серверне обладнання, мережеві пристрої, системи

зберігання даних, програмні платформи та засоби інформаційної безпеки в єдину інфраструктуру, яка повинна функціонувати в умовах постійної зміни навантаження, конфігурації ресурсів та вимог користувачів.

Аналіз існуючих систем управління корпоративними мережами показав, що більшість сучасних програмних засобів орієнтована переважно на моніторинг технічних параметрів мережевої інфраструктури, автоматизацію окремих процесів адміністрування та контроль доступності сервісів. Водночас процеси прийняття конфігураційних рішень, оптимізації використання ресурсів та розподілу навантаження значною мірою залишаються залежними від досвіду адміністратора та не враховують у повному обсязі взаємозв'язки між мережевими ресурсами, сервісами, політиками безпеки та множиною задач, що розв'язуються в мережевому середовищі.

Особливо гостро ця проблема проявляється в умовах динамічної зміни структури мережі, появи нових сервісів, збільшення кількості користувачів та необхідності забезпечення високого рівня доступності інформаційних ресурсів. У таких умовах традиційні підходи до управління мережею втрачають ефективність через неможливість комплексного врахування всіх факторів, які впливають на функціонування корпоративної інфраструктури.

Таким чином, актуальним науково-технічним завданням є розроблення методів, моделей та програмних засобів знання-орієнтованого управління корпоративними комп'ютерними мережами на основі онтологічного представлення знань про ресурси мережі, сервіси та множину задач, що розв'язуються в мережевому середовищі.

Проведений аналіз сучасних методів управління корпоративними комп'ютерними мережами показав, що традиційні системи моніторингу, засоби автоматизації конфігурування, алгоритми балансування навантаження та методи машинного навчання здебільшого працюють із окремими параметрами функціонування мережевих ресурсів і не забезпечують формалізованого представлення знань про предметну область.

Нечітка логіка дозволяє враховувати невизначеність інформації, а нейронні мережі та методи машинного навчання забезпечують прогнозування станів мережі та виявлення аномалій. Проте зазначені підходи не дозволяють явно представляти знання про структуру мережі, її сервіси, ресурси та задачі, а також не забезпечують пояснюваність прийнятих рішень. На відміну від зазначених підходів, онтологічне моделювання дозволяє формалізувати знання про предметну область у вигляді концептів, відношень, аксіом та правил логічного виведення. Такий підхід забезпечує інтеграцію інформації про ресурси мережі, сервіси, політики безпеки та множину задач у межах єдиної моделі знань.

Крім того, використання онтологічного підходу створює можливість реалізації автоматизованого конфігурування мережевих ресурсів, перевірки конфігурацій на відповідність встановленим політикам та підтримки багатокритеріальної оптимізації розподілу навантаження між центрами термінальних кластерів.

Метою дисертаційного дослідження є підвищення ефективності управління корпоративними комп'ютерними мережами шляхом розроблення моделей і методів на основі знання-орієнтованого підходу, що враховують структуру мережі, її ресурси та множину задач, які ними розв'язуються.

Для досягнення цієї мети необхідно виконати наступні завдання:

- провести аналіз сучасних технологій, методів і програмних засобів управління корпоративними комп'ютерними мережами, включаючи системи моніторингу, конфігурації та балансування навантаження;
- дослідити можливості застосування знання-орієнтованих, зокрема, онтологічних підходів для підвищення ефективності управління мережевою інфраструктурою;
- розробити онтологічну модель корпоративної комп'ютерної мережі, що забезпечує формалізацію знань про структуру мережі, її ресурси, сервіси та множину задач, які розв'язуються в мережевому середовищі;

- розробити метод онтологічно-керованого конфігурування корпоративної комп'ютерної мережі, який базується на семантичному поданні знань про мережеві ресурси, сервіси та множину задач і забезпечує автоматизоване прийняття конфігураційних рішень на основі механізмів логічного виведення;

- розробити архітектуру програмної онтологічно-керованої системи управління корпоративними комп'ютерними мережами, яка забезпечує інтеграцію онтологічного представлення знань про структуру мережі, її ресурси та множину задач із механізмами логічного виведення, адаптивного прийняття рішень і динамічного розподілу навантаження;

- удосконалити математичну модель розподілу навантаження між центрами термінальних кластерів на основі багатокритеріальної оптимізації та онтологічного подання знань про ресурси корпоративної комп'ютерної мережі й множину задач;

- провести апробацію розробленої, знання-орієнтованої програмної системи на прикладі діючих корпоративних комп'ютерних мереж, які орієнтовані на розв'язання прикладних задач.

Висновки до розділу 1

1. Проведено комплексний аналіз сучасного стану проблеми управління корпоративними комп'ютерними мережами, досліджено особливості їх функціонування, існуючі методи адміністрування, моніторингу, конфігурування та розподілу навантаження, а також розглянуто сучасні знання-орієнтовані підходи до побудови інтелектуальних інформаційних систем.

2. Показано, що сучасні корпоративні комп'ютерні мережі є складними розподіленими середовищами, які об'єднують значну кількість користувачів, сервісів, інформаційних ресурсів та мережевих пристроїв. Зростання масштабів мережевої інфраструктури та кількості задач, що розв'язуються в мережевому

середовищі, призводить до підвищення складності процесів адміністрування та вимагає використання нових підходів до управління ресурсами й сервісами.

3. Виконано аналіз сучасних методів управління корпоративними мережами, включаючи централізовані, децентралізовані та гібридні підходи, методи моніторингу мережевих ресурсів, автоматизованого конфігурування та балансування навантаження. Встановлено, що існуючі методи орієнтовані переважно на аналіз локальних характеристик ресурсів та не забезпечують комплексного врахування взаємозв'язків між ресурсами, сервісами, користувачами та множиною задач.

4. Проведений аналіз сучасних програмних платформ управління корпоративними мережами, зокрема Cisco DNA Center, SolarWinds Network Performance Monitor, ManageEngine OpManager, PRTG Network Monitor та Microsoft System Center, показав високий рівень автоматизації процесів моніторингу та адміністрування мережевої інфраструктури. Разом із тим виявлено, що більшість існуючих рішень не використовуює онтологічне представлення знань про предметну область, не підтримує механізми логічного виведення та не забезпечує пояснюваного формування управлінських рішень.

5. Досліджено основні підходи до формалізації знань, серед яких логічні моделі, продукційні системи, семантичні мережі, фреймові структури та онтології. Встановлено, що онтологічний підхід найбільш повно відповідає вимогам до побудови знання-орієнтованих систем управління складними інформаційними середовищами, оскільки забезпечує формалізацію концептів предметної області, відношень між ними, аксіом та правил логічного виведення в межах єдиної моделі знань.

6. Також проаналізовано можливості використання сучасних інтелектуальних підходів, зокрема нечіткої логіки, машинного навчання та штучних нейронних мереж. Встановлено, що зазначені методи ефективно розв'язують задачі прогнозування, класифікації та роботи з невизначеністю, однак не забезпечують достатнього рівня формалізації знань про структуру

корпоративної мережі та пояснюваності процесу прийняття рішень. Це обмежує можливості їх самостійного використання для автоматизованого конфігурування мережевих ресурсів та управління складними мережевими середовищами.

7. На основі проведеного аналізу сформульовано науково-технічну проблему, яка полягає у відсутності комплексних підходів до управління корпоративними комп'ютерними мережами, здатних інтегрувати формалізовані знання про ресурси, сервіси, політики безпеки та множину задач із механізмами логічного виведення та багатокритеріальної оптимізації. Обґрунтовано доцільність використання онтологічного підходу як основи для побудови знання-орієнтованої системи управління корпоративними мережами.

8. У результаті сформульовано мету та задачі дисертаційного дослідження, визначено загальну схему розв'язання задачі управління корпоративною мережею на основі знань та окреслено основні напрями подальших досліджень. Це створює теоретичне підґрунтя для розроблення онтологічної моделі корпоративної комп'ютерної мережі, методу онтологічно-керованого конфігурування та математичної моделі багатокритеріального розподілу навантаження, які розглядаються у наступних розділах дисертаційної роботи.

РОЗДІЛ 2

ОНТОЛОГІЧНЕ МОДЕЛЮВАННЯ ПРОЦЕСІВ У КОРПОРАТИВНИХ КОМП'ЮТЕРНИХ МЕРЕЖАХ

Другий розділ дисертаційної роботи присвячений розробці математичного забезпечення знання-орієнтованого управління корпоративними комп'ютерними мережами. Результати аналізу, наведені в першому розділі, показали, що існуючі методи та програмні засоби управління мережевою інфраструктурою переважно орієнтовані на моніторинг окремих параметрів функціонування мережі та виконання локальних операцій адміністрування. Водночас вони недостатньо враховують семантичні взаємозв'язки між мережевими ресурсами, сервісами, користувачами та множиною задач, що розв'язуються у корпоративному середовищі, що обмежує можливості автоматизованого прийняття управлінських рішень.

Одним із перспективних напрямів підвищення ефективності управління корпоративними мережами є використання знання-орієнтованого підходу, який передбачає формалізацію знань про предметну область, застосування механізмів логічного виведення та інтеграцію моделей прийняття рішень із семантичним представленням інформації. Реалізація такого підходу потребує розроблення відповідних математичних моделей, методів та формалізмів, які забезпечують опис структури корпоративної мережі, її ресурсів, сервісів і множини задач, а також підтримують процеси автоматизованого конфігурування та розподілу навантаження.

У даному розділі розроблено онтологічну модель корпоративної комп'ютерної мережі, яка забезпечує формалізацію знань про мережеві ресурси, сервіси, користувачів та множину задач у межах єдиного семантичного простору. Запропоновано метод онтологічно-керованого конфігурування корпоративної мережі, що базується на використанні механізмів логічного виведення та дозволяє автоматизувати процес формування конфігураційних рішень. Крім того,

удосконалено математичну модель розподілу навантаження між центрами термінальних кластерів на основі багатокритеріальної оптимізації та онтологічного подання знань про ресурси корпоративної комп'ютерної мережі й множину задач. Основні результати цього розділу опубліковані в працях [27, 72, 88, 94, 116, 117].

2.1 Теоретичне підґрунтя побудови онтології корпоративної мережі

Сучасні корпоративні комп'ютерні мережі являють собою складні багаторівневі інформаційно-технічні системи, функціонування яких визначається великою кількістю взаємопов'язаних компонентів різної природи. До їх складу входять обчислювальні вузли, мережеві пристрої, програмні сервіси, користувачі та механізми управління доступом, що взаємодіють у межах динамічного середовища з постійно змінними характеристиками.

Знання про такі системи формуються у різних інформаційних контурах і, як правило, мають фрагментований характер. Частина з них зберігається у конфігураційних описах, частина - у системах моніторингу, інша - у вигляді процедурних алгоритмів управління ресурсами [39, 40, 42, 44, 46, 47]. У результаті відсутня єдина узгоджена модель, яка б дозволяла інтегрувати ці знання, виконувати їх формалізований аналіз та використовувати для прийняття рішень.

Традиційні підходи до управління корпоративними мережами, що базуються на використанні локальних метрик та евристичних алгоритмів, не забезпечують належного рівня узагальнення та адаптивності. Вони орієнтовані на обробку поточних даних і, як правило, не враховують семантичні взаємозв'язки між об'єктами мережі. Це суттєво обмежує можливості побудови інтелектуальних систем управління, здатних працювати в умовах невизначеності та багатокритеріальності.

У зв'язку з цим виникає необхідність у формалізованому підході до представлення знань про корпоративну мережу, який би забезпечував їх цілісність, узгодженість та можливість подальшої алгоритмічної інтерпретації.

Одним із найбільш ефективних засобів вирішення зазначеної проблеми є онтологічний підхід, який дозволяє представити предметну область у вигляді формальної системи знань із чітко визначеними поняттями, відношеннями та обмеженнями [72, 89, 93, 96, 97, 98, 116]. У межах цього підходу корпоративна мережа розглядається не лише як сукупність технічних елементів, а як семантично впорядкована система, у якій кожен об'єкт має визначену роль і взаємодіє з іншими відповідно до заданих правил.

На рисунку 2.1. показано взаємозв'язок між особливостями корпоративної мережі, фрагментованістю джерел знань, обмеженнями традиційних підходів та необхідністю застосування онтологічного підходу як основи формалізованого представлення, логічного виведення та подальшої алгоритмічної інтерпретації знань.

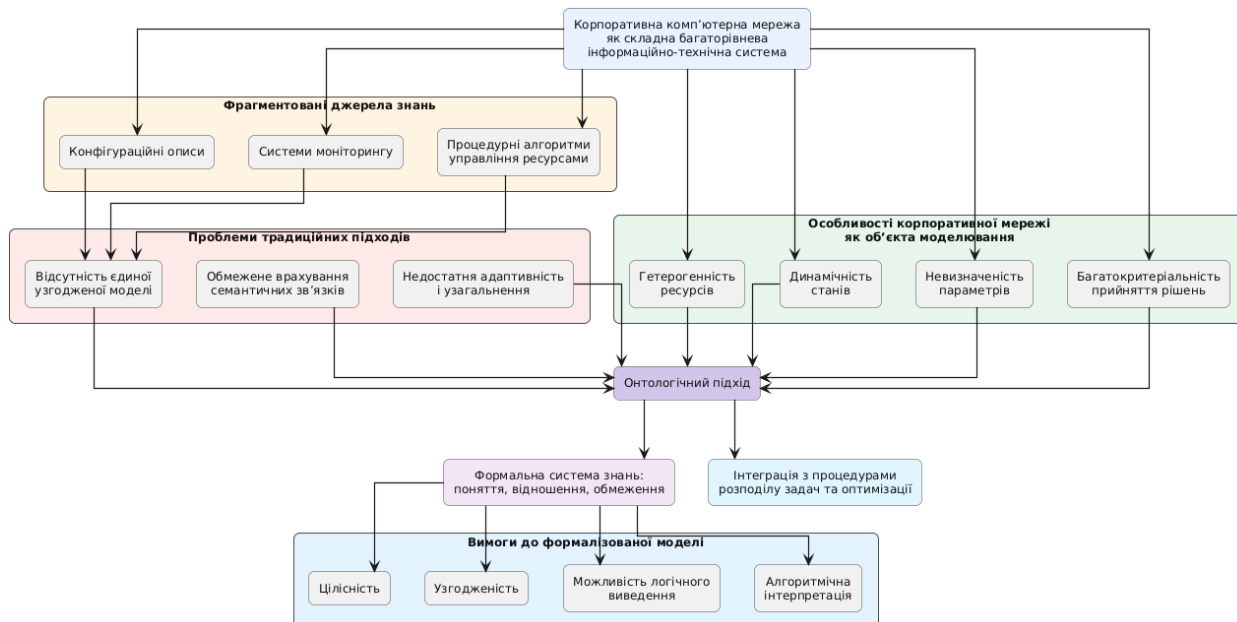


Рис. 2.1. Концептуальна схема теоретичного підґрунтя побудови онтології корпоративної мережі

Важливо підкреслити, що онтологія у даному контексті не обмежується лише описовою функцією. Вона виступає як формальна основа, що дозволяє виконувати логічне виведення, виявляти неявні залежності та формувати узгоджені множини

рішень. Таким чином, онтологічна модель стає інструментом не тільки представлення, але й обробки знань.

Разом з тим, існуючі онтологічні підходи здебільшого орієнтовані на статичне моделювання предметної області і не враховують специфіку процесів управління обчислювальними ресурсами. Це проявляється у відсутності механізмів інтеграції онтологічного опису з алгоритмічними процедурами розподілу задач та оптимізації.

Особливості корпоративних мереж як об'єкта моделювання визначають вимоги до побудови їх онтології. Передусім, мова йде про гетерогенність ресурсів, що зумовлює необхідність уніфікованого представлення різнорідних характеристик у межах єдиної моделі [86, 87, 102, 103, 104]. Водночас динамічний характер функціонування мережі означає, що будь-яка модель повинна враховувати змінність параметрів у часі. Додаткову складність створює невизначеність значень ключових характеристик, таких як навантаження вузлів або час виконання задач. У більшості випадків ці параметри не можуть бути задані у вигляді точкових оцінок і доцільно представляти їх у вигляді інтервалів. Це, у свою чергу, вимагає розширення класичних підходів до формалізації знань. Окрім цього, процеси управління корпоративною мережею мають багатокритеріальний характер, оскільки рішення приймаються з урахуванням одночасно декількох факторів, що часто суперечать один одному. Це обумовлює необхідність інтеграції онтологічного підходу з методами оптимізації.

Онтологія корпоративної мережі формується як багаторівнева система класів, що відображає різні аспекти функціонування ІТ-інфраструктури. Її побудова базується на ідеї переходу від фізичного рівня організації мережі до логічного, функціонального, організаційного та, зрештою, до рівня управління ресурсами й безпеки. Така послідовність дозволяє не лише описати структуру мережі, але й відобразити процеси її функціонування у цілісному вигляді. Основу онтології становлять класи, що описують фізичну інфраструктуру мережі, рисунок 2.2.

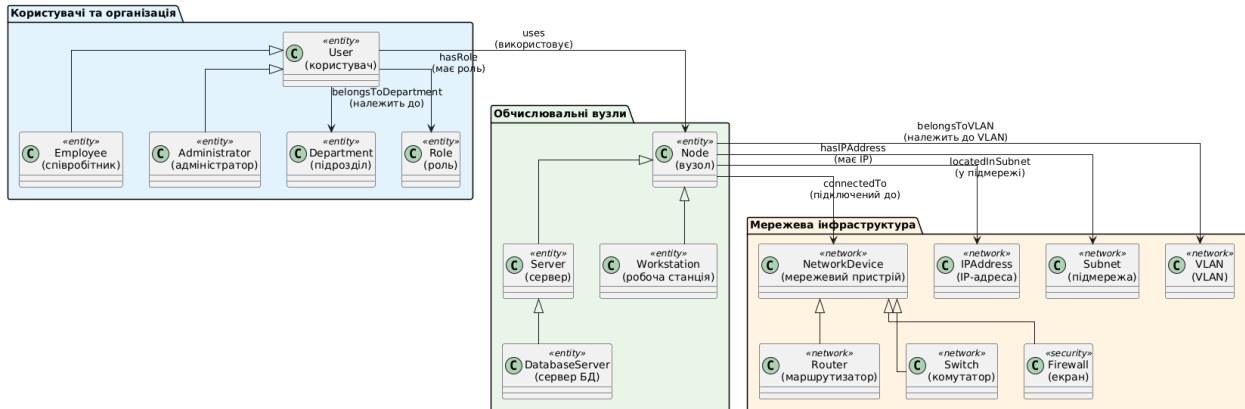


Рис. 2.2. Структура користувачів, вузлів і мережевої інфраструктури

Центральним у цьому контексті є клас NetworkDevice, який охоплює всі пристрої, що забезпечують передачу та обробку мережевого трафіку. У межах цього класу виділяються спеціалізовані типи пристроїв, які виконують різні функції в мережі. Так, клас Router відповідає за маршрутизацію даних між мережами, клас Switch забезпечує комутацію в межах локальної мережі, а клас Firewall виконує функцію контролю доступу та фільтрації трафіку. Розширення цього рівня за рахунок класів AccessPoint та LoadBalancer дозволяє врахувати бездротову взаємодію та механізми балансування навантаження, що є характерними для сучасних корпоративних мереж.

Взаємодіючи з мережевими пристроями, функціонують обчислювальні вузли, які формують наступний рівень онтології. Узагальнюючим класом тут виступає Node, який описує будь-який об'єкт, здатний виконувати обчислення або обробляти дані. У його межах клас Server представляє вузли, що надають ресурси або сервіси іншим компонентам, тоді як клас Workstation описує робочі станції користувачів. Подальша деталізація через класи VirtualMachine та Container дозволяє врахувати віртуалізоване середовище, а включення класів MobileDevice, Printer і StorageNode розширює модель до реальних сценаріїв використання мережі, де присутні як мобільні клієнти, так і спеціалізовані пристрої зберігання або периферії.

Опис фізичних об'єктів є недостатнім без врахування логічної організації

мережі, що формується через класи мережевих сутностей. Клас IPAddress задає логічну адресу вузла, що визначає його місце у мережевому просторі, тоді як клас MACAddress відповідає за фізичну ідентифікацію пристрою. Класи Subnet і VLAN дозволяють відобразити сегментацію мережі, яка використовується для підвищення ефективності та безпеки. Додатково клас Protocol описує правила передачі даних, а класи Port і Session забезпечують можливість формалізації конкретних взаємодій між вузлами. Таким чином, логічний рівень поєднує фізичну інфраструктуру з процесами комунікації.

На цьому рівні стає можливим перейти до опису функціонального аспекту мережі, який представлений класами сервісів і програмних компонентів, рисунок 2.3. Клас Service узагальнює всі функціональні можливості, що надаються мережею, а його спеціалізації, такі як WebService, DatabaseService, MailService чи AuthenticationService, дозволяють відобразити конкретні види сервісів. У цьому контексті сервер перестає бути лише технічним вузлом і набуває ролі середовища виконання сервісів. Клас Application доповнює цю модель, представляючи прикладне програмне забезпечення, тоді як клас OperatingSystem задає базовий рівень функціонування обчислювальних вузлів. У результаті формується повна картина програмного середовища корпоративної мережі.

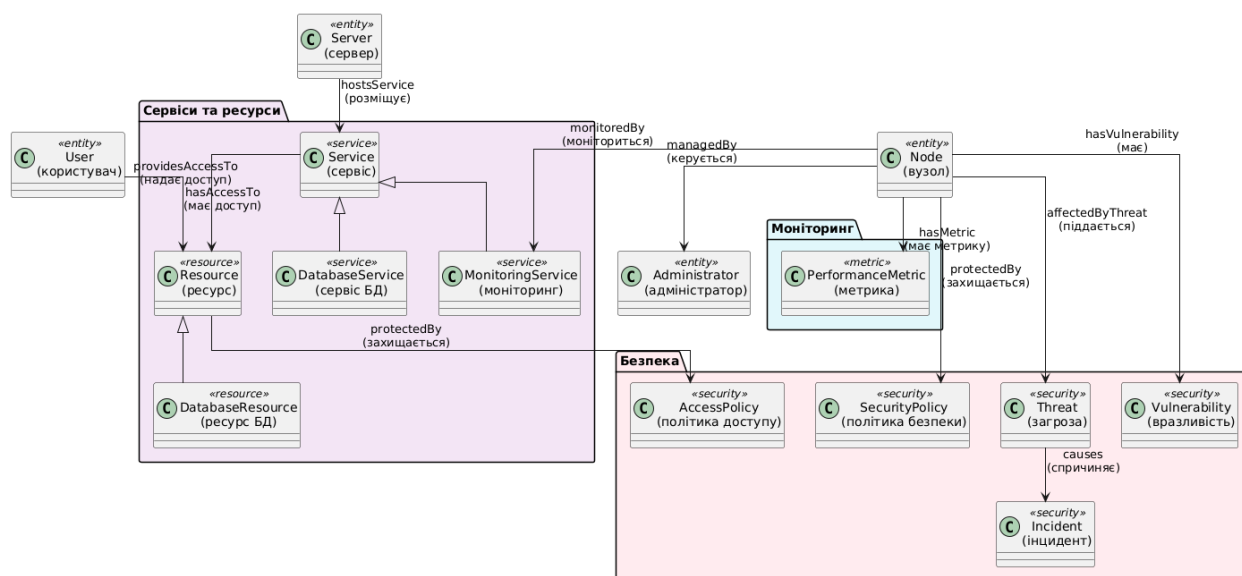


Рис. 2.3. Структура сервісів, ресурсів, моніторингу і безпеки

Функціонування сервісів нерозривно пов'язане з використанням ресурсів, що відображається через клас `Resource` та його спеціалізації. Клас `ComputeResource` характеризує обчислювальні можливості, клас `MemoryResource` - оперативну пам'ять, `StorageResource` - засоби зберігання даних, а `NetworkResource` - пропускну здатність мережі. У цьому контексті ресурси виступають як обмежений фактор, що визначає можливості розміщення сервісів і виконання задач. Саме на цьому рівні онтологія набуває безпосереднього зв'язку з задачами оптимізації, оскільки дозволяє формалізувати співвідношення між потребами системи та доступними ресурсами.

Паралельно з технічною та функціональною структурою мережі формується організаційний рівень, який відображає взаємодію користувачів із системою. Клас `User` описує суб'єктів доступу, а його спеціалізації, зокрема `Administrator`, `Employee`, `GuestUser` та `ServiceAccount`, дозволяють врахувати різні типи користувачів. Клас `Department` вводить організаційний контекст, у якому функціонують користувачі, тоді як класи `Role` і `Group` дозволяють формалізувати механізми розмежування доступу. Завдяки цьому онтологія поєднує технічну інфраструктуру з організаційною структурою підприємства.

З урахуванням взаємодії користувачів із ресурсами та сервісами природним є перехід до рівня безпеки. Клас `AccessPolicy` визначає правила доступу до ресурсів, тоді як клас `SecurityPolicy` описує загальні принципи захисту системи. Класи `Threat`, `Vulnerability` та `Incident` дозволяють формалізувати загрози, слабкі місця та факти порушення безпеки. Додатково класи `AttackVector`, `Risk` і `SecurityControl` забезпечують можливість аналізу сценаріїв атак і засобів їх протидії. Таким чином, безпековий рівень інтегрується з усіма попередніми рівнями онтології.

Завершальним елементом структури є класи, що описують стан системи та її поведінку у часі. Клас `PerformanceMetric` узагальнює показники ефективності, тоді як класи `CPUUsage`, `MemoryUsage`, `BandwidthUsage` та `Latency` деталізують різні аспекти навантаження. Клас `AvailabilityStatus` відображає доступність вузлів і

сервісів, а класи Event і Alert дозволяють зафіксувати події та реакцію системи на них. У цьому контексті онтологія виходить за межі статичного опису і стає інструментом моделювання динамічних процесів.

У результаті така структура класів формує цілісну онтологічну модель корпоративної мережі, в якій фізичні компоненти, логічні зв'язки, сервіси, ресурси, користувачі та механізми безпеки поєднані в єдину систему. Саме ця інтеграція забезпечує можливість подальшого переходу до формалізації відношень, побудови онтологічних аксіом і використання моделі для задач аналізу та оптимізації функціонування корпоративної мережі.

Такі класи набувають повного змісту лише тоді, коли між ними встановлюється система семантичних відношень, що відображає реальні зв'язки між компонентами корпоративної мережі. Саме відношення перетворюють множину окремих об'єктів на цілісну онтологічну модель, придатну для логічного виведення, аналізу залежностей і подальшої формалізації засобами алгебри кортежів.

У межах онтології корпоративної мережі відношення виконують декілька взаємопов'язаних функцій. Передусім вони фіксують структурні зв'язки між фізичними та логічними елементами інфраструктури. Крім цього, вони задають функціональні залежності між користувачами, сервісами та ресурсами, а також відображають правила доступу, адміністрування й захисту. Завдяки цьому онтологія може бути використана не лише для опису конфігурації мережі, але й для інтерпретації процесів її функціонування.

Одним із базових відношень є `connectedTo`, яке описує факт підключення одного вузла або мережевого пристрою до іншого. Це відношення відображає фізичну або логічну топологію мережі та дозволяє представити шляхи передачі трафіку між її елементами. Якщо, наприклад, робоча станція з'єднана з комутатором, а комутатор - з маршрутизатором, то на основі послідовності таких зв'язків можна аналізувати маршрути доступу до мережевих ресурсів або виявляти критичні вузли інфраструктури.

Тісно пов'язаним із ним є відношення `locatedInSubnet`, яке фіксує належність вузла до певної підмережі. На відміну від `connectedTo`, що описує топологічну взаємодію, це відношення відображає логічну сегментацію мережі. Воно є важливим для моделювання правил маршрутизації, розмежування доступу та аналізу мережевих політик. Подібну роль виконує й відношення `belongsToVLAN`, яке встановлює належність пристрою або вузла до певної віртуальної локальної мережі. Завдяки цьому в онтології можна формально представити не тільки фізичну структуру мережі, але й її віртуальну організацію.

Для опису адресації доцільно використовувати відношення `hasIPAddress` та `hasMACAddress`. Перше з них пов'язує вузол або мережевий пристрій з його IP-адресою і тим самим відображає його місце в логічному адресному просторі. Друге фіксує фізичну адресу мережевого інтерфейсу. У поєднанні ці відношення створюють основу для опису ідентифікаційних характеристик об'єктів мережі, що є важливим як для мережевого адміністрування, так і для аналізу подій безпеки. У випадку необхідності ця група відношень може бути доповнена відношенням `hasDNSName`, яке пов'язує вузол або сервіс із символьним іменем у системі доменних імен.

Особливе значення для онтології мають відношення, що описують взаємодію користувачів з інфраструктурою. Відношення `uses` відображає факт використання користувачем певного пристрою або вузла. Воно дозволяє встановити, через які технічні засоби користувач здійснює доступ до корпоративної мережі. Це створює основу для подальшого логічного виведення. Якщо, наприклад, користувач працює за певною робочою станцією, а ця станція має доступ до конкретного сервісу, то можна аналізувати ланцюг взаємодії між користувачем і цим сервісом.

Відношення `belongsToDepartment` описує належність користувача до певного організаційного підрозділу. Його роль виходить за межі простого кадрового опису, оскільки саме через організаційну структуру часто визначаються права доступу, допустимі сервіси та рівень відповідальності користувача. Схожим за змістом є відношення `hasRole`, яке фіксує функціональну роль користувача в системі. Якщо

`belongsToDepartment` відображає організаційне місце користувача, то `hasRole` характеризує його функціональні повноваження. Саме на підставі цього відношення можуть будуватися моделі рольового контролю доступу.

Для формалізації доступу до інформаційних об'єктів центральним є відношення `hasAccessTo`, яке пов'язує користувача або інший суб'єкт доступу з певним ресурсом. Воно є одним із ключових у моделі, оскільки дозволяє безпосередньо відобразити факт надання права доступу. Проте в більш складних випадках це відношення має інтерпретуватися не ізольовано, а в контексті політик доступу, ролей і сервісів, через які реалізується взаємодія з ресурсом. Саме тому важливим є також відношення `protectedBy`, яке вказує, що певний ресурс, вузол або сервіс захищений політикою безпеки або іншим механізмом захисту. Така зв'язка дозволяє аналізувати не просто факт доступу, а його допустимість і рівень контрольованості.

Не менш важливою є група відношень, що описує функціонування сервісів. Відношення `hostsService` встановлює, що певний сервер є середовищем розміщення конкретного сервісу. Завдяки цьому онтологія відображає функціональну залежність між обчислювальними вузлами і програмними компонентами. Якщо сервер розміщує сервіс, а сервіс, своєю чергою, забезпечує доступ до певного ресурсу, то можна перейти від інфраструктурного рівня до прикладного. У цьому контексті корисним є також відношення `usesService`, яке показує, що користувач або прикладна система взаємодіє з певним сервісом. Разом із `hostsService` воно дозволяє відновити повний ланцюг: користувач - сервіс - сервер - ресурс.

Для точнішого опису сервісних залежностей доцільно використовувати відношення `dependsOnService`, яке показує, що один сервіс функціонально залежить від іншого. Таке відношення особливо важливе для корпоративних систем, де, наприклад, вебсервіс може залежати від сервісу автентифікації та сервісу баз даних. У результаті онтологія дозволяє досліджувати каскадні наслідки відмов, перевантажень або інцидентів безпеки.

У задачах управління ресурсами суттєву роль відіграють відношення

`requiresResource` та `allocatedTo`. Перше з них описує потребу задачі, сервісу або прикладного компонента у певному ресурсі. Це відношення відображає попит на обчислювальні, пам'яттєві, мережеві або сховищні потужності. Друге відношення встановлює факт виділення конкретного ресурсу певному об'єкту. Сукупне використання цих відношень дозволяє формалізувати процес розподілу ресурсів, що є важливим для задач оптимізації та побудови алгоритмів керування навантаженням.

Узгодження між потребами та реальними можливостями інфраструктури відображається через відношення `hasCPUCapacity`, `hasMemoryCapacity`, `hasStorageCapacity` та `hasBandwidthCapacity`. Вони зв'язують вузол або інший елемент інфраструктури з відповідними видами ресурсної спроможності. У поєднанні з відношенням `requiresResource` ці конструкції створюють формальну основу для перевірки допустимості виконання задачі або розміщення сервісу на певному вузлі.

Для відображення адміністративного рівня мережі використовується відношення `managedBy`, яке пов'язує пристрій, вузол або сервіс з адміністратором чи відповідальною роллю. Воно дозволяє формалізувати розподіл зон відповідальності та використовується при аналізі експлуатаційних процесів. Подібним чином відношення `monitoredBy` встановлює, що певний об'єкт контролюється сервісом або засобом моніторингу. Це важливо для опису поточного спостереження за станом мережі, фіксації показників продуктивності та своєчасного виявлення відхилень.

Безпековий контур онтології формується завдяки відношенням `hasVulnerability`, `affectedByThreat`, `detectedBy`, `mitigatedBy` та `causesIncident`. Відношення `hasVulnerability` фіксує наявність уразливості в певному вузлі, сервісі або програмному компоненті. Воно є важливим, оскільки дозволяє встановити потенційні точки ризику. Відношення `affectedByThreat` указує, що відповідний об'єкт підпадає під дію певної загрози. Якщо ж загроза реалізується, це може бути відображено через відношення `causesIncident`, яке пов'язує загрозу або атаку з

інцидентом безпеки. Водночас `detectedBy` фіксує засіб або механізм, завдяки якому була виявлена підозріла активність, а `mitigatedBy` указує на контрзахід або політику, що дозволяє зменшити або усунути ризик.

Для аналізу подій і часової динаміки доцільно застосовувати відношення `occursAtTime`, `causedBy`, `resultsIn` та `resolvedBy`. Вони дозволяють розмістити факт, подію чи інцидент у часовому контексті та простежити причинно-наслідкові зв'язки. Це особливо важливо, якщо онтологія використовується не лише як статична модель інфраструктури, але і як основа для аналізу журналів подій, моніторингових даних та інцидентів безпеки.

У цілому система відношень в онтології корпоративної мережі забезпечує формальне поєднання декількох рівнів опису. На одному рівні вона відображає фізичну та логічну структуру мережі, на іншому - моделює організаційні зв'язки та правила доступу, а на ще одному - дозволяє описати функціонування сервісів, використання ресурсів і механізми безпеки. Саме завдяки такій багаторівневій системі відношень онтологія стає придатною для задач логічного виведення, перевірки узгодженості, аналізу ризиків і побудови інтелектуальних механізмів управління корпоративною мережею.

2.2 Формалізація онтології корпоративної мережі

Перехід від концептуального опису корпоративної мережі до її формалізованого представлення є необхідною умовою побудови системи, здатної до автоматизованого аналізу, логічного виведення та підтримки прийняття рішень. Такий перехід передбачає визначення структури знань у вигляді формальної моделі, яка відображає не лише складові елементи мережі, але й закономірності їх взаємодії, а також обмеження, що визначають допустимі режими функціонування [72, 116].

У межах запропонованого підходу онтологія корпоративної мережі розглядається як формальна система, що поєднує опис предметної області з

механізмами її інтерпретації. Це дозволяє перейти від декларативного опису знань до їх використання у процедурних алгоритмах, зокрема в задачах оптимізації розподілу обчислювальних ресурсів.

Формально онтологію корпоративної мережі доцільно представити у вигляді кортежу:

$$O_{net} = \langle D, M, U, P, S, E, A \rangle, \quad (2.1)$$

де кожна компонента відображає окремий аспект предметної області та виконує визначену функцію у процесі формалізації знань.

Першою складовою є множина D , яка задає предметну область. Вона визначає контекст, у межах якого здійснюється побудова онтології, та включає опис топології мережі, її функціональних елементів і зовнішніх умов функціонування. Формально предметна область може бути представлена у вигляді:

$$D = \langle id_D, dom \rangle, \quad (2.2)$$

де id_D – однозначно ідентифікує область, а dom містить її змістовий опис. Важливо підкреслити, що саме через цю компоненту забезпечується зв'язок формальної моделі з реальною системою.

Наступною складовою є множина M , яка описує структуру мережевої інфраструктури. На відміну від предметної області, що задає загальний контекст, ця множина конкретизує елементи системи та їх взаємозв'язки. Формально вона визначається наступним чином:

$$M = \langle id_M, Str_m \rangle, \quad (2.3)$$

де id_M є ідентифікатором елемента мережі, а Str_m - його формалізованим описом. У цьому описі фіксуються типи пристроїв, їх конфігураційні

характеристики та зв'язки з іншими елементами. Таким чином, множина M виконує роль структурного каркасу онтології.

Особливе значення має множина U , яка визначає об'єкти використання мережі. Якщо M описує ресурси, то U відображає процеси їх використання. До цієї множини належать задачі, сервіси та користувачі, що взаємодіють з ресурсами мережі. Формалізація цієї множини має вигляд:

$$U = \langle id_u, Str_u \rangle, \quad (2.4)$$

де id_u ідентифікує об'єкт, а Str_u визначає його характеристики, зокрема вимоги до ресурсів та умови виконання. Таким чином, саме через множину U здійснюється зв'язок між структурою мережі та процесами її функціонування.

Множина параметрів P забезпечує кількісне представлення характеристик системи. Вона дозволяє перейти від якісного опису до формалізованих числових оцінок, необхідних для прийняття рішень. Формально вона визначається як:

$$P = \langle id_p, type_p, val_p \rangle, \quad (2.5)$$

де $type_p$ задає тип параметра, а val_p - його значення. У контексті корпоративної мережі ці параметри описують обчислювальні ресурси, такі як процесорний час, обсяг пам'яті або пропускна здатність каналів зв'язку. Важливою особливістю є можливість інтервального представлення значень параметрів, що дозволяє врахувати невизначеність.

Множина S відображає стани мережі та забезпечує динамічний аспект моделі. Якщо попередні компоненти описують структуру та характеристики системи, то S дозволяє врахувати її змінність у часі. Формально вона задається як:

$$S = \langle id_s, cond_s \rangle, \quad (2.6)$$

де $cond_s$ описує поточний стан мережі. Ця компонента є ключовою для адаптивного управління, оскільки дозволяє враховувати зміну навантаження та доступності ресурсів.

Множина E визначає характеристики виконання задач і забезпечує зв'язок між ресурсами та результатами їх використання. Вона описує показники ефективності, які використовуються для оцінювання рішень.

$$E = \langle id_e, main_e, alt_e, rule_e \rangle, \quad (2.7)$$

де $main_e$ та alt_e відображають основні та альтернативні характеристики, а $rule_e$ задає правила їх інтерпретації.

Нарешті, множина A включає методи та правила, що визначають процеси управління ресурсами. Саме ця компонента забезпечує перехід від статичного опису до динамічного використання знань. Формально вона може бути представлена як:

$$A = \langle id_a, meth_a, oper_a \rangle, \quad (2.8)$$

де $meth_a$ визначає метод, а $oper_a$ - набір операцій, що його реалізують.

Узгоджена взаємодія всіх зазначених компонент формує цілісну онтологічну модель, у якій структурні, функціональні та процедурні аспекти об'єднані в єдину систему. При цьому важливою особливістю є те, що кожна компонента може бути використана незалежно, але максимальна ефективність досягається лише при їх інтегрованому застосуванні.

Таким чином, запропонована формалізація дозволяє представити корпоративну мережу як багаторівневу систему знань, у якій поєднуються декларативні та процедурні елементи, рисунок 2.4.

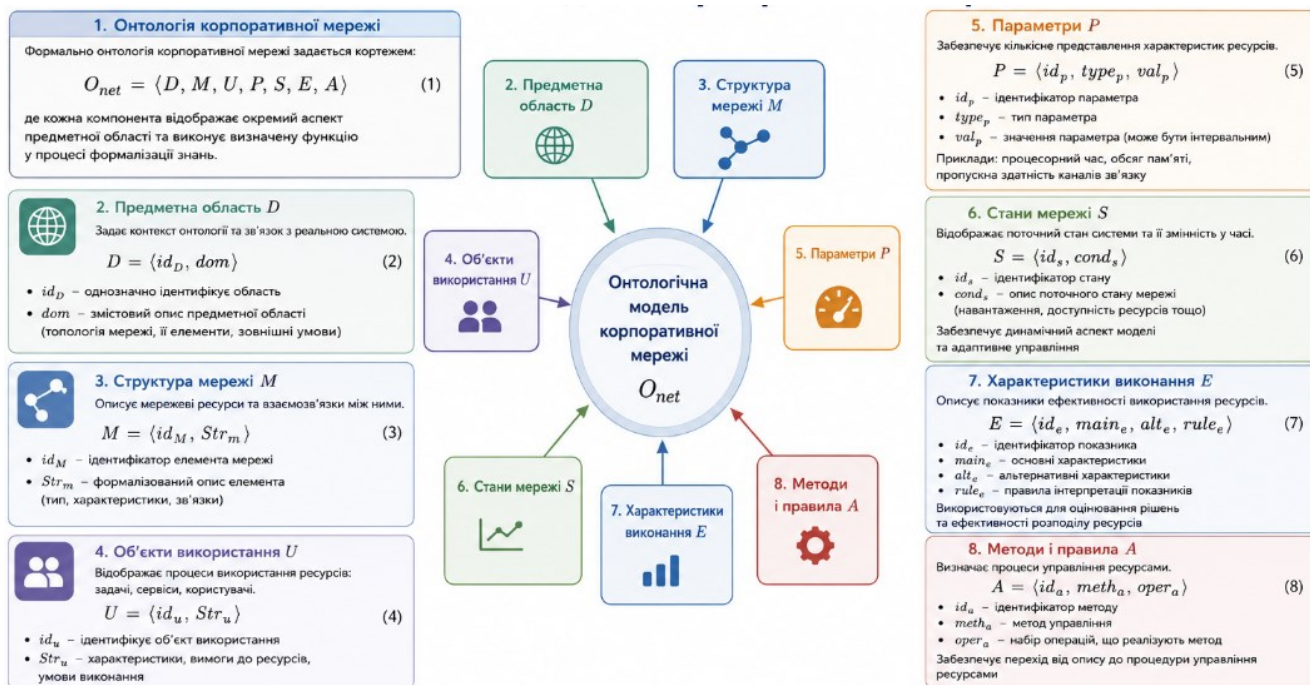


Рис. 2.4. Онтологічна модель корпоративної комп'ютерної мережі

Для забезпечення формалізованого опису процесів використання обчислювальних ресурсів корпоративної мережі вводиться множина онтологічних аксіом A_k :

$$A_k = \langle a_1, a_2, \dots, a_k \rangle, \tag{2.9}$$

які визначають обмеження, правила та залежності між елементами онтології O_{net} . На рисунку 2.5. представлено формалізований опис запропонованої системи онтологічних аксіом в межах побудови онтології корпоративної мережі.

Введення системи аксіом є необхідним для переходу від декларативного опису структури корпоративної мережі до її формально-логічної інтерпретації, яка дозволяє здійснювати автоматизоване виведення знань, контроль узгодженості та підтримку процесів прийняття рішень щодо розподілу обчислювальних ресурсів.

Онтологічні аксіоми формують допустиму область рішень, у межах якої здійснюється оптимізація, та визначають умови коректності функціонування корпоративної мережі. Розглянемо ці аксіоми детальніше.

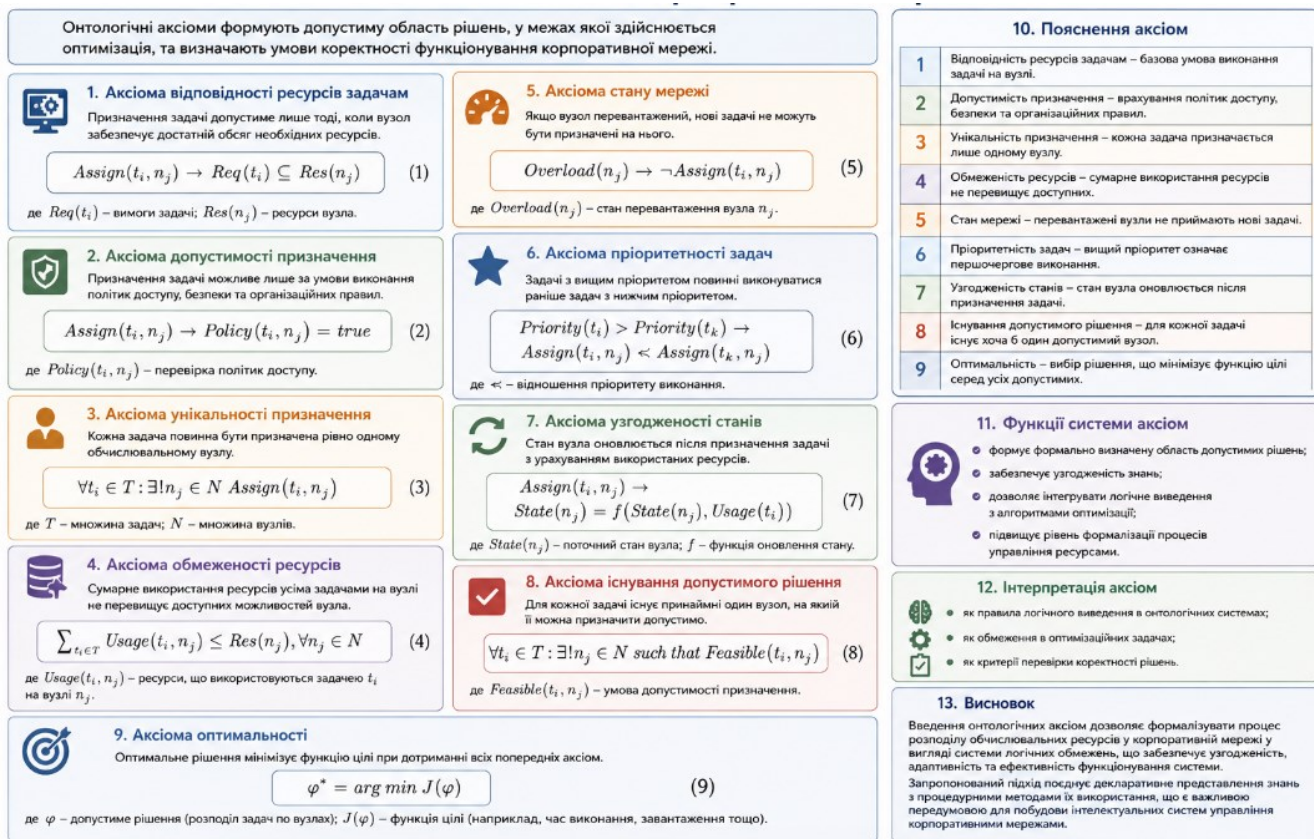


Рис. 2.5. Система онтологічних аксіом корпоративної мережі

1. Аксіома відповідності ресурсів задачам. Для кожної задачі $t_i \in T$ та вузла $n_j \in N$ виконується:

$$\text{Assign}(t_i, n_j) \rightarrow \text{Req}(t_i) \subseteq \text{Res}(n_j), \quad (2.10)$$

Дана аксіома визначає базову умову можливості виконання задачі на конкретному обчислювальному вузлі. Вона відображає необхідність відповідності між ресурсними вимогами задачі $\text{Req}(t_i)$ та доступними ресурсами вузла $\text{Res}(n_j)$. Іншими словами, призначення задачі є допустимим лише у випадку, коли вузол забезпечує достатній обсяг обчислювальних, пам'яттєвих та мережевих ресурсів. Таким чином, ця аксіома виступає первинним фільтром при формуванні множини допустимих рішень.

2. Аксіома допустимості призначення. Дана аксіома формалізує обмеження, пов'язані з політиками доступу, безпеки та організаційними правилами

функціонування корпоративної мережі

$$Assign(t_i, n_j) \rightarrow Policy(t_i, n_j) = true, \quad (2.11)$$

Дана аксіома формалізує обмеження, пов'язані з політиками доступу $Policy(t_i, n_j)$, безпеки та організаційними правилами функціонування корпоративної мережі. Вона забезпечує врахування не лише технічних характеристик ресурсів, але й контекстних умов використання, таких як:

- обмеження доступу користувачів;
- сегментація мережі;
- вимоги до ізоляції задач.

Таким чином, навіть при наявності достатніх ресурсів, призначення задачі може бути заборонене з точки зору політик.

3. Аксіома унікальності призначення. Дана аксіома формалізує обмеження, пов'язані з політиками доступу, безпеки та організаційними правилами функціонування корпоративної мережі

$$\forall t_i \in T: \exists! n_j \in N Assign(t_i, n_j), \quad (2.12)$$

Ця аксіома визначає, що кожна задача у системі повинна бути призначена лише одному обчислювальному вузлу. Вона забезпечує:

- однозначність розподілу;
- уникнення дублювання обчислень;
- коректність обліку використання ресурсів.

У випадках, коли допускається паралельне виконання задач, дана аксіома може бути модифікована шляхом введення множинного відображення, однак у базовій постановці вона забезпечує строгість моделі.

4. Аксіома обмеженості ресурсів. Дана аксіома описує фундаментальне обмеження будь-якої обчислювальної системи - обмеженість ресурсів. Вона

гарантує, що сумарне використання ресурсів усіма задачами, призначеними на вузол, не перевищує доступних можливостей цього вузла.

$$\sum_{t_i \in T} Usage(t_i, n_j) \leq Res(n_j), \forall n_j \in N. \quad (2.13)$$

Ця аксіома є ключовою при формуванні допустимої області рішень та безпосередньо впливає на процес оптимізації.

5. Аксіома стану мережі. Дана аксіома враховує динамічний характер функціонування корпоративної мережі. Вона відображає залежність можливості призначення задач від поточного стану вузла. Якщо вузол знаходиться у стані перевантаження, нові задачі не можуть бути призначені, що дозволяє уникнути деградації продуктивності системи.

$$Overload(n_j) \rightarrow \neg Assign(t_i, n_j). \quad (2.14)$$

Таким чином, аксіома забезпечує адаптивність алгоритму до змін у середовищі.

6. Аксіома пріоритетності задач. Ця аксіома визначає порядок обробки задач залежно від їх пріоритетів. Вона дозволяє враховувати критичність задач та забезпечує першочергове виконання задач з вищим рівнем важливості.

$$Priority(t_i) > Priority(t_k) \rightarrow Assign(t_i, n_j) < Assign(t_k, n_j). \quad (2.15)$$

Таким чином, у процесі розподілу ресурсів враховується не лише технічна доцільність, але й організаційна значущість задач.

7. Аксіома узгодженості станів. Дана аксіома описує зміну стану вузла в результаті призначення задачі. Вона формалізує динамічну поведінку системи, забезпечуючи узгодженість між: поточним станом ресурсів та новими призначеннями задач.

$$\text{Assign}(t_i, n_j) \rightarrow \text{State}(n_j) = f \left(\text{State}(n_j), \text{Usage}(t_i) \right) \quad (2.16)$$

Функція f відображає механізм оновлення стану вузла з урахуванням використаних ресурсів.

8. Аксиома існування допустимого рішення. Ця аксіома гарантує існування хоча б одного допустимого варіанту розподілу для кожної задачі. У випадку порушення цієї умови система повинна змінити параметри задачі, переглянути політики або відкласти виконання задачі.

$$\forall t_i \in T: \exists! n_j \in N \text{ such that } \text{Feasible}(t_i, n_j), \quad (2.17)$$

Таким чином, аксіома забезпечує повноту моделі.

8. Аксиома оптимальності. Дана аксіома формалізує задачу оптимізації розподілу ресурсів. Вона визначає, що оптимальне рішення повинно мінімізувати функцію цілі при дотриманні всіх попередніх аксіом.

$$\phi^* = \arg \min J(\phi), \quad (2.18)$$

Таким чином, аксіоми виступають як система обмежень, у межах якої здійснюється пошук оптимального рішення, рисунок . Запропонована система онтологічних аксіом виконує декілька важливих функцій:

- формує формально визначену область допустимих рішень;
- забезпечує узгодженість знань;
- дозволяє інтегрувати логічне виведення з алгоритмами оптимізації;
- підвищує рівень формалізації процесів управління ресурсами.

Особливістю є те, що аксіоми можуть бути інтерпретовані: як правила логічного виведення в онтологічних системах; як обмеження в оптимізаційних задачах; як критерії перевірки коректності рішень.

Введення онтологічних аксіом дозволяє формалізувати процес розподілу

обчислювальних ресурсів у корпоративній мережі у вигляді системи логічних обмежень, що забезпечує узгодженість, адаптивність та ефективність функціонування системи. Запропонований підхід дозволяє поєднати декларативне представлення знань з процедурними методами їх використання, що є важливою передумовою для побудови інтелектуальних систем управління корпоративними мережами.

2.3 Формалізація зв'язку між онтологічними аксіомами та основними операціями алгебри кортежів

Для забезпечення формальної інтерпретації онтологічних аксіом (2.11)–(2.18) у вигляді процедур обробки даних введемо наступне відображення:

$$\psi : \Omega \rightarrow A, \quad (2.19)$$

де Ω – множина операцій алгебри кортежів; A - множина онтологічних аксіом.

Введене відображення ψ встановлює відповідність між операціями алгебри кортежів та аксіомами, які вони реалізують у процесі розподілу обчислювальних ресурсів корпоративної мережі. На відміну від традиційного використання алгебри кортежів як інструменту маніпулювання даними, у запропонованому підході вона виступає як формальний механізм інтерпретації онтологічних знань, що дозволяє забезпечити єдність декларативного та процедурного рівнів моделі.

Кожна онтологічна аксіома, що задає обмеження або правило функціонування системи, може бути представлена у вигляді відповідної операції або композиції операцій алгебри кортежів. Такий підхід дозволяє перейти від логічного опису до обчислювальної реалізації.

1. Вибірка та аксіома відповідності ресурсів. Операція вибірки $\sigma_{\theta}(R)$ є базовим інструментом фільтрації даних та відповідає первинному етапу

формування множини допустимих рішень. У контексті онтології вона забезпечує перевірку відповідності між вимогами задач та можливостями обчислювальних вузлів. Змістовно ця операція реалізує одну з ключових аксіом системи - можливість виконання задачі лише за умови наявності необхідних ресурсів.

$$Assign(t_i, n_j) \rightarrow Req(t_i) \subseteq Res(n_j), \quad (2.20)$$

$$R_{Valid} = \sigma_{Req(t_i) \subseteq Res(n_j)}(T \times N). \quad (2.21)$$

Таким чином, вона визначає базову область допустимих призначень.

2. З'єднання та аксіоми політик. Операція з'єднання R дозволяє інтегрувати різноманітні джерела знань, що є характерною особливістю онтологічного підходу. У даному випадку вона поєднує інформацію про допустимі призначення задач з інформацією про політики доступу. Це дозволяє врахувати не лише технічні характеристики системи, але й організаційні, безпекові та контекстні обмеження, що є важливим для корпоративних мереж. Операція з'єднання $R \bowtie_{\theta} S$ реалізує аксіому (2.11):

$$Assign(t_i, n_j) \rightarrow Policy(t_i, n_j) = true, \quad (2.22)$$

$$R_{policy} = R_{valid} \bowtie_{Policy(t_i, n_j)} PolicyTable. \quad (2.23)$$

3. Проекція та аксіоми унікальності. Проекція дозволяє виділити суттєві атрибути відношення та усунути надлишкову інформацію. У даному випадку вона використовується для формування множини кандидатів на призначення задач. У поєднанні з операцією усунення дублікатів проекція забезпечує виконання аксіоми унікальності, тобто гарантує однозначність призначення задач.

$$\forall t_i \in T: \exists! n_j \in N Assign(t_i, n_j), \quad (2.24)$$

$$R_{proj} = \pi_{t_i, n_j}(R_{policy}). \quad (2.25)$$

4. Групування та аксіоми обмеженості ресурсів. Групування дозволяє агрегувати дані за певними ознаками та обчислювати сумарні характеристики. У даному випадку воно використовується для визначення сумарного навантаження на кожен вузол. Ця операція є ключовою для реалізації обмежень ресурсів, оскільки дозволяє оцінити допустимість розподілу задач з точки зору доступних ресурсів.

$$\sum_{t_i} \text{Usage}(t_i, n_j) \leq \text{Res}(n_j), \quad (2.26)$$

$$R_{\text{group}} = \gamma_{n_j, \Sigma \text{ Usage}}(R_{\text{proj}}). \quad (2.27)$$

5. Вибірка та аксіоми стану мережі. Дана операція враховує динамічний стан системи та дозволяє виключити з розгляду перевантажені вузли. Таким чином, вона забезпечує адаптивність алгоритму до змін у стані мережі та дозволяє уникнути неефективних або некоректних рішень.

$$\text{Overload}(n_j) \rightarrow \neg \text{Assign}(t_i, n_j), \quad (2.28)$$

$$R_{\text{state}} = \sigma_{\neg \text{Overload}(n_j)}(R_{\text{group}}). \quad (2.29)$$

5. Сортування та аксіоми пріоритетності. Сортування використовується для впорядкування варіантів розподілу відповідно до пріоритетів задач. Це дозволяє забезпечити виконання критичних задач у першу чергу. Операція сортування не змінює множину рішень, але визначає порядок їх розгляду, що є важливим у процедурних алгоритмах.

$$\text{Priority}(t_i) > \text{Priority}(t_k), \quad (2.30)$$

$$R_{\text{sorted}} = \tau_{\text{Priority}(t_i)}(R_{\text{state}}). \quad (2.31)$$

6. Розширена проекція та аксіоми узгодженості. Розширена проекція дозволяє обчислювати нові атрибути на основі існуючих. У даному випадку вона використовується для оновлення стану вузлів після призначення задач. Ця операція забезпечує узгодженість між станом системи та виконаними діями.

$$\text{State}(n_j) = f \left(\text{State}(n_j), \text{Usage}(t_i) \right), \quad (2.32)$$

$$R_{\text{update}} = \pi_{n_j, f(\text{State}, \text{Usage})} (R_{\text{sorted}}). \quad (2.33)$$

7. Зовнішнє з'єднання та аксіоми існування рішення. Зовнішнє з'єднання дозволяє врахувати всі задачі, навіть якщо для деяких з них не знайдено допустимого вузла. Це є важливим для забезпечення повноти моделі та подальшого аналізу невиконаних задач.

$$\forall t_i \in T: \exists n_j \in N, \quad (2.34)$$

$$R_{\text{complete}} = T \bowtie R_{\text{update}}. \quad (2.35)$$

8. Композиція операцій та аксіома оптимальності. Композиція операцій алгебри кортежів дозволяє послідовно застосовувати обмеження, визначені онтологічними аксіомами, та формувати множину допустимих рішень. На завершальному етапі здійснюється вибір оптимального рішення відповідно до функції цілі.

$$R^* = \tau \circ \sigma \circ \gamma \circ \pi \circ \bowtie (T, N), \quad (2.36)$$

$$\phi^* = \arg \min J(\phi). \quad (2.37)$$

Операції алгебри кортежів забезпечують формальну реалізацію онтологічних аксіом та дозволяють представити процес розподілу обчислювальних ресурсів у вигляді послідовності строго визначених операцій. Особливістю запропонованого підходу є те, що:

- онтологічні аксіоми визначають структуру алгоритму;
- алгебра кортежів забезпечує його обчислювальну реалізацію;
- досягається узгодженість між логічним та процедурним рівнями моделі.

Запропонований формалізований зв'язок між онтологічними аксіомами та операціями алгебри кортежів дозволяє забезпечити строгість, узгодженість та ефективність реалізації системи управління обчислювальними ресурсами корпоративної мережі. Такий підхід створює теоретичну основу для побудови інтелектуальних систем, що поєднують семантичне представлення знань з алгоритмічними методами їх обробки.

2.4 Метод онтологічно-керованої конфігурації корпоративної мережі на основі множини задач

У сучасних умовах функціонування корпоративних інформаційно-телекомунікаційних систем ефективність розв'язання прикладних задач значною мірою визначається адекватністю конфігурації мережевої інфраструктури до вимог цих задач. З урахуванням високого рівня гетерогенності мережевих ресурсів, динамічного характеру навантажень, а також необхідності одночасного забезпечення продуктивності, надійності та безпеки, задача конфігурації корпоративної мережі набуває ознак складної багатокритеріальної задачі з великою кількістю обмежень.

У межах даного дослідження пропонується підхід до розв'язання цієї задачі на основі онтології корпоративної мережі, яка задається кортежем (2.1) $O_{net} = \langle D, M, U, P, S, E, A \rangle$, де D - предметна область, M - множина описів елементів мережевої інфраструктури, U - множина об'єктів використання (задач, користувачів, сервісів), P - множина параметрів, S - множина станів мережі, E - множина характеристик виконання задач, A - множина онтологічних аксіом.

У такій постановці процес конфігурації розглядається як відображення задачі

на підмножину елементів онтології, що задовольняє як параметричні вимоги, так і логічні обмеження, зафіксовані в аксіоматичній базі. Розглянемо детальніше процес формалізації множини задач

$$T = \{t_1, t_2, \dots, t_n\}, \quad (2.38)$$

де T - множина задач, що розглядаються в межах корпоративної мережі.

Введення множини T формує формальний простір задач, відносно якого визначається задача конфігурації. У такій постановці конфігурація мережі розглядається як залежна від задачі величина, що визначається її вимогами.

$$t_i = \langle id_i, u_i, p_i, e_i \rangle, \quad (2.39)$$

де id_i - ідентифікатор задачі; $u_i \in U$ - об'єкт використання, асоційований із задачею; $p_i \subseteq P$ - множина параметричних вимог; $e_i \subseteq E$ - множина вимог до характеристик виконання.

Подання задачі у вигляді кортежу забезпечує її безпосередню інтерпретацію в межах онтології O_{net} , що дозволяє розглядати процес конфігурації як операцію над елементами онтологічної моделі. Формалізація множини конфігурацій

$$K = \{k_1, k_2, \dots, k_n\}, \quad (2.40)$$

де K - множина допустимих конфігурацій мережі.

Зазначена множина задає потенційний простір рішень, який підлягає обмеженню шляхом врахування параметричних, структурних та логічних умов.

$$K_i = \langle m_i, p_i^*, s_i, e_i^* \rangle, \quad (2.41)$$

де $m_i \subseteq M$ - множина елементів інфраструктури; $p_i^* \subseteq P$ - параметри

конфігурації; $s_i \subseteq S$ - допустимі стани; $e_i^* \subseteq E$ - очікувані характеристики виконання.

Таким чином, конфігурація інтерпретується як узгоджене відображення структурних і динамічних властивостей мережі, що відповідає вимогам задачі. Розглянемо відображення задачі на конфігурацію

$$\Phi: T \rightarrow K, \quad (2.42)$$

де Φ - оператор конфігурації.

Оператор Φ формалізує процес побудови конфігурації як відображення з простору задач у простір конфігурацій, причому це відображення має бути узгодженим із онтологічною структурою предметної області

$$Valid(t_i, K_i, A) = 1 \Leftrightarrow \forall a_j \in A_{t_i}: a_j(t_i, K_i) = true, \quad (2.43)$$

де $A_{t_i} \subseteq A$ - множина аксіом, релевантних задачі t_i .

Умова допустимості визначає клас конфігурацій, які не суперечать онтологічним обмеженням. Включення аксіом у формалізацію забезпечує логічну коректність процесу конфігурації та узгодженість отриманих рішень із предметною областю.

Розглянемо реляційну інтерпретацію компонент онтології M_{Rel}

$$M_{Rel}(M_{Id}, M_{Type}, M_{Struct}), \quad (2.44)$$

де M_{Id} - ідентифікатор елемента інфраструктури, M_{Type} - його тип, M_{Struct} - структурний опис.

Відношення M_{Rel} формує базову множину ресурсів, що можуть бути використані під час побудови конфігурації

$$U_{Rel}(U_{Id}, U_{Type}, U_{Struct}), \quad (2.45)$$

де U_{Id} - ідентифікатор об'єкта використання. Це відношення задає семантичний зв'язок між задачами та об'єктами, що ініціюють вимоги до мережі.

$$P_{Rel}(P_{Id}, P_{Type}, P_{Value}), \quad (2.46)$$

де P_{Id} - ідентифікатор параметра, P_{Value} - його значення.

Параметричне представлення забезпечує можливість кількісного оцінювання відповідності конфігурації вимогам задачі.

$$S_{Rel}(S_{Id}, S_{Cond}), \quad (2.47)$$

де S_{Cond} - опис стану мережі.

Наступне відношення відображає динамічний аспект функціонування мережі.

$$E_{Rel}(E_{Id}, E_{Main}, E_{Alt}, E_{Rule}), \quad (2.48)$$

де E_{Main} - основні характеристики, E_{Alt} - альтернативні характеристики, E_{Rule} - правила інтерпретації.

Відношення E_{Rel} дозволяє формалізувати критерії ефективності виконання задач.

$$Task_{Rel}(T_{Id}, U_{Id}, P_{Id}, E_{Id}), \quad (2.49)$$

Відношення (2.49) забезпечує зв'язок між задачами та відповідними компонентами онтології.

Виходячи із запропонованих відношень (2.38)-(2.49) пропонується

наступний формалізований алгоритм конфігурації корпоративної мережі.

$$R_1 = \sigma_{M_{Struct}=p_i}(M_{Rel}), \quad (2.50)$$

де σ - операція вибірки, а символ \models інтерпретується як відношення відповідності між структурним описом ресурсу та вимогами задачі.

Операція R_1 формує підмножину елементів інфраструктури, для яких виконується узгодження структурних характеристик із параметричними вимогами задачі. У результаті здійснюється звуження початкового простору ресурсів до множини потенційно релевантних.

$$R_2 = R_1 \bowtie_{M_{Id}=U_{Id}} U_{Rel}, \quad (2.51)$$

де \bowtie - операція з'єднання відношень.

Виконання з'єднання забезпечує узгодження відібраних ресурсів із об'єктами використання, що дозволяє врахувати контекст задачі в межах онтології.

$$R_3 = R_2 \bowtie_{U_{Id}=P_{Id}} P_{Rel}, \quad (2.52)$$

Ця операція розширює множину кандидатних конфігурацій шляхом включення параметричних характеристик, необхідних для подальшого аналізу відповідності.

$$R_4 = \sigma_{P_{Type} \in ReqTypes(t_i) \wedge P_{Value} \geq ReqVals(t_i)}, \quad (2.53)$$

де $ReqTypes(t_i)$ та $ReqVals(t_i)$ визначають відповідно типи та граничні значення параметрів задачі.

Застосування вибірки формує підмножину конфігурацій, що задовольняють параметричні обмеження, визначені задачею.

$$R_5 = R_4 \bowtie_{M_{Id}=S_{Id}} S_{Rel}, \quad (2.54)$$

З'єднання з відношенням станів дозволяє врахувати поточні або допустимі режими функціонування мережі.

$$R_6 = \sigma_{S_{Cond} \in AllowedStates(t_i)} (R_5), \quad (2.55)$$

де $AllowedStates(t_i)$ - множина допустимих станів для задачі t_i .

Виконується відбір конфігурацій, які відповідають допустимим режимам експлуатації.

$$R_7 = R_6 \bowtie_{E_{Id}} E_{Rel}, \quad (2.56)$$

Відношення (2.56) доповнює конфігурації характеристиками ефективності виконання задачі

$$R_8 = \sigma_{a_1 \wedge \dots \wedge a_q} (R_7), \quad (2.57)$$

де $a_j \in A_{t_i}$ - онтологічні аксіоми.

Застосування даної операції реалізує перевірку логічної узгодженості конфігурацій відповідно до аксіоматичної бази онтології.

$$ConfigRel_i = \pi_{M_{Id}, U_{Id}, P_{Id}, S_{Id}, E_{Id}} (R_8), \quad (2.58)$$

де π - операція проекції.

У результаті формується відношення, що визначає структуру конфігурації через релевантні атрибути.

$$J(ConfigRel_i) = \alpha_1 J_P + \alpha_2 J_S + \alpha_3 J_E, \quad (2.59)$$

де $\alpha_1, \alpha_2, \alpha_3$ - вагові коефіцієнти критеріїв, J_P, J_S, J_E - часткові оцінки.

Функція J забезпечує інтегральне оцінювання якості конфігурацій з урахуванням різних аспектів функціонування.

$$K_i^* = \arg \min_{K_i \in \Phi(t_i)} J(K_i) , \quad (2.60)$$

Оператор $\arg \min$ визначає конфігурацію, яка забезпечує мінімальне значення цільової функції серед множини допустимих рішень.

На рисунках 2.6 та 2.7 подано загальну схему реалізації методу онтологічно-керованої конфігурації корпоративної мережі, яка складається з декларативної та процедурної частин.

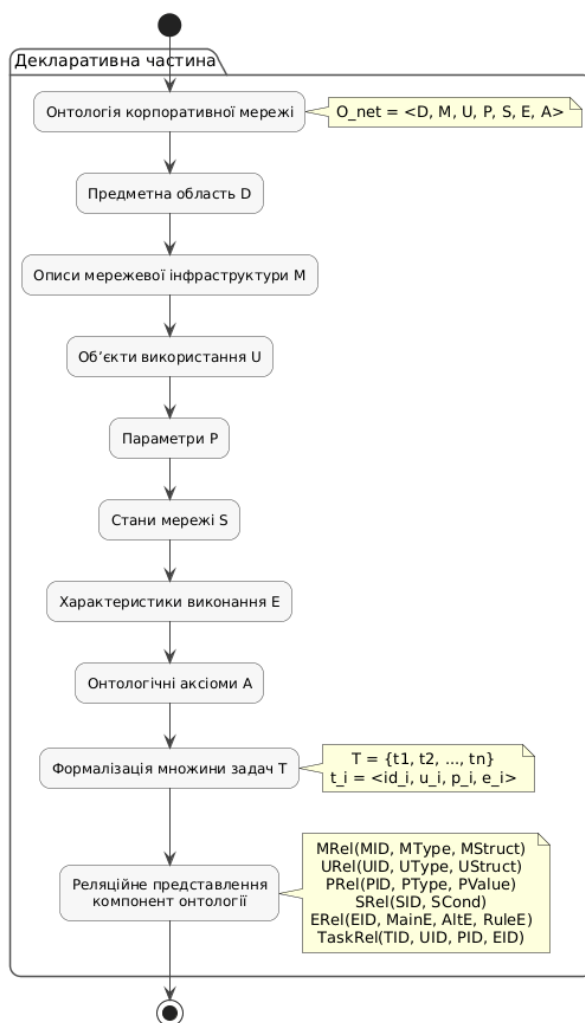


Рис. 2.6. Схема реалізації методу онтологічно-керованої конфігурації корпоративної мережі (декларативна частина)

У межах декларативної частини формується онтологічний опис предметної області, що включає структуру мережі, об'єкти використання, параметри, стани, характеристики виконання та множину аксіом.

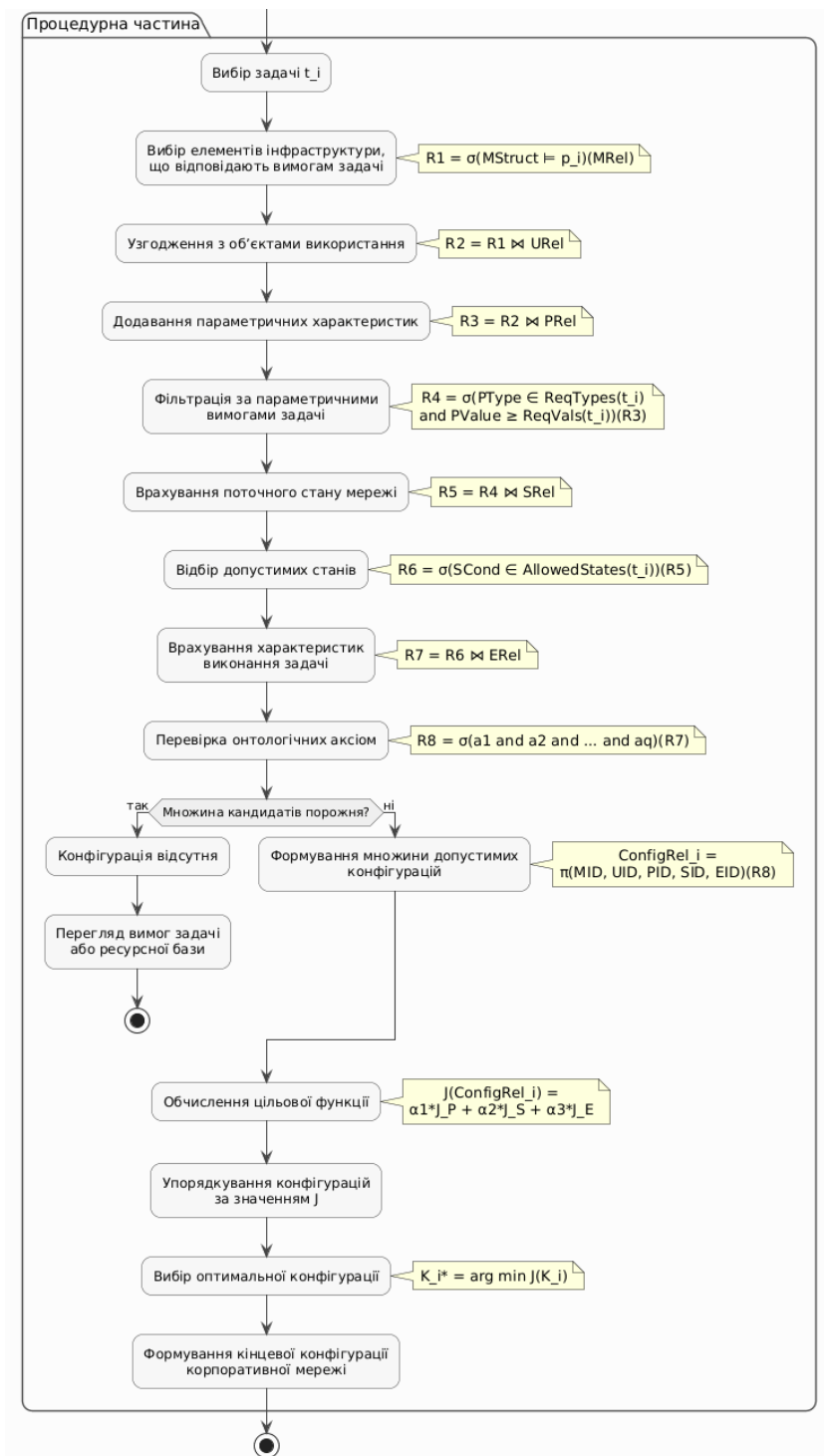


Рис. 2.7. Схема реалізації методу онтологічно-керованої конфігурації корпоративної мережі (процедурна частина)

Процедурна частина реалізує відображення задачі на множину допустимих конфігурацій шляхом послідовного застосування операцій алгебри кортежів, перевірки онтологічних аксіом і вибору оптимальної конфігурації відповідно до цільової функції.

Запропонований метод конфігурації корпоративної мережі базується на інтеграції онтологічного підходу та апарату алгебри кортежів, що дозволяє формалізувати процес підбору конфігурації як послідовність строгих операцій над відношеннями. На рисунку 2.8 представлено загальна схема методу онтологічно-керованого конфігурування корпоративної комп'ютерної мережі. На відміну від традиційних підходів, у яких враховуються переважно параметричні характеристики, у запропонованому методі суттєву роль відіграє множина онтологічних аксіом, яка визначає допустимість конфігурацій з точки зору логічних, функціональних і безпекових обмежень.

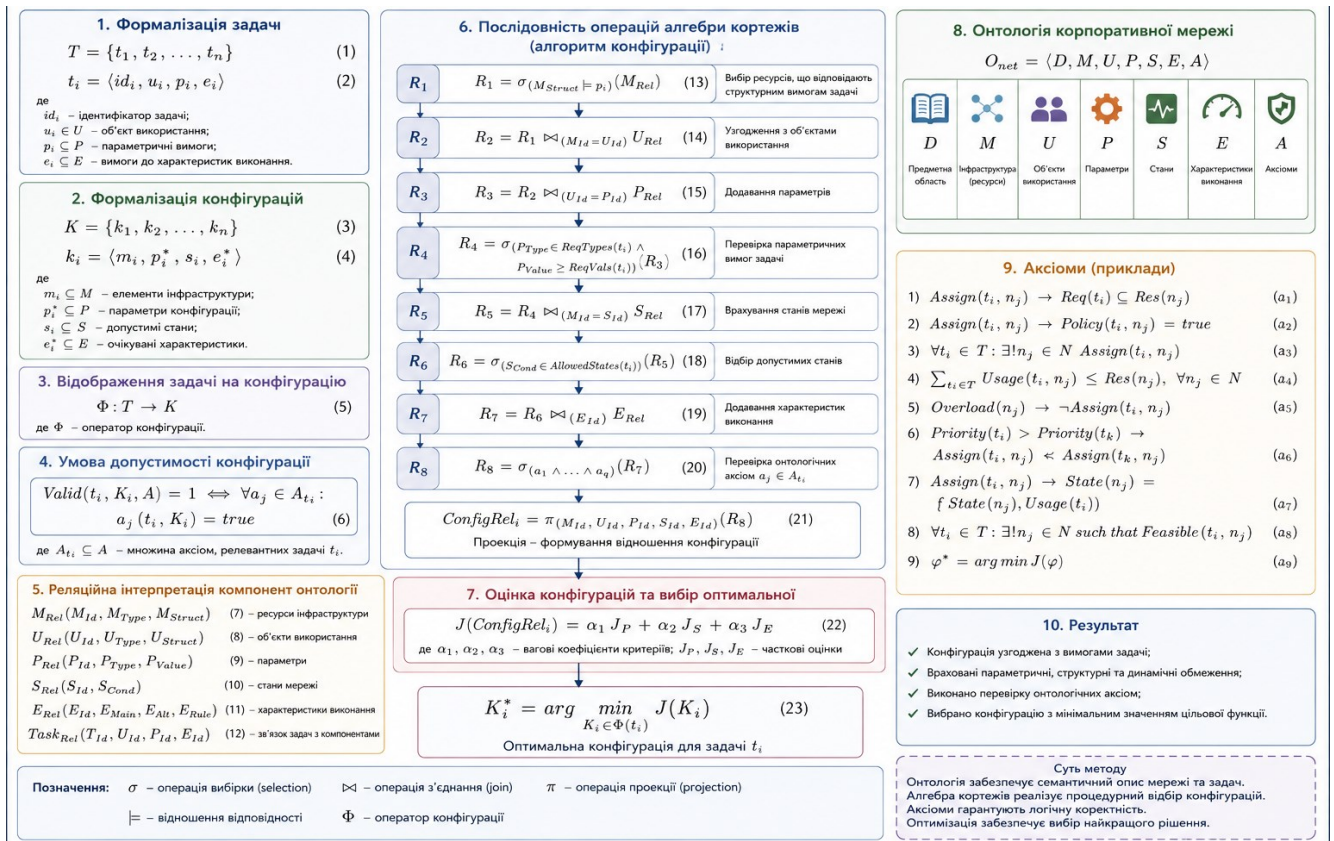


Рис. 2.9. Метод онтологічно-керованого конфігурування корпоративної комп'ютерної мережі

Такий підхід забезпечує не лише узгодженість конфігурації з вимогами задачі, але й її відповідність структурі предметної області, що описується онтологією O_{net} . Використання реляційного представлення та операцій алгебри кортежів створює передумови для ефективної програмної реалізації методу та його інтеграції в системи підтримки прийняття рішень.

У результаті отримано формалізований метод, який можна розглядати як теоретичну основу для побудови інтелектуальних систем автоматичної конфігурації корпоративних мереж, здатних адаптуватися до змінних умов функціонування та різномірних типів задач, що розв'язуються в межах складних інформаційних систем.

Висновки до розділу 2

1. Розроблено математичне та методологічне забезпечення знання-орієнтованої системи управління корпоративними комп'ютерними мережами. Проведено формалізацію корпоративної мережі у вигляді онтологічної моделі, яка забезпечує структуроване подання знань про мережеві ресурси, користувачів, сервіси, політики доступу та задачі, що виконуються в мережевому середовищі. Запропонована модель створює єдиний інформаційний простір для накопичення, обробки та використання знань у процесах управління мережею.

2. На основі розробленої онтології сформовано формальний опис множини задач, конфігурацій та відношень між ними. Використання апарату алгебри кортежів дозволило формалізувати процеси пошуку, відбору та комбінування мережевих ресурсів відповідно до вимог користувачів і поточних умов функціонування корпоративної мережі. Це забезпечує можливість автоматизованого формування допустимих конфігурацій та перевірки їхньої відповідності встановленим обмеженням і правилам.

3. Розроблено метод онтологічно-керованого конфігурування корпоративної комп'ютерної мережі, який базується на використанні онтологічної

бази знань, механізмів логічного виведення та формалізованих правил прийняття рішень. Запропонований метод забезпечує автоматичний аналіз доступних ресурсів, перевірку політик доступу, виявлення конфліктів конфігурацій та формування оптимальних варіантів розподілу ресурсів між задачами.

4. Отримані результати створюють теоретичне підґрунтя для побудови програмної реалізації знання-орієнтованої системи управління корпоративними комп'ютерними мережами, розробка архітектури та програмних компонентів якої розглядається у наступному розділі. Запропоновані моделі та методи забезпечують підвищення ефективності використання мережевих ресурсів, скорочення часу адміністрування та підтримку інтелектуального прийняття рішень у процесах управління корпоративною мережею.

РОЗДІЛ 3

АРХІТЕКТУРА ЗНАННЯ-ОРІЄНТОВАНОЇ СИСТЕМИ УПРАВЛІННЯ КОРПОРАТИВНИМИ КОМП'ЮТЕРНИМИ МЕРЕЖАМИ

У даному розділі розроблено архітектуру знання-орієнтованої системи управління корпоративними комп'ютерними мережами, яка забезпечує інтеграцію онтологічного представлення знань про структуру мережі, її ресурси, сервіси, користувачів та множину задач із механізмами логічного виведення, підтримки прийняття рішень і багатокритеріальної оптимізації. Особливу увагу приділено формуванню програмного середовища, здатного реалізувати запропоновані в роботі математичні моделі та методи в умовах функціонування реальної корпоративної мережевої інфраструктури.

У розділі обґрунтовано концепцію побудови багаторівневої програмної архітектури, яка забезпечує розподіл функцій між підсистемами моніторингу, управління знаннями, логічного виведення, конфігурування та оптимізації мережевих ресурсів. Розглянуто особливості взаємодії програмних компонентів, організацію потоків даних між ними, а також принципи інтеграції онтологічної бази знань із підсистемами моніторингу та підтримки прийняття рішень. Представлено структурні та функціональні моделі системи, які відображають взаємозв'язки між користувачами, мережевими ресурсами, сервісами та програмними модулями.

Значну увагу приділено програмній реалізації онтологічної бази знань та механізму логічного виведення. Описано структуру онтологічного сховища, принципи формалізації знань про корпоративну мережу, способи представлення концептів, відношень та аксіом предметної області. Наведено особливості реалізації механізму логічного виведення, який забезпечує автоматичне формування нових знань на основі фактів, отриманих від підсистеми моніторингу, а також підтримує побудову множини допустимих конфігураційних рішень відповідно до політик функціонування корпоративної мережі.

Окремий підрозділ присвячено реалізації математичної моделі розподілу навантаження між центрами термінальних кластерів на основі багатокритеріальної оптимізації та онтологічного подання знань про ресурси корпоративної комп'ютерної мережі й множину задач. Показано особливості програмної інтеграції моделі оптимізації з онтологічною базою знань та механізмом логічного виведення. Розглянуто процес формування множини допустимих конфігурацій, їх оцінювання за сукупністю критеріїв ефективності та вибір субоптимального рішення для розподілу навантаження між обчислювальними ресурсами корпоративної мережі.

Також у розділі обґрунтовано вибір технологій та програмних засобів реалізації системи. Розглянуто використання сучасних веборієнтованих технологій для побудови користувачького інтерфейсу, серверних засобів реалізації бізнес-логіки, технологій роботи з онтологіями та засобів зберігання даних. Наведено особливості реалізації програмних інтерфейсів взаємодії між компонентами системи, механізмів обміну даними та інтеграції із засобами моніторингу корпоративної мережі.

Основні результати цього розділу опубліковані в працях [27, 65, 116, 117, 118, 119].

3.1 Проектування архітектури знання-орієнтованої системи управління корпоративними комп'ютерними мережами

У попередньому розділі було розроблено онтологічну модель корпоративної комп'ютерної мережі, метод онтологічно-керованого конфігурування та математичну модель розподілу навантаження між центрами термінальних кластерів на основі багатокритеріальної оптимізації. Практичне використання запропонованих моделей потребує створення відповідного програмного середовища, здатного забезпечити інтеграцію онтологічного представлення знань, механізмів логічного виведення, засобів моніторингу мережевої інфраструктури та

алгоритмів підтримки прийняття рішень у межах єдиної програмної системи.

Особливістю розроблюваної системи є необхідність одночасної роботи з великою кількістю взаємопов'язаних сутностей, до яких належать мережеві ресурси, користувачі, сервіси, політики безпеки, конфігурації обладнання та множина задач, що розв'язуються в корпоративному середовищі. На відміну від традиційних систем моніторингу та адміністрування мереж, які оперують переважно окремими параметрами функціонування мережевих пристроїв, запропонована система використовує знання-орієнтований підхід, що дозволяє враховувати семантичні взаємозв'язки між усіма компонентами корпоративної мережі та автоматизувати процес прийняття управлінських рішень.

Концепція побудови системи базується на інтеграції трьох основних складових: онтологічної бази знань, механізму логічного виведення та математичної моделі оптимального розподілу навантаження. Такий підхід дозволяє здійснювати перехід від традиційного управління ресурсами до управління знаннями про ресурси та задачі мережевого середовища.

Загальну структуру знання-орієнтованої системи можна подати у вигляді:

$$SYS = \langle M, O, R, D, U \rangle, \quad (3.1)$$

де M — підсистема моніторингу мережевої інфраструктури; O — онтологічна база знань; R — підсистема логічного виведення та підтримки прийняття рішень; D — модуль оптимізації та розподілу навантаження; U — підсистема взаємодії з користувачем.

Архітектура системи реалізується за багаторівневим принципом, що забезпечує модульність, масштабованість та незалежність окремих компонентів, рисунок 3.1.

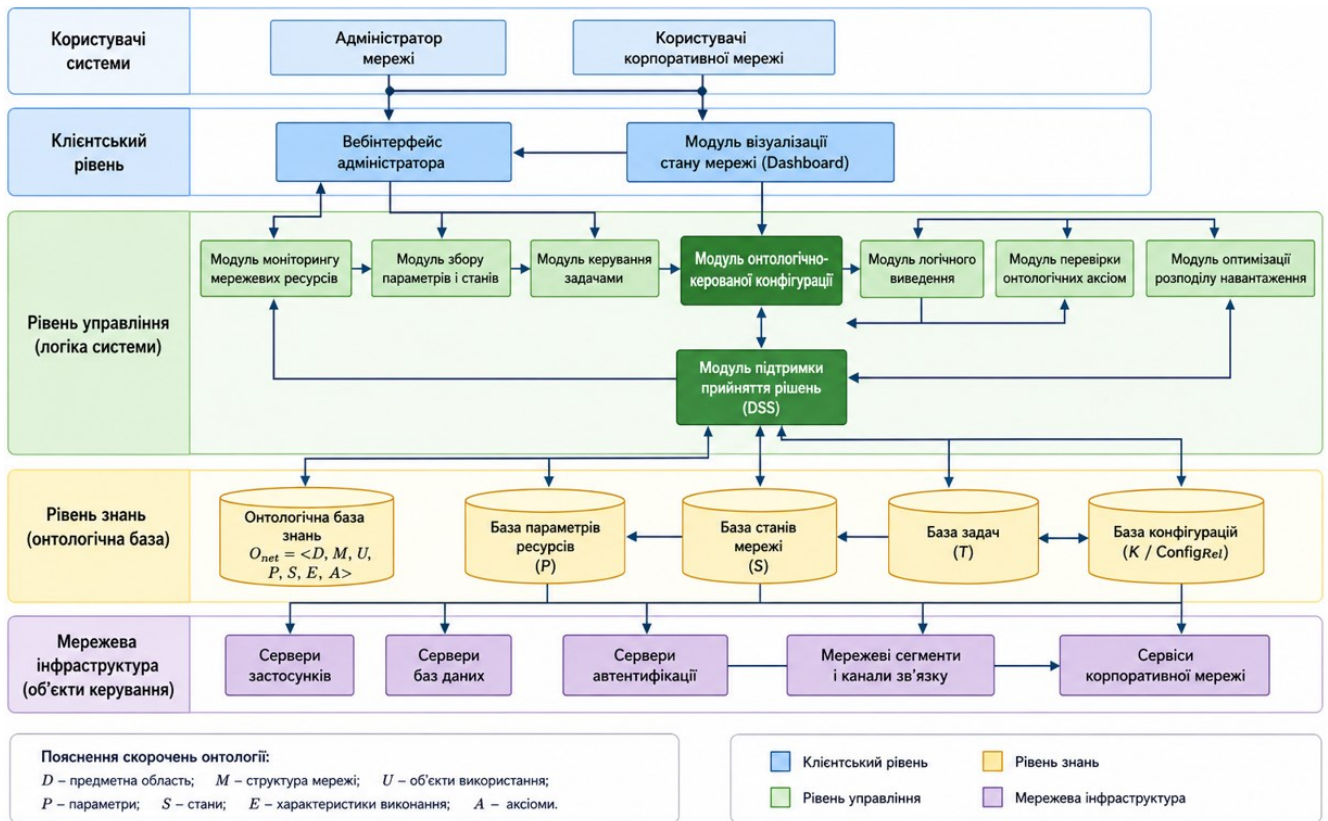


Рис. 3.1. Загальна архітектура онтологічно-керованої системи управління корпоративними комп'ютерними мережами.

З позиції функціонального призначення центральним елементом архітектури виступає онтологічна база знань, яка містить формалізований опис ресурсів мережі, сервісів, користувачів, політик безпеки та множини задач. Саме на її основі здійснюється логічне виведення нових знань та формування допустимих конфігурацій мережевого середовища.

Для забезпечення адаптивного управління мережею всі функціональні модулі працюють у тісній взаємодії між собою. Підсистема моніторингу безперервно збирає інформацію про поточний стан мережевих ресурсів та передає її до онтологічної бази знань. На основі цих даних механізм логічного виведення формує множину допустимих конфігураційних рішень, після чого модуль оптимізації виконує пошук найбільш ефективного варіанта розподілу ресурсів.

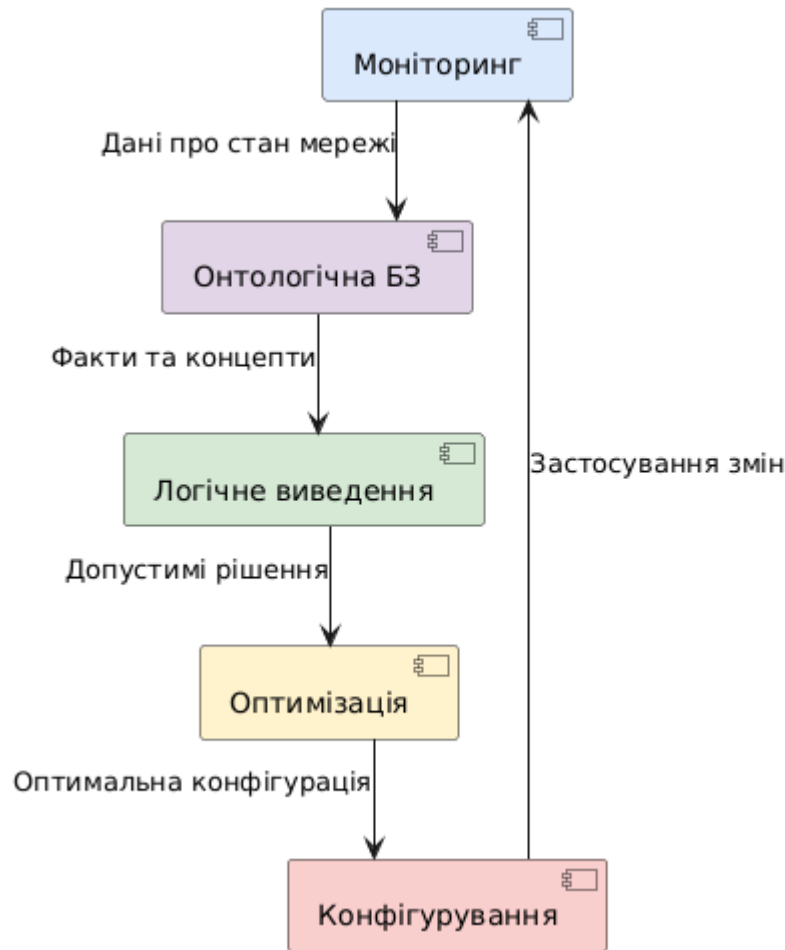


Рис. 3.2. Діаграма взаємодії модулів.

Запропонована схема забезпечує замкнений цикл управління мережею, в якому результати моніторингу безпосередньо впливають на процес прийняття конфігураційних рішень.

Важливою особливістю системи є використання потокового оброблення даних. Потоки інформації надходять від мережевих пристроїв, серверів, систем моніторингу та користувачів. Отримані дані проходять етапи нормалізації, семантичного аналізу та інтеграції до онтологічної бази знань. Формально процес опрацювання даних можна подати у вигляді:

$$D_t \rightarrow O_{net} \rightarrow R_t \rightarrow K_t, \quad (3.2)$$

де D_t - потік даних про поточний стан мережі; O_{net} - онтологія корпоративної

мережі; R_t - результати логічного виведення; K_t - конфігураційні рішення.

Особливістю запропонованої архітектури є інтеграція математичної моделі розподілу навантаження між центрами термінальних кластерів безпосередньо в контур підтримки прийняття рішень. На відміну від класичних балансувальників навантаження, які використовують лише локальні показники завантаженості ресурсів, запропонований підхід враховує знання про ресурси мережі, сервіси та множину задач, представлені в онтології.

Робота системи реалізована у вигляді послідовності взаємопов'язаних етапів, які забезпечують автоматизований аналіз стану мережі, формування конфігураційних рішень, оптимізацію розподілу ресурсів та підтримку прийняття рішень, рисунок 3.3.



Рис. 3.3. Діаграма варіантів використання системи

На першому етапі здійснюється моніторинг корпоративної мережі. Система виконує збір даних про стан мережевих ресурсів, сервісів та каналів зв'язку, а також формує актуальні параметри функціонування мережі. Джерелами інформації

виступають мережеві вузли, сервери та сервіси, які постійно генерують дані про свій поточний стан.

Другий етап пов'язаний з формуванням та підтримкою онтологічної бази знань. На цьому рівні виконується опис ресурсів, сервісів, мережевих сегментів та множини задач у вигляді онтології корпоративної мережі. Після актуалізації бази знань застосовуються механізми логічного виведення та перевірки онтологічних аксіом, що дозволяє формалізувати знання про мережу та забезпечити їх узгодженість.

На третьому етапі здійснюється формування конфігурацій мережі. Для заданої задачі визначаються її вимоги, цілі та обмеження, після чого на основі онтологічної бази знань будується множина допустимих конфігурацій. Отримані конфігурації оцінюються за інтегральним критерієм якості, що враховує продуктивність, завантаженість ресурсів, політики доступу та інші характеристики функціонування мережі. У результаті формується оптимальна конфігурація (К).

Четвертий етап передбачає оптимізацію та прийняття рішень. На цьому рівні виконується розподіл навантаження між вузлами мережі, аналізуються можливі варіанти конфігурацій та формуються рекомендації для системи підтримки прийняття рішень. Таким чином забезпечується адаптивне управління ресурсами відповідно до поточного стану мережі та вимог користувачів.

П'ятий етап полягає у застосуванні сформованої конфігурації. Система автоматично або напівавтоматично виконує налаштування мережевих вузлів і сервісів, розгортає нову конфігурацію в мережевому середовищі та здійснює контроль її працездатності.

Шостий етап забезпечує підтримку прийняття рішень керівником або адміністратором мережі. Система формує рекомендації щодо покращення функціонування мережі, пропонує альтернативні конфігурації та дозволяє оцінити наслідки їх застосування до впровадження.

Завершальним етапом є формування звітності та аналітики. Система генерує звіти про стан мережевої інфраструктури, використання ресурсів, результати

конфігурування та показники якості обслуговування користувачів. Це дозволяє оцінювати ефективність роботи корпоративної мережі та контролювати виконання встановлених показників якості сервісів.

На рисунку 3.4 представлено діаграму розгортання розробленої знання-орієнтованої системи управління корпоративними комп'ютерними мережами, яка відображає фізичну структуру програмно-апаратних компонентів та їх взаємодію під час функціонування системи.

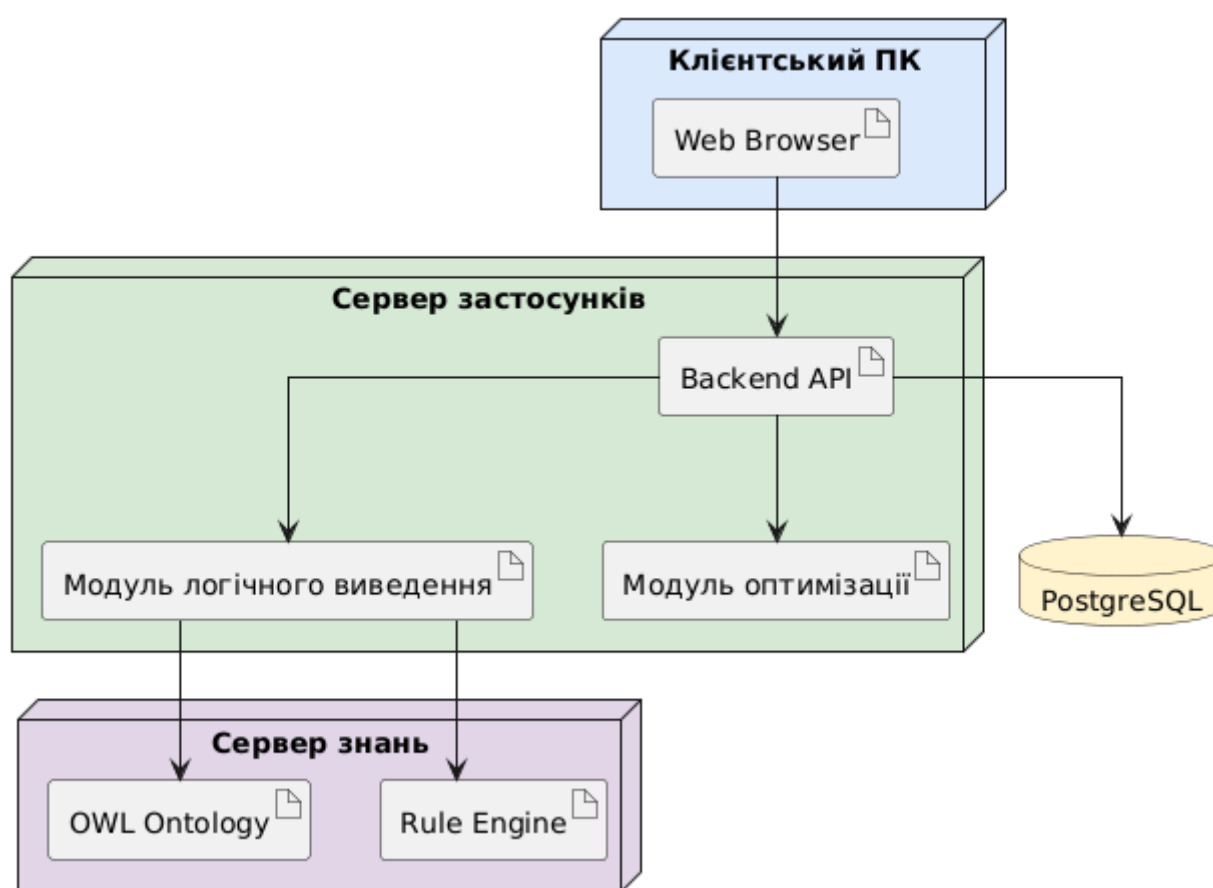


Рис. 3.4. Діаграма розгортання системи

Архітектура системи побудована за клієнт-серверним принципом та складається з трьох основних рівнів: рівня користувацького доступу, рівня обробки бізнес-логіки та рівня представлення і зберігання знань. Користувач взаємодіє із системою за допомогою веббраузера, який функціонує на клієнтському комп'ютері та забезпечує доступ до всіх функціональних можливостей системи через

вебінтерфейс.

Основна обробка запитів здійснюється на сервері застосунків, який містить Backend API та спеціалізовані функціональні модулі. Backend API виконує роль центрального координатора взаємодії між усіма компонентами системи, забезпечуючи приймання запитів користувачів, доступ до бази даних, виклик механізмів логічного виведення та модулів оптимізації.

До складу сервера застосунків входить модуль логічного виведення, який реалізує механізми аналізу онтологічної бази знань, обробки правил предметної області та автоматичного формування конфігураційних рішень. Даний модуль використовує семантичні зв'язки між мережевими ресурсами, сервісами, користувачами та множиною задач для отримання нових знань про поточний стан корпоративної мережі.

Важливим компонентом системи є модуль оптимізації, який реалізує математичну модель багатокритеріального розподілу навантаження між центрами термінальних кластерів. Модуль використовує результати логічного виведення та інформацію про поточний стан мережевої інфраструктури для формування субоптимальних конфігурацій розподілу ресурсів за критеріями продуктивності, збалансованості використання ресурсів та якості обслуговування користувачів.

Окремим вузлом архітектури виступає сервер знань, який містить онтологічну модель корпоративної мережі, реалізовану засобами OWL Ontology, та Rule Engine — механізм обробки правил і аксіом онтології. Така організація забезпечує централізоване зберігання знань про структуру мережі, її ресурси, сервіси, політики безпеки та множини задач, що розв'язуються в корпоративному середовищі. Зберігання оперативної інформації, конфігураційних даних, результатів моніторингу та журналів подій здійснюється в СУБД PostgreSQL, яка забезпечує надійне зберігання даних та підтримує ефективну взаємодію між усіма програмними компонентами системи.

Запропонована архітектура створює технологічне підґрунтя для реалізації онтологічно-керованого управління корпоративними комп'ютерними мережами та

забезпечує інтеграцію знань про ресурси, сервіси та задачі з механізмами логічного виведення й багатокритеріальної оптимізації розподілу навантаження між центрами термінальних кластерів.

3.2 Реалізація онтологічної бази знань та механізму логічного виведення

Однією з ключових особливостей розробленої системи управління корпоративними комп'ютерними мережами є використання онтологічної бази знань як центрального компонента підтримки прийняття рішень. На відміну від традиційних систем мережевого адміністрування, які оперують переважно наборами конфігураційних параметрів та журналами подій, запропонована система використовує формалізоване представлення знань про структуру мережі, її ресурси, сервіси, політики безпеки та множину задач, що розв'язуються користувачами. З програмної точки зору онтологічна база знань реалізована як окрема підсистема, інтегрована із сервісами моніторингу, конфігурування та оптимізації мережі. Її основним призначенням є накопичення, структуризація та аналіз знань про предметну область, а також забезпечення механізмів автоматичного логічного виведення для формування управлінських рішень.

Архітектура підсистеми реалізована відповідно до принципів Service-Oriented Architecture (SOA) та передбачає розділення функцій зберігання знань, виконання логічного виведення та взаємодії з іншими компонентами системи, рисунок 3.5.

Підсистема управління знаннями складається з чотирьох основних компонентів:

- сервісу управління онтологією;
- онтологічного сховища знань;
- механізму логічного виведення;
- підсистеми інтеграції з модулями конфігурування та оптимізації.

Основним елементом реалізації виступає сервіс управління онтологією, який відповідає за формування, актуалізацію та модифікацію знань про корпоративну

мережу. Саме цей сервіс забезпечує синхронізацію між даними моніторингу та онтологічним представленням мережевого середовища.

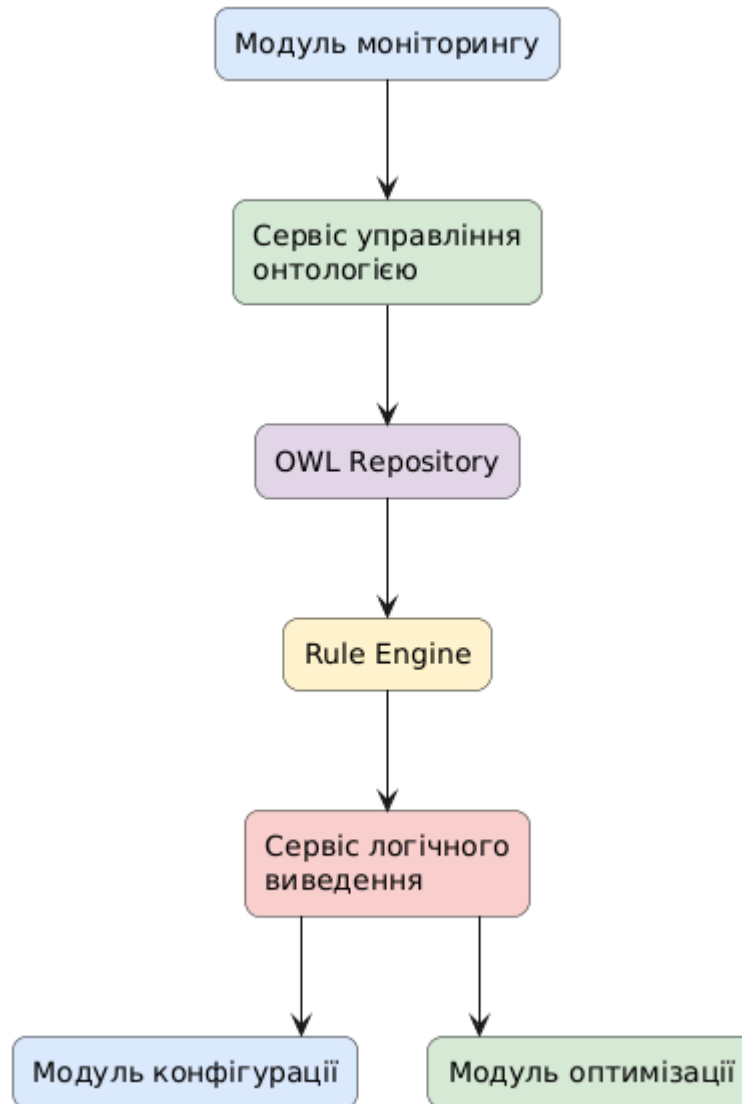


Рис. 3.5. Структура підсистеми управління знаннями

Онтологічне сховище реалізоване у вигляді окремого репозиторію знань, що містить формалізований опис усіх сутностей корпоративної мережі. Структура знань включає:

- мережеві ресурси;
- сервери та центри термінальних кластерів;
- користувачів;

- сервіси;
- політики безпеки;
- конфігурації;
- множину задач.

Зберігання онтології здійснюється у форматі RDF/OWL, що забезпечує незалежність від конкретної СУБД та можливість інтеграції зі стандартними засобами Semantic Web. Для кожного об'єкта підтримується набір атрибутів та семантичних зв'язків, які автоматично оновлюються під час отримання нових даних від підсистеми моніторингу. Програмно кожен концепт реалізований у вигляді окремого класу.

Реалізація механізму логічного виведення. Особливістю запропонованої системи є використання окремого модуля логічного виведення, який функціонує поверх онтологічної бази знань, рисунок 3.6.

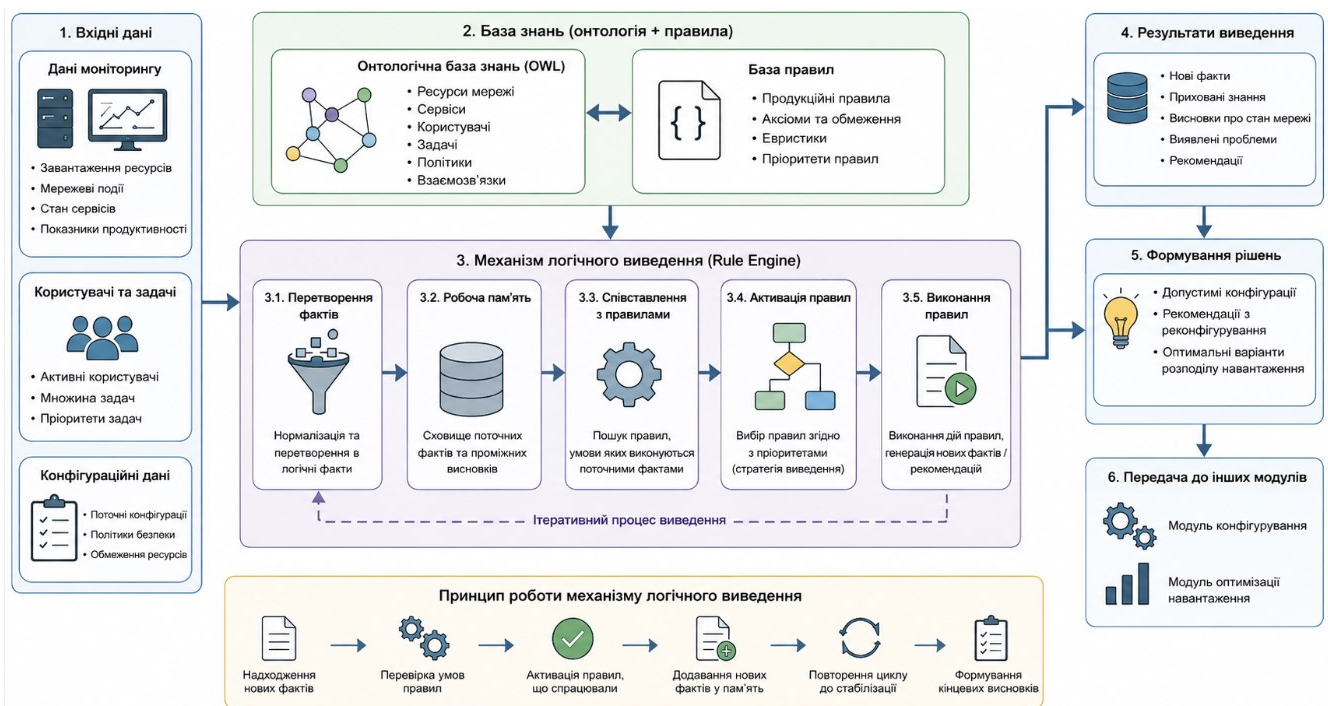


Рис. 3.6. Підсистема реалізації логічного виведення

З програмної точки зору механізм логічного виведення реалізовано як Rule Engine, що підтримує:

- аналіз фактів;
- перевірку аксіом;
- застосування правил;
- формування нових знань;
- генерацію рекомендацій.

На рисунку 3.7 представлено процес інтеграції підсистеми моніторингу з онтологічною базою знань та механізмом логічного виведення. Дані про поточний стан мережевих ресурсів надходять до підсистеми моніторингу, де проходять етапи збору, нормалізації та попереднього аналізу. Після цього інформація інтегрується до онтологічної бази знань у вигляді фактів про ресурси, сервіси, користувачів та задачі корпоративної мережі.

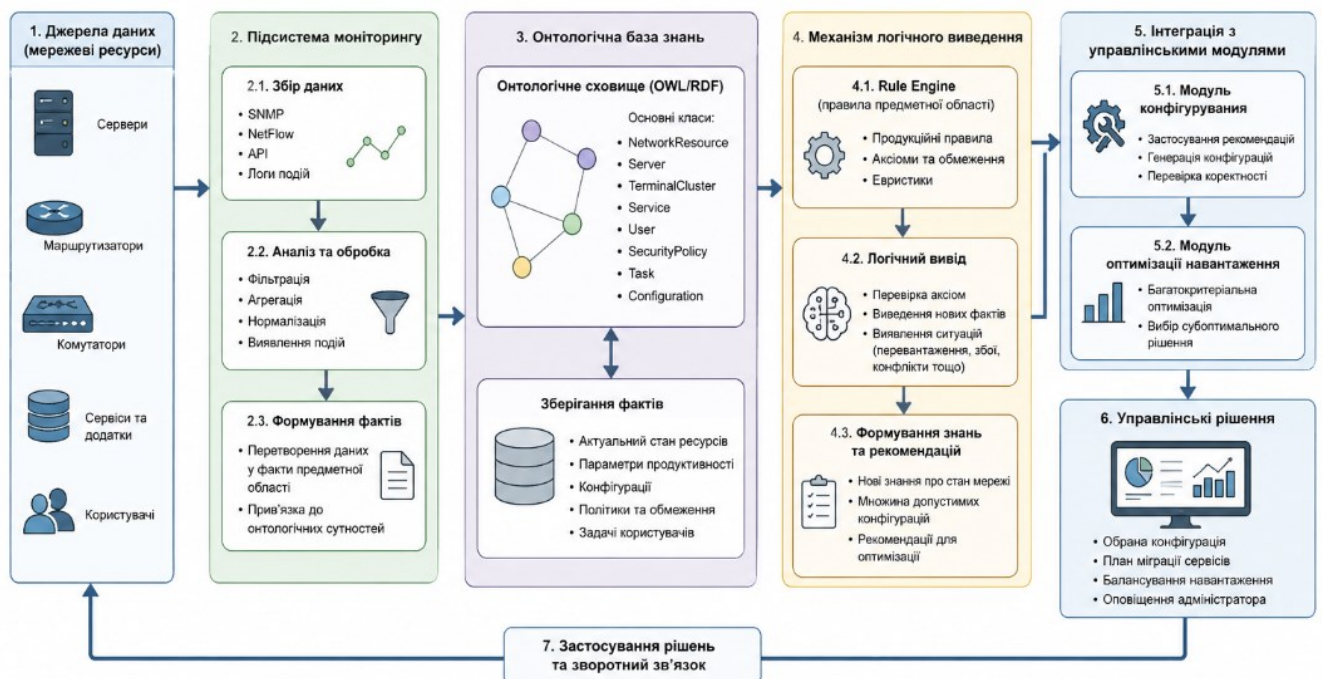


Рис. 3.7. Інтеграція підсистеми моніторингу з онтологічною базою знань та механізмом логічного виведення

На основі актуалізованої онтології механізм логічного виведення виконує перевірку аксіом та правил предметної області, формуючи нові знання про стан мережі та множину допустимих конфігураційних рішень. Отримані результати

передаються до модулів конфігурування та багатокритеріальної оптимізації навантаження, які генерують рекомендації щодо реконфігурації мережевої інфраструктури. Сформовані управлінські рішення застосовуються до мережевих ресурсів, після чого цикл моніторингу повторюється, забезпечуючи адаптивне та знання-орієнтоване управління корпоративною комп'ютерною мережею.

У результаті реалізації онтологічної бази знань та механізму логічного виведення створено програмну підсистему, яка забезпечує формалізацію знань про корпоративну комп'ютерну мережу, автоматичне виявлення взаємозв'язків між ресурсами, сервісами та задачами, підтримку процесів онтологічно-керованого конфігурування та формування множини допустимих конфігураційних рішень для подальшої багатокритеріальної оптимізації розподілу навантаження. Це створює основу для реалізації адаптивного та пояснюваного управління корпоративною мережею на основі знань.

3.3 Математичну модель розподілу навантаження між центрами термінальних кластерів

У сучасних умовах функціонування корпоративних мереж, які є типовими представниками складних інформаційно-комунікаційних систем, спостерігається суттєве зростання неоднорідності навантаження, варіативності профілів запитів та часової нестабільності потоків даних. Це призводить до виникнення локальних перевантажень, деградації показників якості обслуговування та неефективного використання обчислювальних і комунікаційних ресурсів. Зазначені фактори унеможливають застосування класичних стратегій балансування навантаження, орієнтованих на спрощені евристики, які не враховують семантичну структуру запитів, контекст їх обробки та знання про поведінку системи [27, 72, 116].

У цьому контексті виникає необхідність формалізації задачі розподілу навантаження як складної динамічної системи, у якій множина запитів, що надходять у мережу, характеризується не лише кількісними параметрами

(інтенсивність, тривалість обробки), але й якісними характеристиками (тип сервісу, пріоритет, залежності, контекст використання). Відповідно, множина обчислювальних вузлів (центрів термінальних кластерів) повинна розглядатися як гетерогенне середовище з обмеженими ресурсами, змінними станами та різними функціональними ролями. За таких умов задача розподілу навантаження трансформується у задачу багатокритеріальної оптимізації в умовах невизначеності, де необхідно забезпечити компроміс між продуктивністю, збалансованістю та стабільністю функціонування системи.

Особливу складність становить розроблення динамічної моделі розподілу навантаження для центрів термінальних кластерів, які виконують роль критичних точок обробки високоінтенсивних потоків користувацьких запитів у навчальному середовищі. Така модель повинна враховувати часову еволюцію навантаження, топологічні особливості мережі, поведінкові патерни користувачів, а також можливість прогнозування майбутніх станів системи. При цьому принципово важливим є включення знання-орієнтованої складової, яка дозволяє формалізувати експертні уявлення про закономірності функціонування мережі у вигляді онтологічних структур і правил логічного виведення.

Інтеграція такої моделі у програмну архітектуру системи управління супроводжується низкою фундаментальних викликів. По-перше, необхідно забезпечити ефективну взаємодію між онтологічним ядром (що реалізує семантичне представлення знань і механізми reasoning) та аналітичними компонентами, відповідальними за обробку потокових даних і виконання оптимізаційних процедур. По-друге, постає задача синхронізації статичних знань і динамічних даних, що надходять у реальному часі, з метою підтримки актуальності онтологічної моделі (ABox) без порушення її консистентності. По-третє, необхідно мінімізувати обчислювальні витрати, пов'язані з логічним виведенням, що особливо критично в умовах високонавантажених систем.

Окремого розгляду потребує проблема програмної інтеграції онтологічно-керованих механізмів у наявні системи управління мережами. Існуючі

інфраструктури, як правило, базуються на усталених протоколах моніторингу та керування, що не передбачають використання семантичних моделей і логічного виведення. Відповідно, інтеграція повинна забезпечувати сумісність із традиційними засобами збору та обробки даних, підтримку стандартних інтерфейсів взаємодії, а також можливість поетапного впровадження без порушення функціонування діючих систем. Додатково виникають вимоги до забезпечення безпеки, керованості доступу та відмовостійкості в умовах розширення функціональності системи.

Таким чином, проблема полягає у розробленні концептуально цілісної та технологічно узгодженої програмної архітектури онтологічно-керованої системи управління, яка здатна інтегрувати формалізовані знання предметної області, динамічні моделі розподілу навантаження, алгоритми адаптивного прийняття рішень і механізми взаємодії з існуючими інформаційними інфраструктурами. Розв'язання цієї проблеми потребує формування нових підходів до поєднання семантичних і математичних моделей, розроблення ефективних методів їх програмної реалізації та забезпечення масштабованості й продуктивності системи в умовах реального функціонування.

Однією з ключових проблем організації каналів зв'язку між термінальними серверами в корпоративних університетських мережах є забезпечення надійної ідентифікації та автентифікації замкненої групи авторизованих користувачів (рис. 1). У сучасних інфраструктурах ідентифікація та автентифікація користувачів під час встановлення мережеских каналів зв'язку переважно реалізуються за допомогою механізмів на основі Kerberos, побудованих на клієнт-серверній парадигмі. Відповідно до цього підходу, термінальні мережі поділяються на області Kerberos (realm), кожна з яких містить спеціалізований сервер автентифікації, відповідальний за надання авторизованим користувачам контрольованого доступу до дозволених інформаційних ресурсів [27, 71, 72, 116].

Термінальні сервери взаємодіють між собою через парні канали зв'язку, формуючи розподілену систему автентифікації. Ці сервери використовують

спільний секретний ключ та обмінюються автентифікаційною інформацією, необхідною для перевірки ідентичності користувачів у системі. Кожен сервер підтримує локалізовану базу даних, у якій зберігаються облікові дані та атрибути авторизації легітимних користувачів.

Слід підкреслити, що, незважаючи на зростаючий інтерес до термінально-орієнтованих мережевих технологій, низка питань, пов'язаних з їх практичною організацією, масштабованістю та надійністю, залишається недостатньо дослідженою. Однією з найбільш суттєвих проблем у цій сфері є розроблення ефективних методів і алгоритмів, спрямованих на підвищення живучості каналів зв'язку. Такі механізми мають гарантувати своєчасний доступ авторизованих користувачів до необхідних ресурсів навіть за умов різних сценаріїв відмов, зокрема часткового виходу серверів з ладу, деградації каналів зв'язку або раптових пікових навантажень.

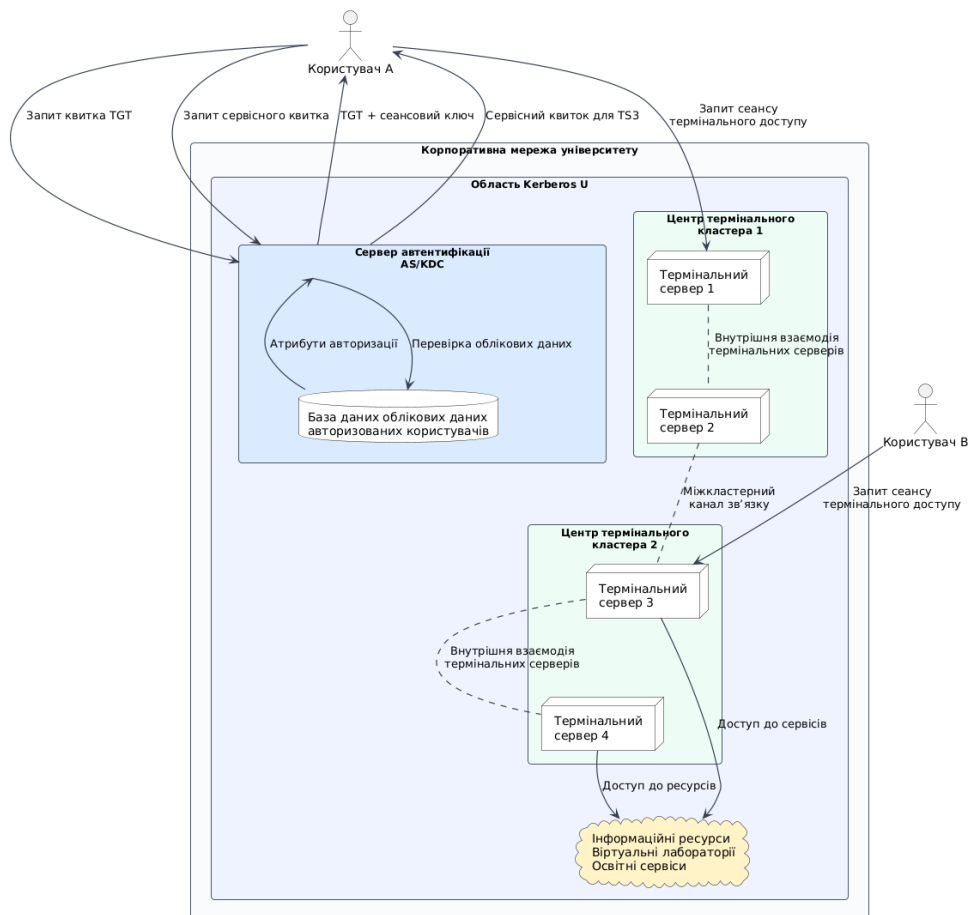


Рис. 3.8. Архітектура корпоративної мережі університету з центрами термінальних кластерів та механізмом автентифікації на основі Kerberos

Для розв'язання зазначених проблем моделювання динамічного розподілу навантаження між центрами термінальних кластерів набуває статусу ключового наукового напрямку. На початковому етапі такого моделювання структура комп'ютерної мережі формалізується у вигляді відкритої стохастичної мережі, що забезпечує аналітичний опис імовірнісних взаємодій, коливань навантаження та переходів станів, які виникають у розподілених середовищах автентифікації [12].

Нехай множина термінальних серверів, сформована в комп'ютерній мережі, позначається як $St = \{St_1, St_2, \dots, St_n\}$, і кожному серверу St_i поставлено у відповідність певну групу авторизованих користувачів із множини Us . Інакше кажучи, для кожного сервера St_i існує підмножина $Us_i \subset Us$ така, що сімейство множин $\{Us_i\}_{i=1}^n$ утворює розбиття множини Us , тобто:

$$Us_i \cap Us_j = \emptyset, i \neq j. \quad (3.3)$$

Кожна підмножина Us_i складається з окремих користувачів usr_{ki} . Отже, можна записати:

$$Us = \bigcup_{i=1}^n Us_i = \bigcup_{i=1}^n [Us_i] = sn_i, \sum_{i=1}^n sn_i = u, u > n, \quad (3.4)$$

де u — загальна кількість авторизованих користувачів, а sn — кількість термінальних серверів.

Очевидно, що всі сервери з множини St виконують однотипні функції автентифікації, оскільки реалізують єдину політику безпеки термінальної мережі. Як критерій ефективності функціонування каналів зв'язку під час автентифікації розглядається величина Tim_0 , що визначає середній час обслуговування запитів користувачів на автентифікацію за умови повної працездатності всіх серверів із множини St .

Припустимо, що термінальна мережа подається у вигляді експоненціальної відкритої стохастичної мережі, яка складається зі скінченної кількості

одноканальних систем масового обслуговування. Такі системи формують вузли обслуговування, що характеризуються сталою інтенсивністю надходження заявок λ_0 , яка не залежить від поточного стану мережі на виході джерела заявок St_0 .

Нехай інтенсивність λ_0 є відомою та розглядається як параметр мережі. Заявки від джерела St_0 надходять до мережі зі сталою ймовірністю p_{0i} маршрутизації до системи масового обслуговування (QS) St_i . Після обслуговування в системі QS St_i заявки з ймовірністю P_{ij} передаються до системи QS St_j , $j = 1, \dots, sn$, або залишають мережу при $j = 0$, тобто повертаються до джерела заявок. Очевидно, для кожного вузла має виконуватися умова нормування: сума ймовірностей маршрутизації з вузла St_i до всіх можливих напрямів $j = 0, 1, \dots, sn$ дорівнює одиниці.

Розглянемо перетворення вхідного потоку заявок з інтенсивністю λ_0 у вхідні потоки окремих систем масового обслуговування мережі в усталеному режимі. Позначимо через α_i коефіцієнт передачі, або трансформації, вхідного потоку до входу системи QS St_i . Кількісно цей коефіцієнт дорівнює середній кількості появ довільної заявки з вхідного потоку мережі у вхідному потоці системи QS St_i . Тоді інтенсивність вхідного потоку до системи QS St_i може бути виражена через λ_0 таким чином:

$$\lambda_i = \alpha_i \lambda_0 . \quad (3.5)$$

З іншого боку, за означенням, частка клієнтів із підмножини US_i у загальній інтенсивності λ_0 може бути подана через індивідуальні інтенсивності заявок l_{kii} користувача us_{ki} , спрямованих до сервера, тобто системи масового обслуговування QS St_i . Поширюючи це співвідношення на всі підмножини US_i , $i = 1$, отримуємо систему залежностей, що пов'язує глобальну вхідну інтенсивність λ_0 з інтенсивностями λ_i усіх систем масового обслуговування мережі.

Оскільки розглядається мережа без втрат, вихідні інтенсивності потоків із систем QSs S_i , $i = 1, \dots, sn$, збігаються з інтенсивностями їхніх вхідних потоків.

Інтенсивність потоку, що надходить на вхід системи $Q S s St_j$ $j = 1, \dots, sn$, дорівнює сумі частки потоку, яка надходить безпосередньо від джерела заявок, і часток потоків, перенаправлених з інших систем масового обслуговування відповідно до заданих імовірностей маршрутизації.

З урахуванням наведених співвідношень і рівності $\lambda_i = \alpha_i \lambda_0$ відповідне балансне рівняння перетворюється на систему лінійних неоднорідних алгебраїчних рівнянь відносно коефіцієнтів передачі α_i , $i = 1, \dots, sn$, яка має єдиний розв'язок:

$$\alpha_i = \frac{\sum_{kj=1}^{sn} \lambda_{kii}}{\sum_{j=1}^{sn} \sum_{k=j=1}^{sn} \lambda_{kii}} + \sum_{i=1}^{sn} \alpha_i \frac{\lambda_{ii}}{\lambda_i}, j = \overline{1, sn} . \quad (3.6)$$

На основі цього розв'язку можна визначити середній час обслуговування клієнтських запитів:

$$Tum_0 = \sum_{i=1}^{sn} \alpha_i t_i , \quad (3.7)$$

де $t_i \frac{1}{\mu_i - \lambda_i}$ — середній час обслуговування заявок у системі масового обслуговування St_i . Відповідно до постановки задачі, інтенсивності обслуговування є однаковими для всіх систем, тобто $\mu_i = \mu$.

Далі сформулюємо оптимізаційну модель, яка становить основу методу підвищення відмовостійкості каналів зв'язку під час автентифікації авторизованих користувачів.

Нехай стани всіх серверів із множини St задаються вектором стану:

$$x(k) = \langle x_1(k), \dots, x_i(k), \dots, x_{sn} sn(k) \rangle, \quad (3.8)$$

де $x_i(k) = 0$, якщо сервер St_i є працездатним, і $x_i(k) = 1$ — у протилежному випадку. Відомо, що загальна кількість таких векторів стану дорівнює 2^{sn} . Серед цих векторів не розглядаються стан $\langle 0, 0, \dots, 0 \rangle$, коли всі сервери є працездатними, і

стан $\langle 1, 1, \dots, 1 \rangle$, коли всі сервери вийшли з ладу. Отже, аналізуються лише нетривіальні стани $x(k)$, $k = 1, \dots, SN$, де $SN = 2^{sn} - 2$.

Сутність задачі полягає в тому, що адміністратор безпеки каналів зв'язку у разі відмови частини серверів, що відповідає певному стану $x(k)$, здійснює перерозподіл їхніх користувачів між працездатними серверами з урахуванням заданих обмежень. Зокрема, оскільки такі сервери можуть бути географічно розподіленими, перенаправлення користувачів між ними може супроводжуватися додатковими витратами.

Нехай вартість перемикання користувача u_{ki} із відмовленого сервера St_i на працездатний сервер St_j у стані $x(k)$ позначається через $C_{ij}^{(k)}$, $i, j = 1, \dots, sn$. Крім того, нехай максимальний обсяг пам'яті мережевого обладнання, на якому реалізовано сервери St_i , задається величиною V_i^{max} , $i = 1, \dots, sn$. Фактичний обсяг пам'яті, необхідний для розміщення сервера St_j , позначимо через V_j , $j = 1, \dots, sn$.

Для опису перерозподілу користувачів відмовлених серверів St_i за заданого вектора стану $x(k)$ між працездатними серверами St_j введемо псевдобулеву змінну $x_{ij}(k)$. При цьому $x_{ij}(k) = 1$, якщо користувачі з множини Us_i відмовленого сервера St_i перемикаються для автентифікації на сервер St_j у стані $x(k)$, і $x_{ij}(k) = 0$ — в іншому випадку. Зауважимо, що для кожного стану $x(k)$ усі користувачі з множини Us_i підключаються рівно до одного працездатного сервера. На основі наведених міркувань, а також формул (3.4) і (3.5), отримуємо таку оптимізаційну модель:

$$Tum_k = \sum_{j=1}^{sn} \alpha_i^{(k)t_j(k)} \rightarrow \min k = \overline{1, SN}, \quad (3.9)$$

Застосовуючи вирази для α_i та t_i до станів $x(k)$, отримуємо відповідні співвідношення для коефіцієнтів передачі, часу обслуговування та системи обмежень:

$$\alpha_i(k) = \frac{\sum_{k_j=1}^{sni} \lambda_{k_j i} + \sum_{i=1}^{sn} \sum_{k_i=1}^{sni} \lambda_{k_i X_i^{(k)}} X_j^{(k)}}{\sum_{j=1}^n \sum_{k_j=1}^{nj} \lambda_{k_j} + \sum_{i=1}^{sn} \alpha_i^{(k)} \frac{\lambda_{ij}}{\lambda_i} (1 - x_i^{(k)} + x_j^{(k)})} + (1 - x_i^{(k)}) \quad (3.10)$$

$$j = \overline{1, sn}, k = \overline{1, SN} \quad (3.11)$$

Середній час обслуговування для відповідних станів визначається як:

$$t_j^{(k)} = \frac{(1 - x_i^{(k)})}{\mu \left\{ \sum_{k_j=1}^{sni} \lambda_{k_j} + \sum_{i=1}^{sn} \left[\sum_{k_i=1}^{sni} \lambda_{k_i} x_{ij}^{(k)} + \lambda_{ij} (1 - x_i^{(k)}) \right] \right\}} \quad (3.12)$$

$$I = \overline{1, sn}, j = \overline{1, sn}, k = \overline{1, SN} \quad (3.13)$$

$$\sum_{i=1}^{sn} x_{ij}^{(k)} x_i^{(k)} = sn - 1, j = \overline{1, sn}, k = \overline{1, SN}$$

$$\sum_{i=1}^{sn} x_{ij}^{(k)} x_i^{(k)} = 1, i = \overline{1, sn}, k = \overline{1, SN}$$

$$\sum_{i=1}^{sn} \sum_{k=1}^{sni} \sum_{j=1}^{sn} c_{kj} x_{ij}^{(k)} x_i^{(k)} \leq C, k = \overline{1, SN} \quad (3.14)$$

$$\sum_{i=1}^{sn} V_i x_{ij}^{(k)} x_i^{(k)} \leq V_j^{\max} - V_j, j = \overline{1, sn}, k = \overline{1, SN}$$

$$\min[\mu/\alpha_j^{(k)}] \sum_{i=1}^n \sum_{k_i=1}^{sni} \lambda_{k_i}, j = \overline{1, sn}, k = \overline{1, SN}$$

Обмеження (3.14) означають, що під час перерозподілу користувачів відмовлених серверів St_i між працездатними серверами St_j за вектора стану $x(k)$ сумарна вартість перемикання не повинна перевищувати наперед заданий поріг C . Крім того, нерівність (12) встановлює верхню межу для інтенсивності вхідного потоку λ_0 за умови існування усталеного режиму в експоненціальній відкритій стохастичній мережі.

Із формул (3.10)–(3.14) випливає, що алгоритм розв'язання сформульованої

оптимізаційної задачі належить до класу задач дискретного програмування з псевдобулевыми змінними. Перед розробленням практичного алгоритму, придатного для використання в реальних умовах проектування та експлуатації термінальних мереж, доцільно оцінити обчислювальну складність цієї моделі за умови повного перебору всіх можливих варіантів.

Відомо, що m відмовлених термінальних серверів можна вибрати з n серверів C_n^m способами. У цьому випадку $n - m$ серверів залишаються працездатними. Відповідно до умов моделі (3.10)–(3.14), користувачі кожного відмовленого сервера мають бути перенаправлені на один із $(n - m)$ працездатних термінальних серверів. Отже, для заданої кількості m відмовлених серверів загальна кількість можливих варіантів їх перерозподілу між $u(n - m)$ працездатними серверами дорівнює:

$$\theta(n, m) = C_n^m(n - m) \cdot (n - m) \cdots (n - m) = C_n^m(n, m)^m. \quad (3.15)$$

Поширюючи цю формулу на всі значення m , де $1 \leq m \leq n - 1$, отримуємо обчислювальну складність моделі (3.11)–(3.14):

$$u(n) = \sum_{m=1}^{n-1} C_n^m(n - m)^m. \quad (3.16)$$

Очевидно, що розв'язання цієї задачі методом повного перебору є практично нездійсненним. Тому для задач такого типу необхідно забезпечити ефективний частковий перебір порівняно невеликої підмножини допустимих варіантів із неявним відсіканням решти. Ця мета досягається за допомогою алгоритму, що відповідає моделі (3.11)–(3.14), ґрунтується на методі гілок і меж та враховує специфічну структуру досліджуваної задачі.

Введемо такі позначення:

$$I_k = \{i \mid x_i^{(k)} = 1\} J_k = \{i \mid x_i^{(k)} = 0\} \quad (3.17)$$

де $N = \{1, 2, \dots, n\}$. Очевидно, що $I_k = J_k = \frac{N}{I_k}$, тобто відмовлені сервери формують множину I_k , а працездатні сервери — комплементарну множину J_k .

Дерево будується таким чином. Підмножина першого рівня формується шляхом фіксації призначення першого сервера з множини I_k до різних серверів із множини J_k : $X_{j_1}, X_{j_2}, X_{j_l} | J_k$. Кожна множина X_{j_1} містить усі варіанти, у яких перший відмовлений сервер із множини J_k призначається серверу $j_1 \in J_k$, тоді як призначення решти відмовлених серверів залишаються довільними.

Аналогічно, підмножина другого рівня формується шляхом фіксації призначення другого відмовленого сервера з множини J_k до різних серверів із множини J_k . Множина X_{j_1, j_2} містить усі варіанти, у яких перший відмовлений сервер призначено серверу $j_1 \in J_k$, другий відмовлений сервер — серверу $j_2 \in J_k$, а призначення решти серверів із I_k залишаються довільними. Такий процес продовжується до повної фіксації всіх необхідних призначень. Для кожної підмножини, тобто для кожного вузла дерева гілкування, необхідно побудувати межі цільової функції (12) та відповідних обмежень. Загальний вираз оцінки цільової функції для підмножини варіантів X_{j_1, j_2, \dots, j_l} у цій задачі можна записати як $V(X_{j_1, j_2, \dots, j_l})$. Тут $V(X_{j_1, j_2, \dots, j_l})$ позначає оцінку цільової функції для всіх варіантів у межах відповідної підмножини, коли перші l параметрів рішення зафіксовані як j_1, j_2, \dots, j_l , а для решти параметрів $i = l + 1, l + 2, \dots, |I_k|$ конкретне призначення ще не вибрано. Така оцінка вважається допустимою лише за умови виконання таких умов здійсненності обмежень:

$$\sum_{j=1}^{jk} \sum_{m=1}^1 C_{mj} X_{mj} + \min \sum_{m=1+1}^{Ik} \sum_{j=1}^{jk} X_m^{(k)} X_{mj}^{(k)} C_{mj} \leq C$$

$$\sum_{j=1}^{jk} \sum_{m=1}^1 V_j + \min \sum_{m=1+1}^{Ik} \sum_{j=1}^{jk} V_i X_m^{(k)} X_{mj}^{(k)} < V_j \max - V_j, j = \overline{1, J_k}, \quad (3.18)$$

$$\min \frac{\mu}{\alpha_i^{(k)}} > \lambda_0$$

На основі наведених формул алгоритм розв'язання оптимізаційної задачі

формується таким чином (рисунок 3.9).

Крок 0. Ініціалізація.

Крок 1. Введення вхідних даних. Задаються початкові параметри: кількість серверів n ; вартості перемикавання користувачів із сервера St_i на сервер St_j , C_{ij} $i, j = 1, \dots, n$; максимальні обсяги пам'яті обладнання, на якому розміщені сервери, V_j^{max} $j = 1, \dots, n$; фактичні обсяги пам'яті серверів V_j $j = 1, \dots, n$; імовірності передавання заявок із сервера St_i до сервера St_j , P_{ij} $i, j = 1, \dots, n$ Також обчислюються значення P_{0i} $i, j = 1, \dots, n$.

Крок 2. Генерація наступного вектора стану X_k . Визначаються множини працездатних і відмовлених серверів: I_0 — множина працездатних серверів, I_1 — множина відмовлених серверів.

Крок 3. Вибір наступного непризначеного відмовленого сервера з множини I_1 .

Крок 4. Обчислення залишкової інтенсивності обслуговування. Для кожного сервера з множини I_0 обчислюється його залишкова інтенсивність обслуговування, яка визначається як різниця між інтенсивністю обслуговування μ та сумою інтенсивностей заявок від сервера, що розглядається, і всіх відмовлених серверів, уже призначених цьому працездатному серверу. Якщо інтенсивність заявок вибраного відмовленого сервера з I_1 перевищує всі залишкові інтенсивності обслуговування серверів із I_0 , алгоритм переходить до кроку 10. В іншому випадку вибирається перший сервер із I_0 , залишкова інтенсивність якого перевищує інтенсивність заявок відмовленого сервера. Пара (i, j) , де $i \in I_1$ — відмовлений сервер, а $j \in I_0$ — вибраний працездатний сервер, заноситься до списку призначень.

Крок 5. Перевірка обмежень. Перевіряється виконання обмежень (3.13) і (3.14) для поточного часткового призначення. Якщо хоча б одне з обмежень порушується, алгоритм переходить до кроку 10; в іншому випадку виконання продовжується.

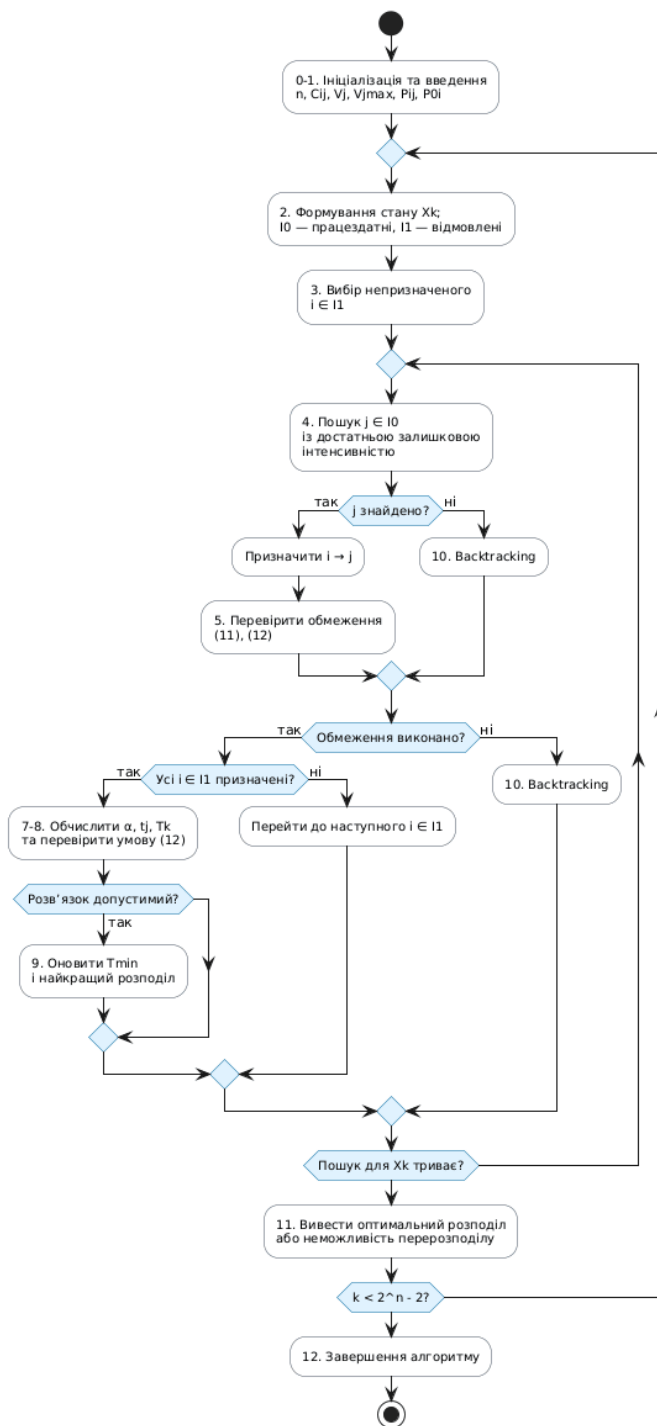


Рис. 3.9. Алгоритм перерозподілу користувачів відмовлених серверів.

Крок 6. Перевірка завершеності призначення. Перевіряється, чи є поточний відмовлений сервер останнім елементом множини I_1 . Якщо так, алгоритм переходить до наступного кроку; інакше — до кроку 10.

Крок 7. Розв'язання системи рівнянь. Система (3.13) розв'язується методом простих ітерацій для отримання значень $\alpha_{i,j} = 1, \dots, n$. Далі обчислюються

значення $t_j, j = 1, \dots, n$, і значення T_k відповідно до формул (3.9) і (3.10). Оскільки знаменник у формулі (3.14) задовольняє умову збіжності, ітераційна процедура є збіжною.

Крок 8. Перевірка умови (3.14). Якщо умова (3.14) не виконується, алгоритм переходить до кроку 10; в іншому випадку виконання продовжується.

Крок 9. Оновлення поточного найкращого розв'язку. Якщо раніше вже було обчислено значення T_{min} , воно порівнюється з новоотриманим значенням T_k . Якщо $T_{min} < T_k$, алгоритм переходить до наступного кроку. В іншому випадку, або якщо значення T_{min} ще не було задане, встановлюється $T_{min} = T_k$, а відповідне призначення серверів зберігається як поточний найкращий розв'язок.

Крок 10. Повернення назад. Перевіряється можливість повернення до попереднього кроку пошуку. Із списку призначень вибирається остання сформована пара. Якщо список порожній, повернення неможливе, і алгоритм переходить до кроку 11. В іншому випадку здійснюється спроба знайти інший працездатний сервер, на який може бути перенаправлений відповідний відмовлений сервер. Якщо такий сервер знайдено, призначення оновлюється, і алгоритм повертається до кроку 4. Якщо такого сервера немає, відповідний відмовлений сервер вилучається зі списку призначень, після чого крок 10 повторюється.

Крок 11. Виведення результатів для поточного стану. Якщо значення T_{min} було отримано, виводиться відповідний оптимальний розподіл відмовлених серверів між працездатними серверами. Якщо для заданого вектора стану X_k такого значення не існує, виводиться повідомлення про неможливість перерозподілу. Якщо кількість оброблених станів менша за $2^n - 2$, індекс стану збільшується, і алгоритм повертається до кроку 2; в іншому випадку він переходить до кроку 12.

Крок 12. Завершення роботи алгоритму. Обчислювальна складність $\theta_\alpha(n)$ запропонованого алгоритму є суттєво нижчою за складність $\theta(n)$ методу повного перебору. Зі збільшенням розмірності задачі (кількості серверів) запропонований

алгоритм демонструє не лише нижчу складність, але й експоненційне зростання відносної ефективності, що підтверджує його придатність для застосування у великомасштабних термінальних мережах, рисунок 3.10.

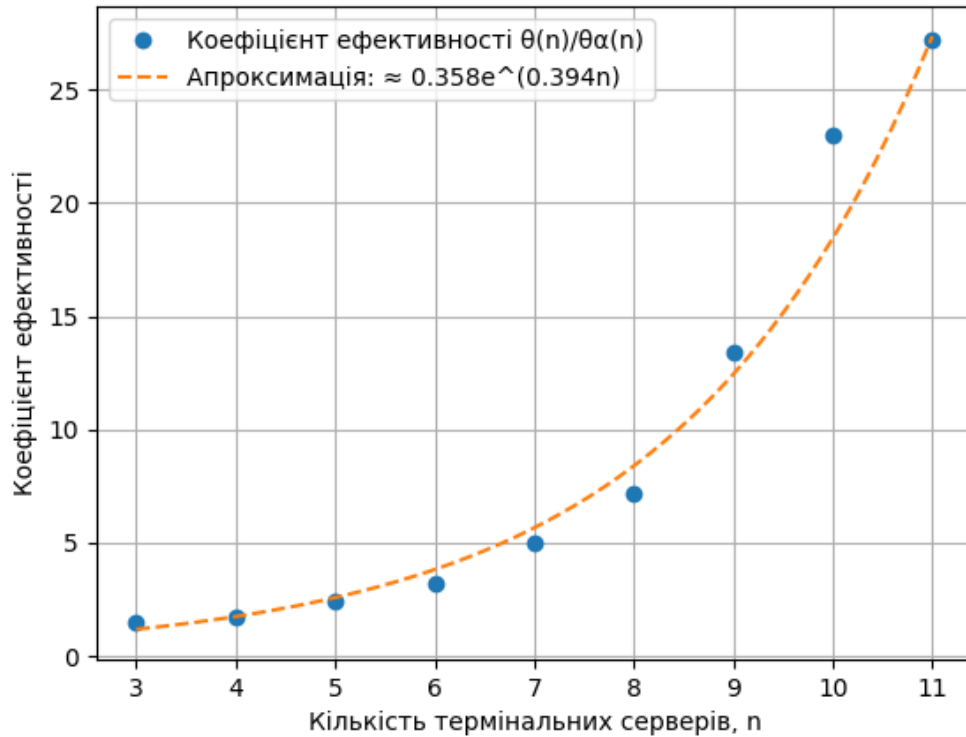


Рис. 3.10. Експоненційне зростання коефіцієнта ефективності $\theta(n)/\theta_\alpha(n)$ залежно від кількості термінальних серверів з апроксимаційною кривою.

Розроблена динамічну модель розподілу навантаження для центрів термінальних кластерів, яка враховує часову змінність потоків запитів, гетерогенність ресурсів та багатокритеріальний характер задачі оптимізації. Схемупобудови та імплементації цієї моделі представлено на рисунку 3.11.

На відміну від класичних підходів, запропонована модель інтегрована з онтологічною базою знань і доповнена механізмами логічного виведення, що забезпечує врахування контексту обробки запитів, пріоритетів, обмежень і структурних залежностей між елементами системи.

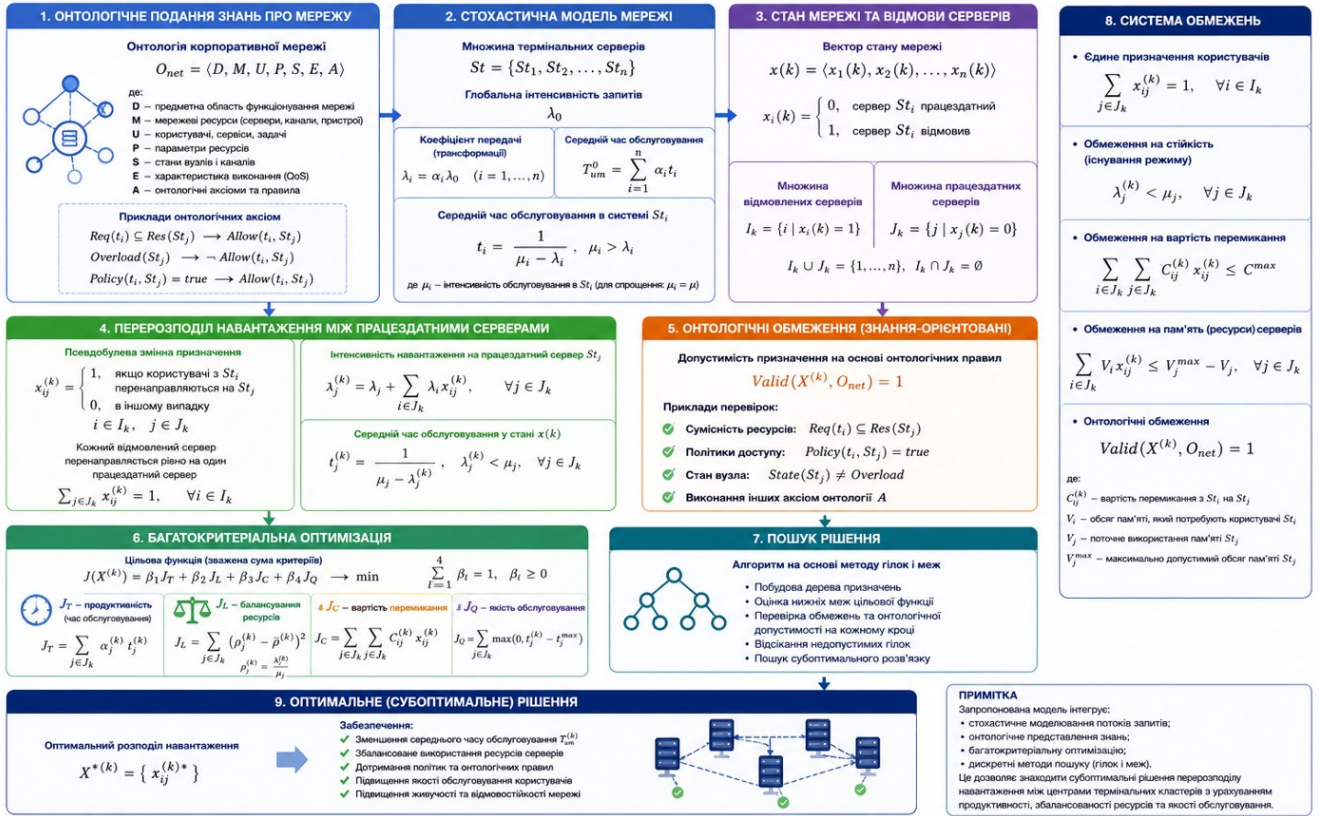


Рис. 3.11. Математична модель розподілу навантаження між центрами термінальних кластерів на основі онтологічного подання знань та багатокритеріальної оптимізації

Висновки до розділу 3

1. Розроблено архітектуру знання-орієнтованої системи управління корпоративними комп'ютерними мережами, яка забезпечує інтеграцію онтологічного представлення знань про структуру мережі, її ресурси, сервіси та множину задач із механізмами логічного виведення, підтримки прийняття рішень та багатокритеріальної оптимізації розподілу навантаження.

2. Запропоновано багаторівневу архітектуру програмної системи, яка включає підсистеми моніторингу мережевої інфраструктури, управління онтологічною базою знань, логічного виведення, конфігурування, оптимізації та підтримки прийняття рішень. На відміну від традиційних систем мережевого адміністрування, розроблена архітектура дозволяє використовувати формалізовані

знання про ресурси, сервіси, користувачів та задачі для автоматизованого формування конфігураційних рішень.

3. Реалізовано онтологічну базу знань, яка забезпечує централізоване представлення знань про корпоративну комп'ютерну мережу на основі концептів, відношень та аксіом предметної області. Це створило можливість інтеграції інформації про мережеві ресурси, сервіси, політики безпеки та множину задач у межах єдиного семантичного простору.

4. Розроблено програмну реалізацію механізму логічного виведення, який забезпечує автоматичний аналіз фактів, що надходять від підсистеми моніторингу, перевірку онтологічних аксіом та формування нових знань про поточний стан мережевого середовища. Використання механізму логічного виведення дозволило автоматизувати процес побудови множини допустимих конфігурацій мережі та забезпечити підтримку прийняття рішень на основі формалізованих знань.

5. Реалізовано інтеграцію онтологічної бази знань із математичною моделлю розподілу навантаження між центрами термінальних кластерів. Це дозволило використовувати результати логічного виведення для формування множини допустимих альтернативних конфігурацій та виконувати багатокритеріальну оптимізацію лише серед рішень, які відповідають структурі мережі, політикам безпеки та вимогам задач, що розв'язуються в корпоративному середовищі.

6. Обґрунтовано вибір технологічних засобів реалізації програмної системи, зокрема використання платформи Java та фреймворку Spring Boot для серверної частини, HTML5, CSS3, JavaScript і Bootstrap для реалізації користувацького інтерфейсу, PostgreSQL для зберігання даних та OWL для представлення онтологічної бази знань. Запропонований набір технологій забезпечує масштабованість, модульність та можливість подальшого розвитку системи.

РОЗДІЛ 4

ЗНАННЯ-ОРІЄНТОВАНА ПРОГРАМНА СИСТЕМА ДЛЯ УПРАВЛІННЯ КОРПОРАТИВНИМИ КОМП'ЮТЕРНИМИ МЕРЕЖАМИ

У розділі описано питання програмної реалізації розробленої знання-орієнтованої системи управління корпоративними комп'ютерними мережами. Основну увагу приділено організації користувацького інтерфейсу, реалізації основних функціональних модулів, інтеграції онтологічної бази знань із механізмами логічного виведення та оптимізації, а також забезпеченню взаємодії між компонентами програмного комплексу.

Крім того, у розділі наведено результати експериментальних досліджень запропонованого методу онтологічно-керованого конфігурування та удосконаленої математичної моделі розподілу навантаження між центрами термінальних кластерів. Дослідження виконано на прикладі реальних корпоративних комп'ютерних мереж, що дозволило оцінити ефективність розроблених моделей і методів за показниками продуктивності, збалансованості використання ресурсів, якості обслуговування користувачів та рівня інформаційної безпеки.

У кінці розділу здійснено оцінку ефективності розробленої знання-орієнтованої програмної системи управління корпоративними комп'ютерними мережами та експериментальну перевірку запропонованого методу онтологічно-керованого конфігурування і математичної моделі багатокритеріального розподілу навантаження між центрами термінальних кластерів.

Основні результати даного розділу опубліковані в наступних наукових працях [65, 71, 72, 88, 94, 116].

4.1 Організація користувацького інтерфейсу програмної системи управління корпоративними комп'ютерними мережами

У попередніх розділах дисертаційної роботи було розроблено онтологічну модель корпоративної комп'ютерної мережі, метод онтологічно-керованого конфігурування мережевих ресурсів, математичну модель розподілу навантаження між центрами термінальних кластерів та архітектуру знання-орієнтованої програмної системи управління корпоративними комп'ютерними мережами. Реалізація запропонованих моделей і методів потребує створення ефективного користувацького інтерфейсу, який забезпечує взаємодію користувача з усіма функціональними компонентами системи, а також надає засоби візуалізації результатів логічного виведення, процесів конфігурування та оптимізації мережевої інфраструктури.

Особливістю розробленої програмної системи є інтеграція традиційних функцій моніторингу та адміністрування корпоративної мережі з механізмами семантичного аналізу, онтологічного представлення знань і підтримки прийняття рішень. Тому користувацький інтерфейс повинен не лише відображати поточний стан мережевої інфраструктури, але й забезпечувати доступ до знань, накопичених в онтологічній базі, результатів роботи механізмів логічного виведення та рекомендацій щодо конфігурування мережевих ресурсів.

При проектуванні інтерфейсу використано концепцію інформаційної панелі управління (Dashboard-Oriented Architecture), яка широко застосовується у сучасних мережевих системах управління корпоративного рівня, зокрема в програмних продуктах Cisco DNA Center, SolarWinds Network Performance Monitor, ManageEngine OpManager та PRTG Network Monitor [25]. Застосування такого підходу дозволяє забезпечити централізований доступ до всієї інформації про мережу та мінімізувати часові витрати адміністратора на пошук необхідних даних.

Функціональна структура інтерфейсу побудована відповідно до архітектури програмної системи, розробленої у розділі 3, та включає набір взаємопов'язаних

модулів, кожен з яких реалізує окремий аспект управління корпоративною мережею. Усі модулі інтегровані через єдиний механізм автентифікації та використовують спільну онтологічну базу знань, що забезпечує узгодженість інформації між підсистемами.

Головна сторінка системи представлена на рисунку 4.1. Центральним елементом користувацького інтерфейсу є головна сторінка системи, яка виконує роль інтегрованої інформаційної панелі управління мережею. На цій сторінці відображаються агреговані показники функціонування корпоративної мережі, включаючи кількість активних вузлів, стан серверної інфраструктури, рівень завантаження мережевих ресурсів, кількість активних сервісів та задач, що розв'язуються в мережевому середовищі.

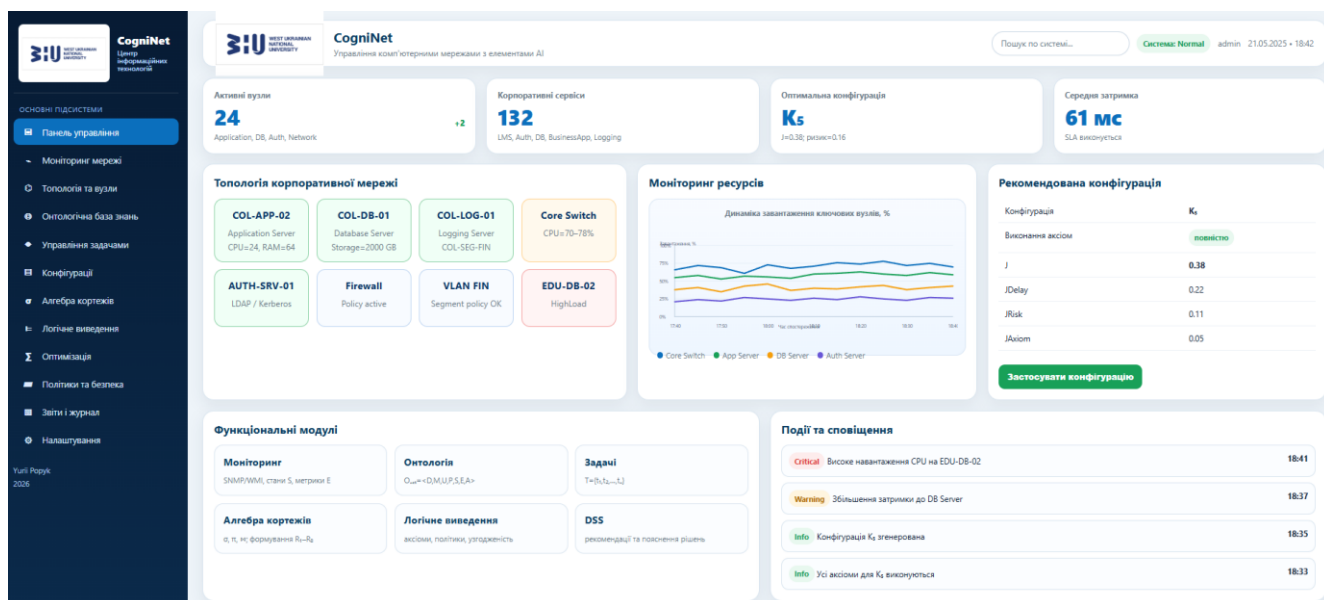


Рис. 4.1. Головна сторінка системи управління корпоративною комп'ютерною мережею

Особливістю головної сторінки є інтеграція результатів роботи всіх функціональних модулів системи. Тут відображаються повідомлення про критичні події, рекомендації щодо оптимізації конфігурації мережі, результати останнього логічного виведення та показники ефективності функціонування мережевої інфраструктури. Головна сторінка формує цілісне уявлення про поточний стан

корпоративної мережі та забезпечує оперативну підтримку процесів прийняття рішень.

Важливим компонентом користувацького інтерфейсу розробленої знання-орієнтованої програмної системи є сторінка топології та вузлів мережі (рисунк 4.2.), яка забезпечує графічне представлення структури корпоративної комп'ютерної мережі та відображення взаємозв'язків між її елементами. Необхідність реалізації такого модуля обумовлена тим, що розроблена онтологічна модель корпоративної мережі містить значну кількість сутностей та відношень між ними, аналіз яких у табличній формі є малоефективним для адміністратора мережі.

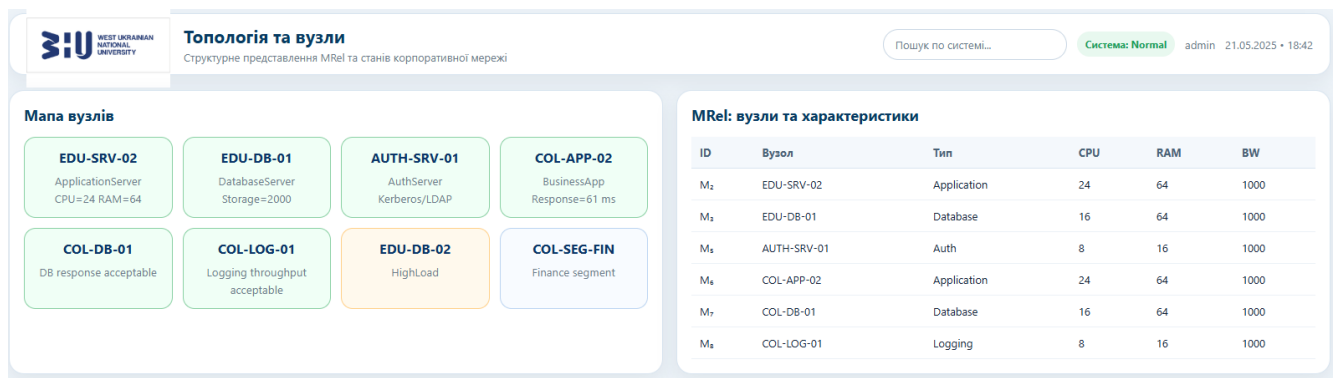


Рис. 4.2. Сторінка топології та вузлів мережі

На відміну від традиційних систем моніторингу, де мережеві вузли відображаються як ізольовані об'єкти, запропонована сторінка реалізує інтегроване подання топологічної та семантичної інформації. Кожний вузол мережі розглядається не лише як фізичний або логічний ресурс, а як елемент онтологічної моделі, який має множину властивостей, взаємозв'язків та ролей у процесах розв'язання задач у мережевому середовищі.

Основною функцією сторінки є побудова інтерактивної карти корпоративної мережі, на якій відображаються сервери, маршрутизатори, комутатори, центри термінальних кластерів, бази даних, сервіси, віртуальні машини та інші компоненти мережевої інфраструктури. Візуалізація виконується у вигляді графа, вершинами якого є мережеві вузли, а ребрами – фізичні або логічні канали зв'язку

між ними.

Сторінка моніторингу мережі представлена на рисунку 4.3. Підсистема моніторингу реалізує функції безперервного збору та аналізу інформації про стан мережевої інфраструктури. Її розроблення безпосередньо пов'язане із запропонованою онтологічною моделлю корпоративної мережі, оскільки саме результати моніторингу використовуються для актуалізації знань про поточний стан ресурсів, сервісів та мережевих сегментів.

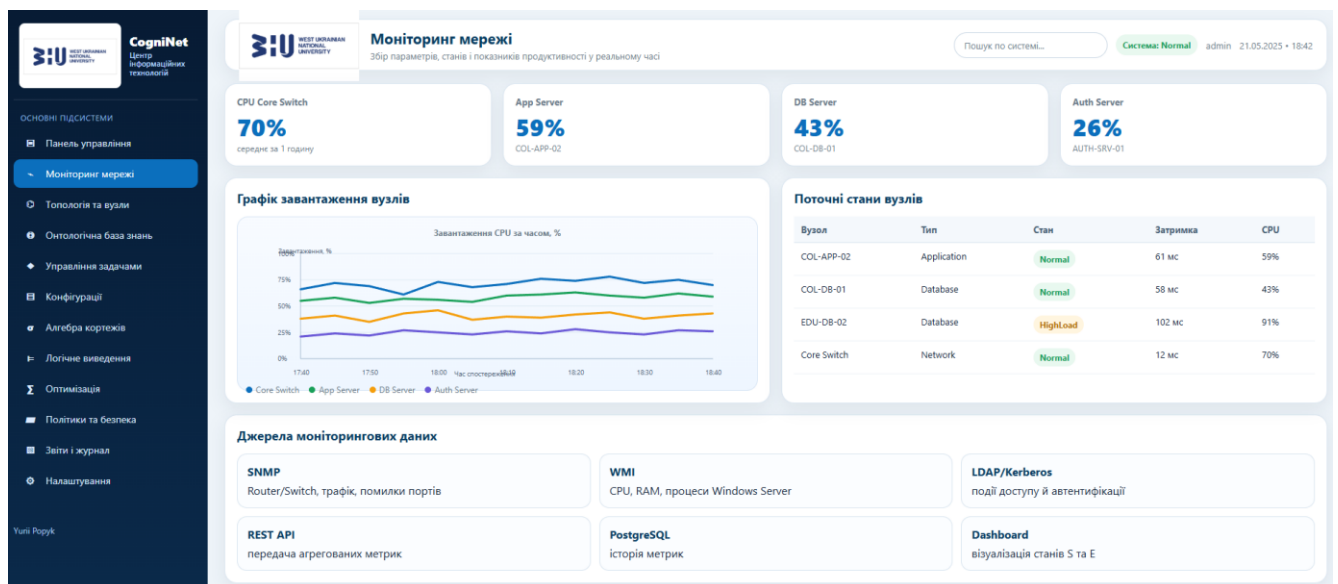


Рис. 4.3. Реалізація підсистеми моніторингу

Основна область сторінки містить набір інтерактивних графіків та діаграм, які відображають динаміку зміни параметрів функціонування мережевої інфраструктури. Зокрема, візуалізуються показники використання процесорних ресурсів серверів, оперативної пам'яті, мережевого трафіку, пропускної здатності каналів зв'язку та часу відгуку корпоративних сервісів. Відображення часових рядів забезпечує можливість аналізу тенденцій зміни навантаження та прогнозування потенційних перевантажень окремих компонентів мережі.

Особливе значення для реалізації запропонованого знання-орієнтованого підходу має інтеграція сторінки моніторингу з онтологічною базою знань. Кожний параметр, що надходить від мережевого обладнання або програмних сервісів,

автоматично пов'язується із відповідними сутностями онтології. У результаті формується актуальна семантична модель мережевого середовища, яка використовується підсистемою логічного виведення для формування рекомендацій щодо управління ресурсами.

Важливою функцією сторінки є виявлення аномальних ситуацій у роботі мережевої інфраструктури. Для цього здійснюється аналіз відхилень поточних значень параметрів від нормативних або очікуваних значень, визначених у базі знань. У разі виявлення перевантаження серверів, деградації каналів зв'язку або порушення політик безпеки система автоматично формує повідомлення та передає їх до підсистеми підтримки прийняття рішень.

Окремий блок інтерфейсу присвячений моніторингу центрів термінальних кластерів, що є особливо важливим у контексті розробленої математичної моделі розподілу навантаження. Для кожного кластерного центру відображаються показники поточного навантаження, кількість активних користувацьких сесій, інтенсивність вхідних запитів та рівень використання обчислювальних ресурсів. Отримані дані використовуються оптимізаційним модулем для формування рішень щодо перерозподілу навантаження між центрами термінальних кластерів. На сторінці також реалізовано механізм прогнозування стану мережі на основі накопичених статистичних даних. Використання історичних журналів та результатів моніторингу дозволяє оцінювати можливість виникнення критичних ситуацій та формувати превентивні рекомендації щодо зміни конфігурації мережевих ресурсів.

Сторінка онтологічної бази знань представлена на рисунку 4.4. Одним із ключових компонентів розробленої системи є підсистема онтологічного представлення знань. Її інтерфейс реалізовано у вигляді сторінки онтологічної бази знань, яка забезпечує візуалізацію структури онтології корпоративної мережі.



Рис. 4.4. Реалізація онтологічної підсистеми

Ключовою особливістю розробленої програмної системи, яка принципово відрізняє її від більшості існуючих засобів управління корпоративними комп'ютерними мережами, є використання онтологічного представлення знань про мережеву інфраструктуру, її ресурси, сервіси, користувачів та множину задач, що розв'язуються в мережевому середовищі. Саме тому центральне місце в структурі користувацького інтерфейсу займає сторінка онтологічної бази знань, яка забезпечує взаємодію користувача з розробленою онтологічною моделлю корпоративної мережі.

На відміну від традиційних баз даних, у яких інформація представлена у вигляді таблиць та записів, онтологічна база знань забезпечує зберігання не лише даних, але й знань про взаємозв'язки між об'єктами предметної області. Саме тому сторінка онтологічної бази знань виконує функцію інтелектуального центру

програмної системи, через який здійснюється управління структурою знань та підтримка процесів логічного виведення.

Основним елементом сторінки є інтерактивний онтологічний граф, який відображає структуру корпоративної мережі у вигляді множини взаємопов'язаних сутностей. Вершини графа відповідають конкретним об'єктам онтології, а ребра відображають семантичні відношення між ними. Завдяки цьому користувач отримує можливість аналізувати не лише структуру мережі, але й логіку її функціонування та взаємозв'язки між окремими компонентами.

Особливістю реалізованого інтерфейсу є можливість одночасного відображення декількох рівнів абстракції знань. На верхньому рівні користувач може аналізувати загальну структуру корпоративної мережі, тоді як на нижчих рівнях доступний перегляд властивостей конкретних ресурсів, сервісів або задач. Такий підхід відповідає принципам багаторівневого онтологічного моделювання та суттєво спрощує навігацію в базі знань.

Сторінка забезпечує підтримку основних операцій керування онтологією. Адміністратор системи може створювати нові класи та екземпляри сутностей, модифікувати існуючі зв'язки між об'єктами, додавати нові атрибути ресурсів і сервісів, а також визначати онтологічні аксіоми та правила логічного виведення. Таким чином забезпечується адаптація бази знань до змін структури корпоративної мережі та появи нових задач. Важливою функцією сторінки є візуалізація онтологічних аксіом, які були розроблені в попередніх розділах дисертації. Користувач може переглядати правила відповідності ресурсів задачам, обмеження доступу, правила допустимості конфігурацій, умови розподілу навантаження та інші логічні залежності, що використовуються механізмом підтримки прийняття рішень. Це дозволяє контролювати коректність знань та забезпечує прозорість роботи системи.

Особливе значення має інтеграція сторінки онтологічної бази знань із модулем логічного виведення. Будь-яка зміна в структурі онтології автоматично враховується під час формування конфігураційних рішень, оптимізації розподілу

навантаження та перевірки політик безпеки. Таким чином досягається узгодженість між декларативною частиною системи, яка представлена онтологією, та процедурною частиною, що реалізує механізми прийняття рішень.

Сторінка управління задачами представлена на рисунку 4.5. Розроблений у попередньому розділі метод онтологічно-керованого конфігурування базується на множині задач, що розв'язуються в мережевому середовищі. Саме тому окремим функціональним модулем реалізовано підсистему управління задачами.

Управління задачами
Формування множини задач T та параметричних вимог

Пошук по системі... Система: Normal admin 21.05.2025 - 18:42

Активні задачі

ID	Назва	Вимоги	Пріоритет	Стан
T ₁	Оптимізація LMS	CPU≥24 RAM≥64	Високий	готово
T ₂	Бізнес-аналітика	BusinessApp + DB + Log	Високий	готово
T ₃	Безпека сегментації	Policy FIN	Середній	очікує

Формалізація задачі

$T = \{t_1, t_2, \dots, t_n\}$
 $t_i = \langle id_i, u_i, p_i, c_i \rangle$

Кожна задача містить об'єкт використання, параметричні вимоги та характеристики виконання.

Рис. 4.5. Сторінка управління задачами

На сторінці кожна задача відображається як окремий об'єкт із набором атрибутів, серед яких особливе значення мають тип задачі, її пріоритетність, поточний стан, необхідні обчислювальні ресурси, задіяні сервіси та часові характеристики розв'язання. Такий підхід забезпечує формалізований зв'язок між прикладним рівнем функціонування мережі та інфраструктурними ресурсами, що використовуються для підтримки відповідних бізнес-процесів. Важливою особливістю сторінки є інтеграція з онтологічною базою знань. Під час створення нової задачі система автоматично аналізує структуру онтології та встановлює зв'язки між задачею, необхідними ресурсами, мережевими сервісами та політиками безпеки. У результаті формується семантичний опис задачі, який надалі використовується механізмом логічного виведення та підсистемою конфігурування.

На відміну від класичних систем управління ресурсами, запропонований модуль дозволяє аналізувати не лише технічні характеристики задач, але й їх

змістовні взаємозв'язки. Наприклад, система може автоматично визначати групи взаємопов'язаних задач, виявляти конкуренцію за ресурси або встановлювати залежності між окремими бізнес-процесами. Така інформація використовується під час формування конфігурацій мережевих ресурсів та реалізації процедур балансування навантаження.

Окремий блок сторінки призначений для контролю використання ресурсів задачами. Для кожної задачі відображаються показники використання процесорного часу, оперативної пам'яті, мережевого трафіку та інших ресурсів корпоративної мережі. Це дозволяє оцінювати ефективність використання інфраструктури та своєчасно виявляти потенційні конфлікти ресурсів.

Важливим елементом інтерфейсу є відображення пріоритетів задач, які використовуються розробленою математичною моделлю розподілу навантаження між центрами термінальних кластерів. Система автоматично враховує рівень важливості задач під час прийняття рішень щодо перерозподілу ресурсів та формування нових конфігурацій мережі. Завдяки цьому забезпечується першочергове обслуговування критично важливих задач навіть за умов високого навантаження або часткової відмови окремих компонентів мережевої інфраструктури.

Сторінка конфігурацій представлена на рисунку 4.6. Підсистема конфігурування реалізує практичне застосування запропонованого методу онтологічно-керованої конфігурації корпоративної мережі. На сторінці відображаються всі доступні конфігурації ресурсів, результати перевірки онтологічних аксіом та ступінь відповідності конфігурацій вимогам задач.

На відміну від традиційних засобів керування конфігураціями, запропонований інтерфейс не обмежується відображенням технічних параметрів мережевих вузлів. Основна увага приділяється аналізу відповідності конфігурацій вимогам задач, що розв'язуються в мережевому середовищі. Для цього використовуються результати роботи онтологічного механізму логічного виведення та розроблені аксіоми сумісності ресурсів. Однією з ключових функцій

сторінки є автоматизоване формування допустимих конфігурацій. Після вибору конкретної задачі система здійснює пошук відповідних ресурсів у базі знань, перевіряє структурну відповідність вузлів вимогам задачі, аналізує параметричні характеристики ресурсів та виконує перевірку множини онтологічних аксіом. У результаті формується набір конфігурацій, які потенційно можуть бути використані для розв’язання поставленої задачі.

Конфігурації
Побудова ConfigRel і вибір допустимих конфігурацій

Пошук по системі... Система: Normal admin 21.05.2025 • 18:42

Допустимі конфігурації для задачі бізнес-аналітики

Конфігурація	App Server	DB Server	Logging	Segment	Аксіоми	Затримка	Ризик	J
K_4	COL-APP-01	COL-DB-01	COL-LOG-01	COL-SEG-GEN	порушено	84	0.41	0.74
K_5	COL-APP-02	COL-DB-01	COL-APP-02	COL-SEG-FIN	повністю	61	0.16	0.38
K_6	COL-APP-02	COL-DB-02	COL-LOG-01	COL-SEG-FIN	повністю	58	0.19	0.41

Оптимальна конфігурація K_5

- COL-APP-02 (Application Server)
- COL-DB-01 (Database Server)
- COL-APP-02 (Logging)
- COL-SEG-FIN (Finance segment)

Інтегральний критерій

$$J(K) = \alpha_1 J_{\text{Delay}} + \alpha_2 J_{\text{Risk}} + \alpha_3 J_{\text{Axiom}}$$

$$K^* = K_5$$

Застосувати K_5

Рис. 4.6. Сторінка управління конфігураціями комп’ютерної мережі

Окремий функціональний блок сторінки забезпечує порівняння альтернативних конфігурацій. Для кожного варіанта відображаються показники затримки доступу до сервісів, рівень використання обчислювальних ресурсів, ризику виникнення перевантажень та інтегральні критерії якості функціонування мережі. Це дозволяє оцінювати переваги різних варіантів конфігурації ще до їх практичного застосування.

Сторінка оптимізації представлена на рисунку 4.7. Сторінка оптимізації забезпечує реалізацію розробленої математичної моделі багатокритеріального розподілу навантаження між центрами термінальних кластерів.

Сторінка оптимізації є інтерфейсною реалізацією розробленої математичної моделі розподілу навантаження між центрами термінальних кластерів та забезпечує інтеграцію результатів роботи підсистем моніторингу, онтологічного аналізу, конфігурування і логічного виведення. На основі даних, що надходять із

зазначених модулів, формується множина допустимих альтернатив, серед яких здійснюється пошук найбільш ефективного варіанта розподілу ресурсів.

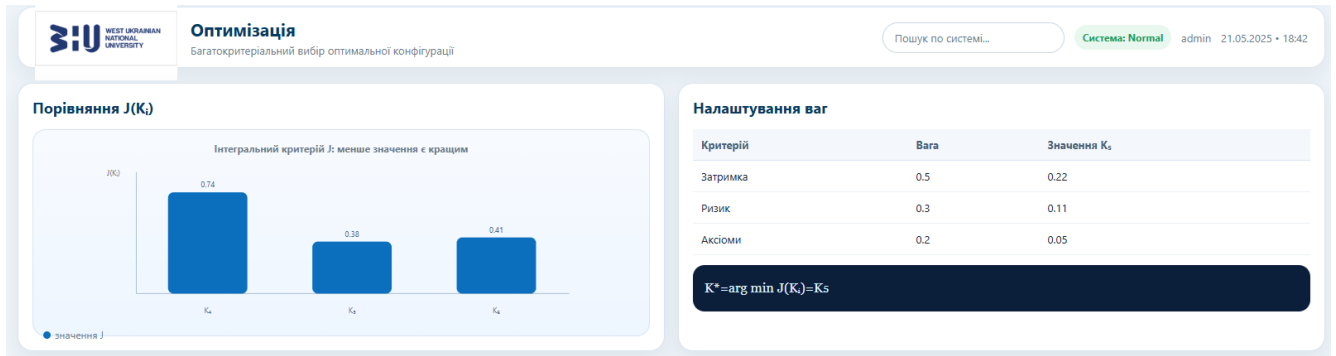


Рис. 4.7. Сторінка оптимізації конфігурацій комп'ютерної мережі

Основною метою оптимізації є мінімізація інтегрального критерію функціонування мережі, який враховує продуктивність обчислювальних ресурсів, рівень завантаження серверів, затримки обслуговування користувачів, ризики виникнення перевантажень та якість функціонування сервісів. Таким чином реалізується багатокритеріальний підхід до управління мережею, що забезпечує збалансоване використання ресурсів корпоративної інфраструктури.

Сторінка логічного виведення представлена на рисунку 4.8. Однією з принципових відмінностей запропонованої системи від існуючих рішень є використання механізмів логічного виведення на основі онтологічних аксіом. Відповідно до цього розроблено окрему сторінку, яка забезпечує пояснення процесу формування управлінських рішень.

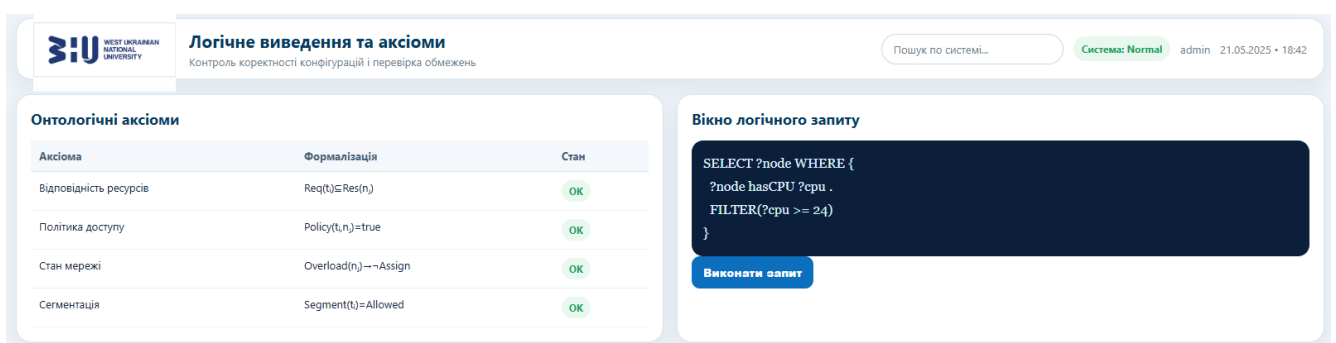


Рис. 4.8. Сторінка логічного виведення

Необхідність створення даного модуля обумовлена результатами попередніх розділів дисертаційної роботи, у яких було розроблено онтологічну модель корпоративної мережі та метод онтологічно-керованого конфігурування. Якщо онтологічна база знань забезпечує формалізоване представлення знань про предметну область, то сторінка логічного виведення реалізує механізми їх практичного використання для підтримки процесів прийняття управлінських рішень.

На відміну від класичних систем управління мережею, де рішення приймаються безпосередньо адміністратором на основі аналізу технічних параметрів, у запропонованій системі значна частина процесів аналізу виконується автоматизовано за допомогою механізмів семантичного виведення. При цьому логічні висновки формуються на основі сукупності фактів, що надходять із підсистем моніторингу, конфігурування, управління задачами та політик безпеки.

Окремим функціональним блоком сторінки є підсистема аналізу причинно-наслідкових зв'язків. Вона дозволяє встановлювати взаємозв'язки між подіями в мережі, змінами навантаження, порушеннями політик безпеки та станом окремих сервісів. Наприклад, система може автоматично встановити, що збільшення часу відгуку певного сервісу пов'язане з перевантаженням конкретного центру термінального кластера або дефіцитом обчислювальних ресурсів на одному із серверів.

Важливою особливістю реалізованого інтерфейсу є підтримка пояснюваності процесу оптимізації. Після завершення роботи оптимізаційного модуля користувач може переглянути не лише кінцевий результат, але й обґрунтування вибору конкретної конфігурації або схеми розподілу навантаження. Для цього система відображає послідовність активованих правил, результати перевірки аксіом та критерії, які вплинули на прийняття рішення. Сторінка також забезпечує інтеграцію з онтологічним графом. Для кожного логічного висновку користувач може перейти до відповідних сутностей онтології та переглянути семантичні зв'язки між ресурсами, сервісами, задачами та правилами, які брали участь у

процесі виведення. Це значно підвищує інтерпретованість результатів та спрощує аналіз складних сценаріїв функціонування мережі.

Сторінка політик та безпеки відображає механізм інтеграції в діючі конфігурація, рисунок 4.9. Враховуючи критичну роль інформаційної безпеки у корпоративних мережах, у системі реалізовано спеціалізований модуль управління політиками безпеки.

The screenshot displays the 'Політики та безпека' (Policies and Security) interface. At the top, there is a header with the ZNU logo and the title 'Політики та безпека' (Policies and Security), with a subtitle 'Контроль доступу, сегментації та правил корпоративної мережі' (Access control, segmentation and corporate network rules). A search bar and system status 'Система: Normal' are also visible.

The main content is divided into two sections:

- Політики сегментації** (Segmentation Policies): A table with three columns: 'Сегмент' (Segment), 'Дозволені сервіси' (Allowed services), and 'Стан' (Status).

Сегмент	Дозволені сервіси	Стан
COL-SEG-FIN	BusinessApp, DB, Logging	активна
NET-SEG-A	LMS, Auth, DB	активна
COL-SEG-GEN	General services	обмежено
- Автентифікація** (Authentication): Three cards representing different methods:
 - LDAP**: каталог користувачів (user catalog)
 - Kerberos**: квитки доступу (access tickets)
 - RBAC**: ролі адміністратора, користувача, DSS (administrator, user, DSS roles)

Рис. 4.9. Сторінка політик та безпеки

Необхідність реалізації такого функціонального модуля безпосередньо впливає з результатів попередніх розділів дисертаційної роботи. Розроблена онтологічна модель корпоративної мережі включає не лише ресурси, сервіси та задачі, але й множину правил, обмежень та аксіом, які визначають допустимі умови функціонування мережевого середовища. Саме ці знання формують основу для автоматизованого контролю безпеки та підтримки процесів прийняття рішень щодо конфігурування ресурсів.

На відміну від традиційних систем адміністрування, де політики безпеки часто реалізуються у вигляді розрізнених конфігураційних параметрів окремих пристроїв, у запропонованій системі вони представлені як складова єдиної онтологічної бази знань. Такий підхід забезпечує централізоване управління правилами доступу та дозволяє враховувати їх на всіх етапах функціонування мережі, включаючи моніторинг, конфігурування, логічне виведення та оптимізацію.

Інтерфейс сторінки дозволяє формувати та редагувати політики доступу до

мережевих ресурсів відповідно до ролей користувачів. Для кожної категорії користувачів визначаються дозволені сервіси, ресурси та операції, які можуть виконуватись у межах корпоративної мережі. Такий механізм забезпечує реалізацію принципу мінімально необхідних привілеїв та сприяє зменшенню ризиків несанкціонованого доступу.

Особливістю реалізованого підходу є інтеграція політик безпеки з онтологічною моделлю мережі. Кожна політика представлена як окремий об'єкт бази знань та пов'язана з відповідними ресурсами, сервісами, користувачами та задачами. У результаті система отримує можливість автоматично враховувати вимоги безпеки під час формування конфігураційних рішень і розподілу навантаження між центрами термінальних кластерів. Важливе місце на сторінці займає підсистема аналізу відповідності конфігурацій політикам безпеки. Після формування нової конфігурації або виконання процедури оптимізації система автоматично перевіряє її на відповідність визначеним правилам доступу, сегментації мережі та вимогам інформаційної безпеки. У випадку виявлення порушень користувач отримує детальну інформацію про невідповідності та рекомендації щодо їх усунення.

Сторінка звітів та журналів відображена на рисунку 4.10. Завершальним елементом користувацького інтерфейсу є підсистема звітності та аудиту. Вона забезпечує накопичення історичних даних про стан мережі, результати конфігурування, роботу оптимізаційних алгоритмів та дії користувачів системи.

Звіти та журнал дій
Аналіз ефективності, історія конфігурацій і подій системи

Пошук по системі... Система: Normal admin 21.05.2025 • 18:42

Час	Користувач	Дія	Статус
18:41:02	admin	Застосовано конфігурацію K ₂	OK
18:40:11	admin	Переглянуто аксіоми	OK
18:39:55	system	Логічне виведення виконано	OK
18:39:21	admin	Запущено оптимізацію	OK

Підсумкові показники K₅

61 мс затримка	0.16 ризик	0.38 J
--------------------------	----------------------	------------------

Конфігурація забезпечує виконання вимог задачі бізнес-аналітики, політик безпеки та онтологічних обмежень.

Рис. 4.10. Сторінка звітів та журналів дій

У структурі сторінки реалізовано декілька взаємопов'язаних функціональних блоків, кожен з яких виконує окрему аналітичну функцію. Перший блок забезпечує ведення системного журналу подій. У ньому відображаються всі дії користувачів та автоматизованих компонентів програмної системи. Для кожного запису зберігаються часові мітки, тип події, джерело її виникнення, задіяні ресурси та результат виконання відповідної операції. Такий підхід забезпечує повну простежуваність процесів управління мережею та створює основу для проведення аудиту функціонування системи.

Другий блок призначений для аналізу конфігураційних змін. Оскільки розроблений метод онтологічно-керованого конфігурування передбачає автоматичне формування нових конфігурацій ресурсів, виникає необхідність збереження інформації про всі виконані зміни. На сторінці відображається історія конфігурацій, результати перевірки онтологічних аксіом та показники ефективності відповідних рішень після їх впровадження в мережевому середовищі.

Особливу роль сторінка відіграє під час оцінювання ефективності запропонованої математичної моделі розподілу навантаження між центрами термінальних кластерів. У відповідному розділі звітності відображаються показники навантаження до та після застосування оптимізаційних процедур, зміни середнього часу обслуговування користувачів, коефіцієнтів використання ресурсів та інтегральних критеріїв якості функціонування мережі. Це дозволяє об'єктивно оцінювати практичний ефект від використання розроблених моделей та методів.

Важливою особливістю сторінки є підтримка засобів візуальної аналітики. Результати роботи мережі відображаються у вигляді інтерактивних графіків, гістограм, часових діаграм, теплових карт навантаження та інформаційних панелей. Використання засобів візуалізації суттєво спрощує сприйняття великих обсягів інформації та підвищує ефективність аналізу складних процесів функціонування корпоративної мережі.

З метою забезпечення вимог інформаційної безпеки реалізовано механізми аудиту доступу до журналів та звітів. Кожна операція перегляду або зміни

аналітичної інформації фіксується в системному журналі та може бути використана для проведення подальшого розслідування інцидентів безпеки або перевірки коректності дій адміністраторів. Особливу увагу приділено можливості експорту результатів аналізу. Система підтримує автоматичне формування звітів у форматах PDF, XLSX та CSV, що забезпечує інтеграцію з іншими корпоративними інформаційними системами та дозволяє використовувати результати аналізу для підготовки управлінської документації.

Розроблений користувацький інтерфейс є логічним продовженням запропонованих у попередніх розділах математичних моделей, методів та архітектурних рішень. Він забезпечує практичну реалізацію знання-орієнтованого підходу до управління корпоративними комп'ютерними мережами, інтегруючи функції моніторингу, конфігурування, оптимізації, логічного виведення та підтримки прийняття рішень у межах єдиного програмного середовища.

4.2 Експериментальні дослідження методу онтологічно-керованої конфігурації корпоративної мережі

Експериментальна перевірка запропонованого методу виконувалася на двох прикладних середовищах, що відрізняються за функціональним призначенням, структурою мережі, складом сервісів і профілем навантаження. Перший експеримент стосується корпоративної мережі Західноукраїнського національного університету, в якій домінують навчальні, адміністративні та інформаційно-сервісні задачі. Другий експеримент проведено для мережі компанії «Колумбус», де основне навантаження формують бізнес-процеси, доступ до корпоративних прикладних систем та аналітичних сервісів.

Метою експериментів є встановлення того, чи забезпечує метод, побудований на основі онтології корпоративної мережі та алгебри кортежів, побудову конфігурації, яка краще відповідає вимогам задачі, ніж початкова або інтуїтивно сформована конфігурація. У рамках експериментів оцінювалися

ресурсоемність задач, відповідність вузлів вимогам задач, рівень завантаження серверів, затримка доступу до сервісів, інтегральний критерій якості конфігурації та кількість конфігурацій, відсіяних онтологічними аксіомами.

У першому експерименті розглядалася задача забезпечення функціонування системи дистанційного навчання та електронного тестування в період пікового навантаження. Така задача є характерною для університетського середовища, оскільки поєднує одночасний доступ великої кількості користувачів, потребу в роботі вебсервісів, сервісів автентифікації, баз даних та файлових ресурсів.

Задача T_1 відповідає сценарію проведення масового онлайн-тестування. Для неї були задані наступні вимоги:

- кількість одночасних користувачів: 850;
- мінімально необхідна кількість логічних ядер CPU: 24;
- необхідний обсяг оперативної пам'яті: 64 ГБ;
- необхідний обсяг дискового простору для активної роботи сервісів: 500 ГБ;
- мінімальна пропускна здатність сегмента: 1 Гбіт/с;
- наявність сервісів: вебсервіс навчальної платформи, сервіс автентифікації, сервер баз даних;
- обов'язкова наявність політики ізоляції навчального сегмента та журналювання доступу;
- допустима середня затримка не більше 120 мс.

У таблиці 4.1. представлено параметри вузлів мережі ЗУНУ, використаних в експерименті.

Для кожного вузла в онтології були задані класи, атрибути та відношення типу `hostsService`, `hasCPUCapacity`, `hasMemoryCapacity`, `locatedInSubnet`, `protectedBy`, `hasAvailabilityStatus`.

Таблиця 4.1 – Параметри вузлів мережі ЗУНУ

Вузол	Тип	CPU, ядер	RAM, ГБ	Storage, ГБ	Bandwidth, Мбіт/с	Стан	Розміщені сервіси
EDU-SRV-01	Application Server	16	32	500	1000	нормальний	LMS Web
EDU-SRV-02	Application Server	24	64	1000	1000	нормальний	LMS Web, Auth
EDU-DB-01	Database Server	16	64	2000	1000	нормальний	DB
EDU-DB-02	Database Server	32	128	4000	1000	частково перевантажений	DB
AUTH-SRV-01	Authentication Server	8	16	250	1000	нормальний	Auth
FILE-SRV-01	File Server	12	32	6000	1000	нормальний	Files
NET-SEG-A	Network Segment	—	—	—	1000	нормальний	VLAN-Academic
NET-SEG-B	Network Segment	—	—	—	100	нормальний	VLAN-Admin

У таблицях 4.2-4.8 представлено початкові дані онтології корпоративної мережі.

Таблиця 4.2 - Елементи мережевої інфраструктури (M)

Id	Вузол	Клас	CPU (ядра)	RAM (ГБ)	Storage (ГБ)	Bandwidth (Мбіт/с)	Стан	Сервіси
M_1	EDU-SRV-01	ApplicationServer	16	32	500	1000	Normal	LMS
M_2	EDU-SRV-02	ApplicationServer	24	64	1000	1000	Normal	LMS, Auth
M_3	EDU-DB-01	DatabaseServer	16	64	2000	1000	Normal	DB
M_4	EDU-DB-02	DatabaseServer	32	128	4000	1000	HighLoad	DB
M_5	AUTH-SRV-01	AuthServer	8	16	250	1000	Normal	Auth
M_6	COL-APP-02	ApplicationServer	24	64	1000	1000	Normal	App, Log
M_7	COL-DB-01	DatabaseServer	16	64	2000	1000	Normal	DB
M_8	COL-LOG-01	LoggingServer	8	16	1000	1000	Normal	Log

Таблиця 4.3 - Параметри мережевої інфраструктури (P)

Id	Назва	Тип	Опис	Діапазон / значення
P_1	CPU_Load	Resource	завантаження CPU	[0;1]
P_2	Memory	Resource	обсяг оперативної пам'яті	ГБ
P_3	Latency	Network	затримка	мс
P_4	Bandwidth	Network	пропускна здатність	Мбіт/с
P_5	AccessLevel	Security	рівень доступу	role-based
P_6	RiskScore	Security	рівень ризику	[0;1]

Таблиця 4.4 – Стани мережі (S)

Id	Стан	Умова
S_1	Normal	CPU_Load < 0.7
S_2	HighLoad	CPU_Load > 0.85
S_3	Degraded	Latency > 150 мс
S_4	Secure	ізольований сегмент

Таблиця 4.5 – Характеристики виконання мережі (E)

Id	Показник	Тип	Значення
E_1	ResponseTime	Performance	< 120 мс
E_2	Throughput	Performance	> 500 req/s
E_3	Availability	Reliability	> 99.5%
E_4	ErrorRate	Quality	< 2%

Подані таблиці 4.2-4.7 містять структурований набір початкових даних онтології корпоративної мережі, які використовуються як вхід для реалізації методу конфігурації. На відміну від неструктурованого опису, розділення даних за компонентами множини O_{net} дозволяє забезпечити однозначну інтерпретацію кожної сутності та її ролі у процесі формування конфігурації.

Множина M задає простір доступних ресурсів, U — функціональний контекст задач, P — кількісні характеристики, S — динамічні стани системи, E — критерії ефективності, а A — логічні обмеження. Саме ця структурована база даних

використовується для виконання операцій алгебри кортежів, що приводить до формування множини допустимих конфігурацій.

Таблиця 4.6 – Онтологічні аксіоми (A)

<i>Id</i>	Назва	Формула	Інтерпретація
A ₁	ResourceSufficiency	$Req(t) \subseteq Cap(n)$	достатність ресурсів
A ₂	ServiceCompatibility	$Service(t) \in Services(n)$	наявність сервісу
A ₃	SecurityPolicy	$Access \subseteq Policy$	контроль доступу
A ₄	StateConstraint	$State(n) \in Allowed$	допустимий стан
A ₅	RiskConstraint	$Risk(n) < threshold$	обмеження ризику
A ₆	Segmentation	$Node \in SecureSegment$	ізоляція

У таблиці 3.7. представлено формалізацію задач, на основі яких проводилися експериментальні дослідження.

Таблиця 4.7 – Формалізація аналізованих задач (T)

<i>Id</i>	Назва	Користувачі	CPU	RAM	Latency	Сервіси
T ₁	OnlineTesting	850	24	64	<120 мс	LMS, Auth, DB
T ₂	BusinessAnalytics	180	16	48	<70 мс	App, DB, Log

У таблиці 4.8. відображено поетапне проходження елементів мережевої інфраструктури через послідовність операцій методу конфігурації.

У першому стовпці наведено вихідні записи відношення M_{Rel} , які містять вузли мережі та їх основні характеристики. На рівні R_1 перевіряється, чи відповідає структурний опис вузла типу задачі. Для задачі T_1 , що пов'язана з онлайн-тестуванням, релевантними є лише сервери навчального середовища, сервери баз даних та сервери автентифікації, розміщені у відповідному сегменті університетської мережі.

На рівнях R_2 і R_3 здійснюється узгодження вузлів із об'єктами використання та параметрами задачі. Це означає, що вузол має бути пов'язаний із необхідним

сервісом — у цьому випадку з LMS, DB або Auth — та мати відповідні параметри для подальшого аналізу. Після цього на рівні R_4 виконується перевірка параметричних вимог задачі. Зокрема, для задачі T_1 встановлено мінімальні значення для CPU, RAM, обсягу дискового простору та пропускної здатності. Саме на цьому етапі вузол EDU-SRV-01 виключається з подальшого розгляду, оскільки його ресурсні характеристики не задовольняють вимоги задачі.

Таблиця 4.8 – Знання-орієнтований аналіз формування конфігурації для задачі T_1

Id вузла	M_{Rel} : вузол і характеристики	(R_1) : структурна відповідність ($M_{struct}^{=p_i}$)	(R_2) – (R_3) : узгодження з об'єктами використання та параметрами	(R_4) : відповідність параметричним вимогам	(R_5) – (R_6) : допустимий стан мережі	(R_7) : характеристики виконання	(R_8) : перевірка аксіом	Включення до (<i>Config Rel</i>)
M_1	EDU-SRV-01, ApplicationServer, CPU=16, RAM=32, Storage=500, BW=1000	так	пов'язаний з LMS	ні (CPU та RAM нижчі за вимоги задачі)	—	—	—	ні
M_2	EDU-SRV-02, ApplicationServer, CPU=24, RAM=64, Storage=1000, BW=1000	так	пов'язаний з LMS, Auth	так	стан Normal	Response Time = 96 мс	усі аксіоми виконано	так
M_3	EDU-DB-01, DatabaseServer, CPU=16, RAM=64, Storage=2000, BW=1000	так	пов'язаний з DB	так	стан Normal	DB response acceptable	усі аксіоми виконано	так
M_4	EDU-DB-02, DatabaseServer, CPU=32, RAM=128, Storage=4000, BW=1000	так	пов'язаний з DB	так	стан HighLoad	затримка прийнятна	порушено аксіому допустимого стану	ні
M_5	AUTH-SRV-01, AuthServer, CPU=8, RAM=16, Storage=250, BW=1000	так	пов'язаний з Auth	так	стан Normal	Auth response acceptable	усі аксіоми виконано	так
M_6	COL-APP-02, ApplicationServer, CPU=24, RAM=64, Storage=1000, BW=1000	ні	—	—	—	—	—	ні
M_7	COL-DB-01, DatabaseServer, CPU=16, RAM=64, Storage=2000, BW=1000	ні	—	—	—	—	—	ні
M_8	COL-LOG-01, LoggingServer, CPU=8, RAM=16, Storage=1000, BW=1000	ні	—	—	—	—	—	ні

На рівнях R_5 і R_6 враховується поточний стан вузлів. Цей етап є принципово важливим для динамічного середовища, оскільки навіть ресурсно потужний вузол

не може бути включений до конфігурації, якщо він перебуває у стані перевантаження або деградації. Саме тому сервер EDU-DB-02, попри достатній ресурсний потенціал, виключається через порушення аксіоми допустимого стану.

На рівні R_7 враховуються характеристики виконання задачі, зокрема допустима затримка та очікувана якість функціонування сервісу. Однак остаточне рішення щодо включення вузла до конфігурації приймається лише після перевірки онтологічних аксіом на рівні R_8 . У даному випадку такими аксіомами виступають аксіоми достатності ресурсів, наявності потрібного сервісу, допустимого стану вузла та відповідності задачі навчальному сегменту мережі.

У результаті до кінцевого відношення *ConfigRel* входять лише ті вузли, які пройшли всі етапи фільтрації та перевірки. Для задачі T_1 це EDU-SRV-02, EDU-DB-01 та AUTH-SRV-01, що разом формують допустиму та оптимальну конфігурацію для розв'язання задачі онлайн-тестування.

Для задачі T_1 кінцеве відношення конфігурації можна подати як

$$ConfigRel_{T_1} = \left\{ \begin{array}{l} (M_2, U_3, P_{cpu}, S_1, E_1), \\ (M_3, U_3, P_{db}, S_1, E_1), (M_5, U_4, P_{auth}, S_1, E_1) \end{array} \right\}$$

В даній конфігурації M_2 відповідає серверу EDU-SRV-02, M_3 серверу EDU-DB-01, M_5 серверу AUTH-SRV-01, U_3 сервісу LMS, U_4 сервісу автентифікації, S_1 нормальному стану, E_1 - допустимому часу відгуку.

Після відбору за структурними, параметричними та безпековими обмеженнями метод сформував три допустимі конфігурації для задачі T_1 , які представлено у таблиці 4.9.

Конфігурація K_1 виявилася ресурсно граничною: хоч вона формально забезпечувала необхідні сервіси, однак використання сервера EDU-SRV-01 призводило до перевищення допустимого рівня навантаження. Конфігурація K_3 була відсіяна на підставі аксіоми стану мережі, оскільки сервер EDU-DB-02 перебував у частково перевантаженому стані.

Найкращою виявилася конфігурація K_2 , яка задовольнила всі онтологічні аксіоми та забезпечила мінімальне значення інтегрального критерію.

Таблиця 4.9 Допустимі конфігурації для задачі T_1 онлайн-тестування

Конфігурація	Application Server	DB Server	Auth Server	Segment	Виконання аксіом	Середня затримка, мс	Завантаження CPU, %	Інтегральний критерій (J)
(K_1)	EDU-SRV-01	EDU-DB-01	AUTH-SRV-01	NET-SEG-A	частково	148	91	0.79
(K_2)	EDU-SRV-02	EDU-DB-01	EDU-SRV-02	NET-SEG-A	повністю	96	73	0.42
(K_3)	EDU-SRV-02	EDU-DB-02	AUTH-SRV-01	NET-SEG-A	не виконує аксіому стану	102	68	0.61

У таблиці 4.10 представлено порівняння початкової та оптимальної конфігурацій для мережі ЗУНУ при розв'язанні заданого класу задач.

Таблиця 4.10 – Порівняння показників до і після застосування досліджуваного методу для мережі ЗУНУ

Показник	Початкова конфігурація	Оптимальна конфігурація (K_2)	Зміна
Середня затримка доступу, мс	151	96	-36.4%
Середнє завантаження CPU application server, %	92	73	-20.7%
Частка відмов авторизації, %	4.8	1.3	-72.9%
Частка конфліктів доступу до БД, %	3.1	0.9	-71.0%
Коефіцієнт використання ресурсів	0.94	0.78	-17.0%
Інтегральний критерій якості (J)	0.81	0.42	-48.1%

На рисунку 4.11 представлено нормоване порівняння показників

функціонування корпоративної мережі до та після застосування методу конфігурації для задачі онлайн-тестування. Нормування виконано відносно значень початкової конфігурації, що дозволяє привести всі показники до єдиної шкали [0,1] та забезпечити їх коректне порівняння.

Аналіз графіка свідчить про суттєве покращення всіх показників після застосування методу. Найбільше зниження спостерігається для показників відмов авторизації та конфліктів доступу до баз даних, що підтверджує ефективність врахування функціональних і логічних обмежень у межах онтологічного підходу.

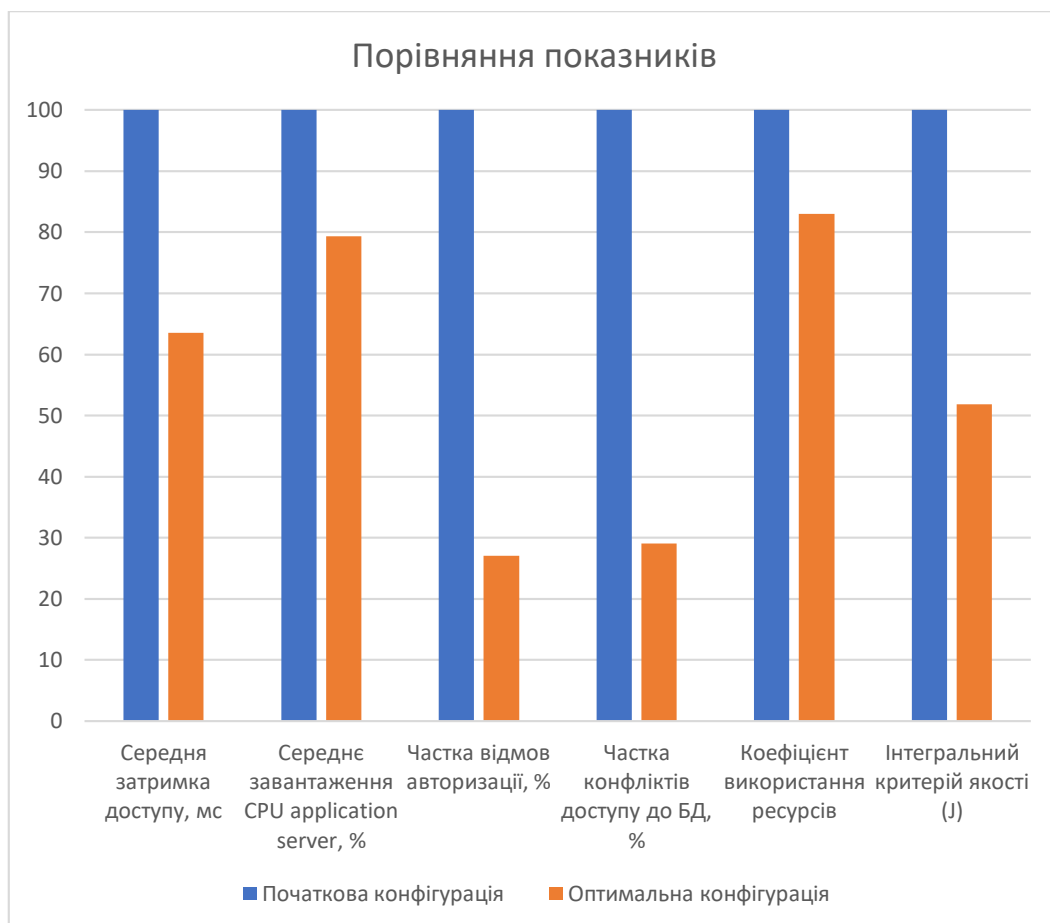


Рис. 4.11. Порівняння показників до та після застосування методу (задача T_1)

Зменшення інтегрального критерію якості конфігурації з 0.81 до 0.42 підтверджує підвищення узгодженості конфігурації з вимогами задачі та поточним станом мережі, рисунок 4.12. Отримані результати показують, що використання онтологічно-керованого підходу дозволило не лише знайти допустиму

конфігурацію, але й суттєво підвищити ефективність розв'язання задачі. Особливо показовим є зменшення середньої затримки та зниження частоти конфліктів при доступі до сервісів автентифікації та баз даних.

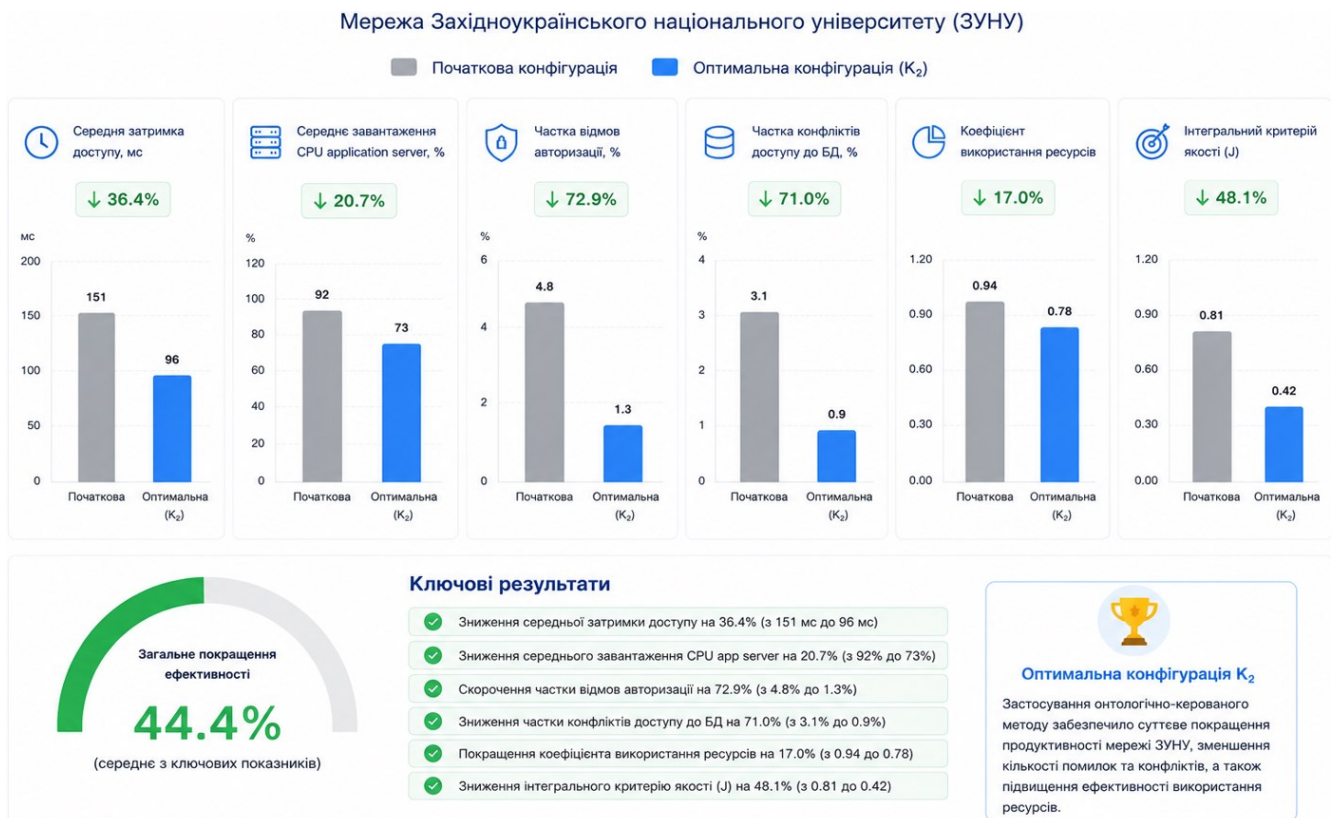


Рис. 4.12. Оцінка ефективності управління корпоративною комп'ютерною мережею ЗНУ

У другому експерименті досліджувалася задача T_2 конфігурації корпоративної мережі для забезпечення роботи бізнес-аналітичного сервісу з доступом до централізованої бази даних, файлового сховища та прикладного сервера компанії «Колумбус». На відміну від університетського середовища, у цьому випадку головними є не масовість одночасних користувачів, а стабільність роботи бізнес-сервісу, низька затримка, висока доступність і дотримання суворих політик доступу до даних. Задача T_2 характеризується наступними вимогами:

- кількість одночасних користувачів: 180;
- мінімально необхідна кількість логічних ядер CPU: 16;

- необхідний обсяг оперативної пам'яті: 48 ГБ;
- необхідний обсяг дискового простору для активної роботи сервісів: 800 ГБ;
- мінімальна пропускна здатність сегмента: не менше 500 Мбіт/с;
- наявність сервісів: application service, database service, logging service;
- політики: рольовий доступ, сегментація фінансового контуру, журналювання подій;
- допустима середня затримка не більше 70 мс;
- заборона використання вузлів із критичними вразливостями.

Таблиця 4.11 – Параметри вузлів мережі компанії «Колумбус»

Вузол	Тип	CPU, ядер	RAM, ГБ	Storage, ГБ	Bandwidth, Мбіт/с	Стан	Сервіси
COL-APP-01	Application Server	12	32	500	1000	нормальний	App
COL-APP-02	Application Server	24	64	1000	1000	нормальний	App, Log
COL-DB-01	Database Server	16	64	2000	1000	нормальний	DB
COL-DB-02	Database Server	24	128	4000	1000	нормальний	DB
COL-LOG-01	Logging Server	8	16	1000	1000	нормальний	Log
COL-FS-01	File Server	12	32	6000	1000	нормальний	Files
COL-SEG-FIN	Secure Segment	—	—	—	1000	нормальний	VLAN-Finance
COL-SEG-GEN	General Segment	—	—	—	1000	нормальний	VLAN-General

У таблиці 4.12 наведено результати послідовного проходження вузлів корпоративної мережі через етапи методу конфігурації для задачі T_2 , яка відповідає сценарію функціонування бізнес-аналітичного сервісу в мережі компанії «Колумбус». На відміну від задачі T_1 , у даному випадку релевантними є виключно вузли корпоративної інфраструктури компанії, тому всі елементи університетської мережі виключаються вже на етапі R_1 , оскільки їх структурний опис не відповідає предметній області та сервісному профілю задачі.

Таблиця 4.12 – Знання-орієнтований аналіз формування конфігурації для задачі T_2

Id вузла	M_{Rel} : вузол і характеристи ки	(R_1) : структурна відповідність $(M_{Struct})_{=Pi}$	$(R_2)-(R_3)$: узгодження з об'єктами використання та параметрами	(R_4) : відповідність параметричним вимогам	$(R_5)-(R_6)$: допустимий стан мережі	(R_7) : характеристики виконання	(R_8) : перевірка аксіом	Включення до $(Config Rel)$
M_1	EDU-SRV-01, ApplicationServer, CPU=16, RAM=32, Storage=500, BW=1000	ні	—	—	—	—	—	ні
M_2	EDU-SRV-02, ApplicationServer, CPU=24, RAM=64, Storage=1000, BW=1000	ні	—	—	—	—	—	ні
M_3	EDU-DB-01, DatabaseServer, CPU=16, RAM=64, Storage=2000, BW=1000	ні	—	—	—	—	—	ні
M_4	EDU-DB-02, DatabaseServer, CPU=32, RAM=128, Storage=4000, BW=1000	ні	—	—	—	—	—	ні
M_5	AUTH-SRV-01, AuthServer, CPU=8, RAM=16, Storage=250, BW=1000	ні	—	—	—	—	—	ні
M_6	COL-APP-02, ApplicationServer, CPU=24, RAM=64, Storage=1000, BW=1000	так	пов'язаний з BusinessApp, Log	так	стан Normal	ResponseTime = 61 мс	усі аксіоми виконано	так
M_7	COL-DB-01, DatabaseServer, CPU=16, RAM=64, Storage=2000, BW=1000	так	пов'язаний з DB	так	стан Normal	DB response acceptable	усі аксіоми виконано	так
M_8	COL-LOG-01, LoggingServer, CPU=8, RAM=16, Storage=1000, BW=1000	так	пов'язаний з Log	так	стан Normal	logging throughput acceptable	усі аксіоми виконано	так

Для задачі T_2 кінцеве відношення конфігурації можна подати як

$$ConfigRel_{T_2} = \left\{ \begin{array}{l} (M_6, U_5, P_{app}, S_1, E_1), \\ (M_7, U_5, P_{db}, S_1, E_1), (M_8, U_5, P_{log}, S_1, E_1) \end{array} \right\}$$

В даній конфігурації M_6 відповідає вузлу COL-APP-02, M_7 вузлу COL-DB-01, M_8 вузлу COL-LOG-01, U_5 сервісу BusinessApp, S_1 нормальному стану функціонування, E_1 - допустимому рівню часу відгуку.

На рівні R_1 здійснюється відбір лише тих елементів мережі, які структурно придатні для забезпечення бізнес-аналітичного процесу. Для задачі T_2 такими елементами є прикладний сервер, сервер баз даних і сервер журналювання. Саме тому вузли COL-APP-02, COL-DB-01 та COL-LOG-01 проходять первинну фільтрацію.

На рівнях R_2 – R_3 встановлюється їх відповідність об'єктам використання та сервісам, потрібним для задачі. Зокрема, COL-APP-02 асоціюється з прикладним сервісом BusinessApp і сервісом журналювання, COL-DB-01 — з сервером баз даних, а COL-LOG-01 — із сервісом логування. Після цього на рівні R_4 перевіряється відповідність параметричних характеристик вузлів вимогам задачі T_2 , які передбачають не менше 16 логічних ядер CPU, 48 ГБ оперативної пам'яті, 800 ГБ дискового простору та достатню пропускну здатність мережевого сегмента.

На рівнях R_5 – R_6 враховується стан вузлів. У межах даного експерименту всі три релевантні вузли перебувають у стані Normal, що дозволяє не виключати їх за критерієм експлуатаційного стану. На рівні R_7 до розгляду додаються характеристики виконання задачі, зокрема середня затримка бізнес-сервісу, яка для COL-APP-02 становить 61 мс і не перевищує допустимого значення 70 мс.

Остаточне рішення щодо включення вузлів до конфігурації приймається на рівні R_8 , де виконується перевірка онтологічних аксіом. У цьому експерименті принципове значення мають аксіоми достатності ресурсів, сервісної сумісності, допустимого стану, а також аксіома безпекової сегментації. Оскільки розглянуті

вузли належать до допустимого сегмента мережі компанії та не мають критичних суперечностей із безпековими політиками, усі вони входять до кінцевого відношення конфігурації. У результаті для задачі T_2 до складу конфігурації входять вузли COL-APP-02, COL-DB-01 та COL-LOG-01, які разом утворюють узгоджену конфігурацію для забезпечення функціонування бізнес-аналітичного сервісу. У таблиці 4.13. представлено результати конфігурації мережі із застосуванням розробленого онтологічно-керованого методу.

Таблиця 4.13 – Допустимі конфігурації для задачі бізнес-аналітики

Конфігурація	App Server	DB Server	Logging	Segment	Виконання аксіом	Середня затримка, мс	Ризик	(J)
(K_4)	COL-APP-01	COL-DB-01	COL-LOG-01	COL-SEG-GEN	не виконує політику сегментації	84	0.41	0.74
(K_5)	COL-APP-02	COL-DB-01	COL-APP-02	COL-SEG-FIN	повністю	61	0.16	0.38
(K_6)	COL-APP-02	COL-DB-02	COL-LOG-01	COL-SEG-FIN	повністю	58	0.19	0.41

У даному експерименті конфігурація K_4 була виключена на основі онтологічної аксіоми безпекової сегментації, оскільки прикладний і базовий сервіси розміщувалися в загальному сегменті мережі, що суперечило вимогам задачі. Конфігурації K_5 та K_6 виявилися допустимими, однак конфігурація K_5 мала менше значення інтегрального критерію за рахунок кращого балансу між навантаженням, ризиком та затримкою.

У таблиці 4.13 представлено порівняння початкової та оптимальної конфігурацій для мережі компанії «Колумбус» при розв'язанні заданого класу задач.

Таблиця 4.13 – Показники до і після застосування онтологічно-керованого методу для мережі компанії «Колумбус»

Показник	Початкова конфігурація	Оптимальна конфігурація (K_5)	Зміна
Середня затримка бізнес-сервісу, мс	87	61	-29.9%
Середнє завантаження app server, %	88	67	-23.9%
Кількість порушень політик доступу за добу	7	1	-85.7%
Кількість конфігурацій, відсіяних аксіомами	—	4 з 7	—
Інтегральний ризик	0.43	0.16	-62.8%
Інтегральний критерій якості (J)	0.77	0.38	-50.6%

На рисунку 4.13 представлено порівняння основних показників функціонування корпоративної мережі до та після застосування методу онтологічно-керованої конфігурації для задачі бізнес-аналітики. Нормування виконано відносно значень початкової конфігурації, що дозволяє привести всі показники до єдиної шкали $[0, 1]$. Як видно з діаграми, застосування методу забезпечує системне покращення всіх розглянутих показників. Найбільш суттєві зміни спостерігаються для кількості порушень політик доступу та інтегрального ризику, що свідчить про підвищення рівня безпеки системи. Водночас зменшення затримки та навантаження на сервери підтверджує покращення продуктивності мережі.

Зниження інтегрального критерію якості конфігурації з 0.77 до 0.38 підтверджує ефективність запропонованого підходу та його здатність формувати збалансовані конфігурації з урахуванням ресурсних, функціональних і безпекових обмежень.

Отримані результати показали, що онтологічно-керований метод не лише забезпечив підбір ресурсно достатньої конфігурації, але й суттєво посилив узгодженість мережі з безпековими та функціональними вимогами прикладної

задачі, рисунок 4.14.

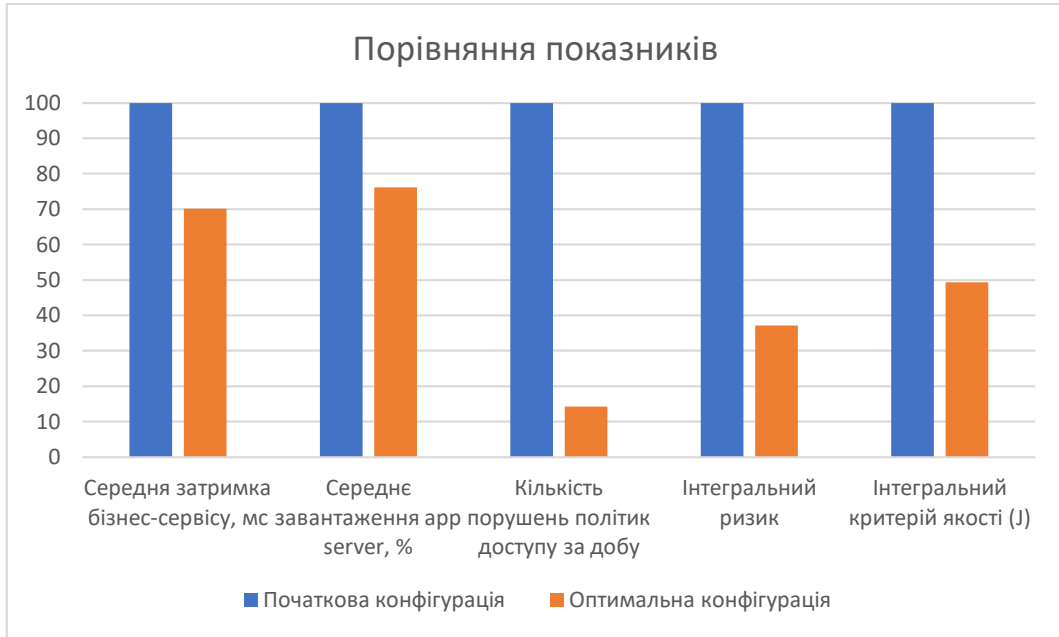


Рис. 4.13. Порівняння показників до та після застосування методу (задача T_2)

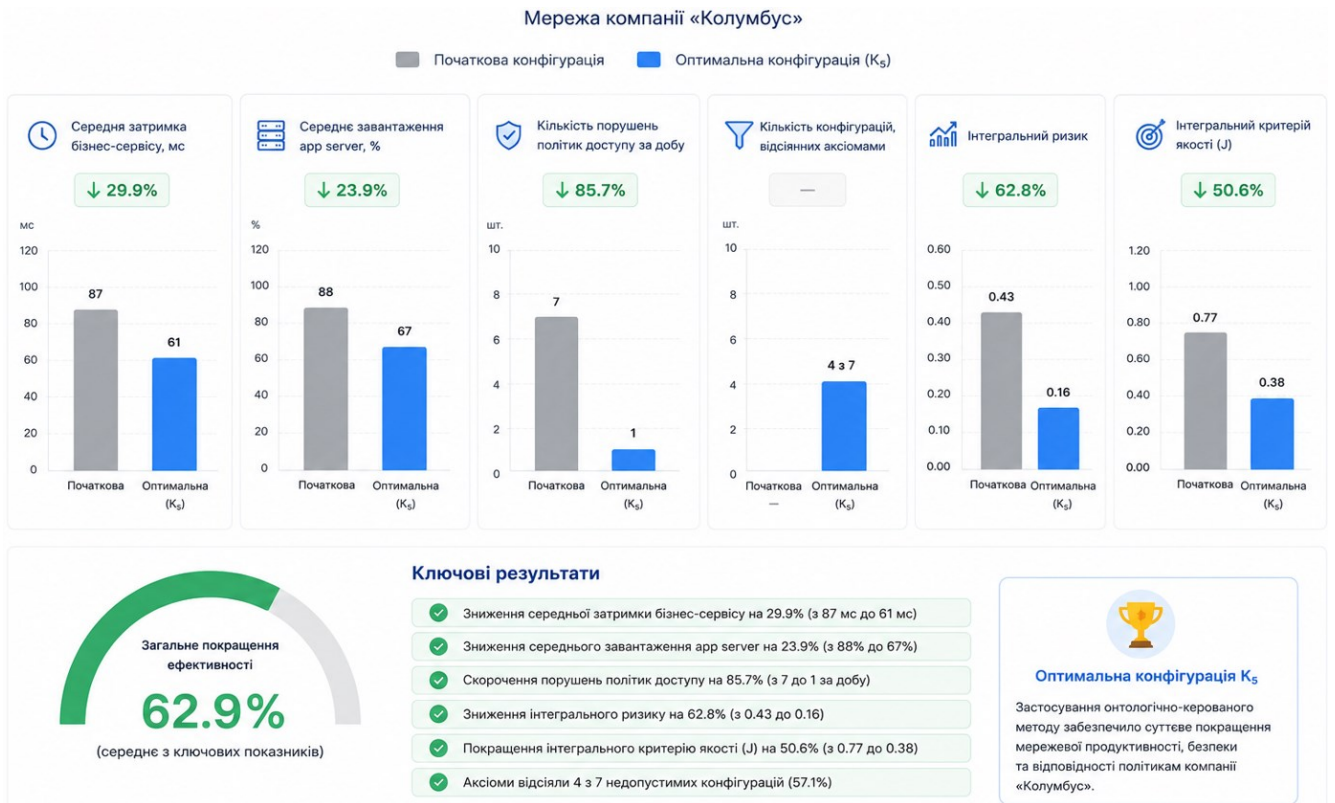


Рис. 4.14. Оцінка ефективності управління корпоративною комп'ютерною мережею компанії «Колумбус»

Аналіз результатів обох експериментів засвідчив, що запропонований метод має стабільний характер роботи в середовищах різної природи, таблиця 3.14. Для університетської мережі домінуючим фактором стали ресурсні та продуктивні вимоги, тоді як для мережі компанії «Колумбус» вирішальне значення мали аксіоми безпеки, рольового доступу та сегментації.

Таблиця 4.15 - Порівняльні результати застосування онтологічно-керованого методу

Мережа	Кількість кандидатних конфігурацій	Кількість конфігурацій, відсіяних аксіомами	Оптимальна конфігурація	Покращення за (J)
ЗУНУ	6	3	(K_2)	48.1%
«Колумбус»	7	4	(K_5)	50.6%

У кожному з випадків онтологічна модель дозволила звужити множину конфігурацій до допустимих рішень ще до етапу оптимізації, що зменшило число неефективних або недопустимих варіантів.

Висновки до розділу 4

1. Представлено практичну реалізацію знання-орієнтованої програмної системи управління корпоративними комп'ютерними мережами, розробленої на основі запропонованих у попередніх розділах онтологічної моделі корпоративної мережі, методу онтологічно-керованого конфігурування та математичної моделі багатокритеріального розподілу навантаження між центрами термінальних кластерів.

2. Розроблено користувацький інтерфейс програмної системи, який забезпечує інтегровану взаємодію з основними функціональними підсистемами, зокрема модулями моніторингу мережі, онтологічної бази знань, управління задачами, конфігурування ресурсів, оптимізації, логічного виведення, управління політиками безпеки та формування звітності. Запропонована структура інтерфейсу дозволяє реалізувати єдиний інформаційний простір для аналізу стану мережі, управління її ресурсами та підтримки процесів прийняття рішень.

3. Виконано програмну реалізацію онтологічної бази знань та механізмів логічного виведення, які забезпечують автоматизовану перевірку відповідності конфігурацій вимогам задач, ресурсним обмеженням та політикам безпеки. Інтеграція онтологічного представлення знань із засобами моніторингу та оптимізації дозволила реалізувати адаптивний механізм управління корпоративною мережею на основі семантичного аналізу її поточного стану.

4. Проведено експериментальні дослідження методу онтологічно-керованого конфігурування корпоративної мережі на прикладах реальних мережевих інфраструктур. Отримані результати підтвердили ефективність використання онтологічних аксіом для скорочення множини допустимих конфігурацій та підвищення якості прийнятих рішень. Зокрема, для мережі компанії «Колумбус» онтологічний механізм дозволив автоматично відсіяти 4 із 7 потенційних конфігурацій, які не відповідали політикам сегментації та вимогам безпеки.

5. Експериментальна апробація запропонованих підходів продемонструвала покращення основних показників функціонування мережевої інфраструктури. Для корпоративної мережі Західноукраїнського національного університету середня затримка доступу зменшилася на 36,4 %, частка відмов авторизації – на 72,9 %, частка конфліктів доступу до баз даних – на 71,0 %, а інтегральний критерій якості функціонування мережі покращився на 48,1 %. Для мережі компанії «Колумбус» середня затримка бізнес-сервісів зменшилася на 29,9 %, кількість порушень політик доступу скоротилася на 85,7 %, інтегральний ризик

функціонування мережі знизився на 62,8 %, а значення інтегрального критерію якості покращилося на 50,6 %.

6. Результати тестування підтвердили працездатність усіх функціональних компонентів програмної системи та їх коректну взаємодію в межах єдиної архітектури. Реалізована система забезпечує автоматизоване формування конфігураційних рішень, підтримку логічного виведення, аналіз поточного стану мережі та пошук субоптимальних рішень щодо розподілу навантаження між ресурсами корпоративної мережі.

ВИСНОВКИ

У дисертаційній роботі розв'язано науково-технічне завдання підвищення ефективності управління корпоративними комп'ютерними мережами шляхом розроблення моделей і методів на основі знання-орієнтованого підходу, що враховують структуру мережі, її ресурси та множину задач, які ними розв'язуються. При цьому отримано такі наукові та практичні результати.

1. Проведено комплексний аналіз сучасних технологій, методів і програмних засобів управління корпоративними комп'ютерними мережами, включаючи системи моніторингу, конфігурування та балансування навантаження. Встановлено, що існуючі рішення переважно базуються на локальних показниках використання ресурсів та евристичних правилах прийняття рішень, що обмежує їхню адаптивність у динамічних умовах функціонування корпоративної мережі. За результатами аналізу обґрунтовано доцільність застосування знання-орієнтованих підходів для підвищення рівня автоматизації процесів управління мережевою інфраструктурою.

2. Досліджено можливості застосування знання-орієнтованих та онтологічних підходів для підтримки процесів управління корпоративними комп'ютерними мережами. Показано, що використання формалізованого подання знань дозволяє враховувати не лише параметри мережевих ресурсів, але й семантичні зв'язки між сервісами, користувачами, мережевими ресурсами та множиною задач. Це створило теоретичне підґрунтя для розроблення інтелектуалізованих механізмів прийняття рішень та стало основою подальших наукових результатів дисертаційного дослідження.

3. Розроблено онтологічну модель корпоративної комп'ютерної мережі, яка забезпечує формалізацію знань про структуру мережі, її ресурси, сервіси та множину задач у вигляді системи взаємопов'язаних онтологічних сутностей, відношень та аксіом. На відміну від існуючих підходів до опису мережевої інфраструктури, запропонована модель враховує семантичні зв'язки між

елементами мережі та дозволяє використовувати механізми логічного виведення для підтримки процесів конфігурування й управління. Використання онтологічної моделі забезпечило підвищення обґрунтованості прийняття управлінських рішень та скорочення часу налаштування мережевих ресурсів.

4. Розроблено метод онтологічно-керованого конфігурування корпоративної комп'ютерної мережі, який базується на семантичному поданні знань про ресурси, сервіси та множину задач і використовує механізми логічного виведення для автоматизованого формування конфігураційних рішень. На відміну від традиційних підходів, запропонований метод забезпечує автоматичну перевірку сумісності ресурсів, політик доступу та станів мережевих вузлів на основі онтологічних аксіом. Результати експериментальних досліджень показали можливість автоматичного відсікання до 57 % потенційно некоректних конфігурацій ще на етапі логічної перевірки, що дозволило зменшити простір пошуку допустимих рішень та підвищити ефективність процесу конфігурування.

5. Розроблено архітектуру програмної онтологічно-керованої системи управління корпоративними комп'ютерними мережами, яка інтегрує онтологічне представлення знань про структуру мережі, її ресурси та множину задач із механізмами логічного виведення, адаптивного прийняття рішень і динамічного розподілу навантаження. Запропонована архітектура забезпечує взаємодію підсистем моніторингу, онтологічного аналізу, конфігурування та балансування навантаження в межах єдиного програмного середовища, що створює основу для побудови самоналаштовуваних корпоративних мереж.

6. Удосконалено математичну модель розподілу навантаження між центрами термінальних кластерів шляхом поєднання апарату відкритих стохастичних мереж, багатокритеріальної оптимізації та онтологічного подання знань про ресурси корпоративної мережі й множину задач. На відміну від існуючих моделей, запропонований підхід дозволяє враховувати продуктивність серверів, збалансованість використання ресурсів, якість обслуговування користувачів, вартість перенаправлення навантаження та онтологічні обмеження предметної

області. Використання методу гілок і меж разом із механізмами логічного відсікання недопустимих рішень забезпечило суттєве зменшення обчислювальної складності задачі порівняно з повним перебором та дозволило знаходити субоптимальні рішення в умовах великої кількості можливих конфігурацій мережі.

7. Проведено апробацію розробленої знання-орієнтованої програмної системи на прикладі діючих корпоративних комп'ютерних мереж Західноукраїнського національного університету та ПП «Колумбус». Результати експериментальних досліджень підтвердили ефективність запропонованих моделей і методів. Для мережі ЗУНУ середня затримка доступу до сервісів зменшилась з 151 мс до 96 мс (на 36,4 %), середнє завантаження application server знизилось з 92 % до 73 % (на 20,7 %), частка відмов авторизації скоротилась з 4,8 % до 1,3 % (на 72,9 %), а інтегральний критерій якості покращився з 0,81 до 0,42 (на 48,1 %). Для мережі ПП «Колумбус» середня затримка бізнес-сервісу зменшилась з 87 мс до 61 мс (на 29,9 %), середнє завантаження серверів прикладних сервісів скоротилось з 88 % до 67 % (на 23,9 %), кількість порушень політик доступу зменшилась з 7 до 1 випадку на добу (на 85,7 %), інтегральний ризик функціонування мережі знизився на 62,8 %, а інтегральний критерій якості конфігурації покращився на 50,6 %. Отримані результати підтверджують практичну ефективність розробленої знання-орієнтованої системи управління корпоративними комп'ютерними мережами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Allemang, Dean, and James Hendler. 2020. *Semantic Web for the Working Ontologist*. 3rd ed. Amsterdam: Elsevier. doi:10.1016/C2018-0-01984-9.
2. Alshaer, Ehab, and Hamed Hamed. 2004. "Modeling and Management of Firewall Policies." *IEEE Transactions on Network and Service Management* 1 (1): 2–10. doi:10.1109/TNSM.2004.840431.
3. Andročec, Darko. 2025. "Using Large Language Models for Ontology Development." *Proceedings* 104 (1): 9. doi:10.3390/proceedings2025104009.
4. Antoniou, Grigoris, and Frank van Harmelen. 2008. *A Semantic Web Primer*. 2nd ed. Cambridge, MA: MIT Press. doi:10.7551/mitpress/9780262018289.001.0001.
5. Atencio, Edison, Mauro Mancini, and Guillermo Bustos. 2025. "Ontology-Based Integration of Enterprise Architecture and Project Management: A Systems Thinking Approach for Project-Based Organizations in the Architecture, Engineering, and Construction Sector." *Systems* 13 (6): 477. doi:10.3390/systems13060477.
6. Ayoubi, Sara, Noura Limam, Mohamed Ali Salahuddin, Nashid Shahriar, Felix Estrada-Solano, and Raouf Boutaba. 2018. "Machine Learning for Cognitive Network Management." *IEEE Communications Magazine* 56 (1): 158–165. doi:10.1109/MCOM.2018.1700609.
7. Bai, L., X. Zhang, Y. Chen, and J. Li. 2026. "A Deep Learning-Based Multi-Objective Optimization Method for Resource-Constrained Scheduling Problems." *Applied Sciences* 16 (5): 2260. doi:10.3390/app16052260.
8. Benson, Theophilus, Aditya Akella, and David A. Maltz. 2009. "Unraveling the Complexity of Network Management." *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation*. doi:10.5555/1558977.1558983.

9. Benzekki, Kamal, Abdeslam El Fergougui, and Abdelbaki Elbelrhiti Elalaoui. 2016. "Software-Defined Networking (SDN): A Survey." *Security and Communication Networks* 9 (18): 5803–5833. doi:10.1002/sec.1737.
10. Berners-Lee, Tim, James Hendler, and Ora Lassila. 2001. "The Semantic Web." *Scientific American* 284 (5): 34–43. doi:10.1038/scientificamerican0501-34.
11. Bjorklund, Martin. 2010. "YANG—A Data Modeling Language for the Network Configuration Protocol (NETCONF)." RFC 6020. doi:10.17487/RFC6020.
12. Bordes, Antoine, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. "Translating Embeddings for Modeling Multi-Relational Data." *Advances in Neural Information Processing Systems* 26. doi:10.5555/2999792.2999923.
13. Boutaba, Raouf, Mohammad A. Salahuddin, Noura Limam, Sara Ayoubi, Nashid Shahriar, Felipe Estrada-Solano, and Oscar M. Caicedo. 2018. "A Comprehensive Survey on Machine Learning for Networking: Evolution, Applications and Research Opportunities." *Journal of Internet Services and Applications* 9: 16. doi:10.1186/s13174-018-0087-2.
14. Boutaba, Raouf, Mohammad Salahuddin, Noura Limam, Sara Ayoubi, Nashid Shahriar, Felipe Estrada-Solano, and Oscar Caicedo. 2018. "Machine Learning for Networking: Evolution and Research Opportunities." *Journal of Internet Services and Applications* 9 (1): 16. doi:10.1186/s13174-018-0087-2.
15. Caballero, Víctor, Jose L. Pérez, and Antonio F. Skarmeta. 2019. "Ontology-Defined Middleware for Internet of Things Architectures." *Sensors* 19 (5): 1163. doi:10.3390/s19051163.
16. Cappelli, M. A., F. De Angelis, and L. Greco. 2025. "Methodological Exploration of Ontology Generation with a Large Language Model." *Electronics* 14 (14): 2863. doi:10.3390/electronics14142863.
17. Chen, Yanjun, Min Zhou, Meizhou Zhang, and Meng Zha. 2025. "Knowledge-Graph-Driven Fault Diagnosis Methods for Intelligent Production Lines." *Sensors* 25 (13): 3912. doi:10.3390/s25133912.

18. Clark, David D., Craig Partridge, J. Christopher Ramming, and John T. Wroclawski. 2003. "A Knowledge Plane for the Internet." *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. doi:10.1145/863955.863957.
19. Clemm, Alexander, Lisa Ciavaglia, Lisandro Granville, and Raouf Boutaba. 2021. "Intent-Based Networking – Concepts and Definitions." *Internet Engineering Task Force Draft*. doi:10.17487/RFC9315.
20. Clemm, Alexander. 2017. *Intent-Based Networking*. Indianapolis: Cisco Press. doi:10.5860/choice.55-4453.
21. Coello Coello, Carlos A., Gary B. Lamont, and David A. Van Veldhuizen. 2007. *Evolutionary Algorithms for Solving Multi-Objective Problems*. 2nd ed. New York: Springer. doi:10.1007/978-0-387-36797-2.
22. Corcho, Oscar, Mariano Fernández-López, and Asunción Gómez-Pérez. 2003. "Methodologies, Tools and Languages for Building Ontologies." *Data & Knowledge Engineering* 46 (1): 41–64. doi:10.1016/S0169-023X(02)00195-7.
23. De Bartolomeo, M., A. D'Ambrosio, and G. De Felice. 2025. "Ontologies and Knowledge Graphs for Railway Safety: A Systematic Literature Review." *Safety* 11 (4): 100. doi:10.3390/safety11040100.
24. Deb, Kalyanmoy. 2001. *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester: Wiley. doi:10.1002/9780470743615.
25. Dettmers, Tim, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. "Convolutional 2D Knowledge Graph Embeddings." *AAAI 2018*. doi:10.1609/aaai.v32i1.11573.
26. Ding, Li, Tim Finin, Anupam Joshi, Yun Peng, Rong Pan, and Pranam Kolari. 2004. "Search on the Semantic Web." *Computer* 38 (10): 62–69. doi:10.1109/MC.2005.347.
27. Dyvak, M.; Tsapiv, Y.; Pukas, A.; Petrovskiy, Y.; Melnyk, A.; Dyvak, A.; Banasik, A.; Czupryna-Nowak, A.; Pikiwicz, P.; Popyk, Y.; Dzyha, Y. AR Technology for Restoring Upper-Limb Joint Mobility in Patients. *Appl. Sci.* 2026, 16, 5878.

28. Ehrgott, Matthias. 2005. *Multicriteria Optimization*. 2nd ed. Berlin: Springer. doi:10.1007/3-540-27659-9.
29. Ehrlinger, Lisa, and Wolfram Wöß. 2016. “Towards a Definition of Knowledge Graphs.” *SEMANTiCS 2016*. doi:10.1145/2993318.2993341.
30. Enns, Rob, Martin Bjorklund, Andy Bierman, and Juergen Schoenwaelder. 2011. “Network Configuration Protocol (NETCONF).” *RFC 6241*. doi:10.17487/RFC6241.
31. Euzenat, Jérôme, and Pavel Shvaiko. 2013. *Ontology Matching*. 2nd ed. Berlin: Springer. doi:10.1007/978-3-642-38721-0.
32. Feamster, Nick, Jennifer Rexford, and Ellen Zegura. 2014. “The Road to SDN: An Intellectual History of Programmable Networks.” *ACM Queue* 11 (12): 20–40. doi:10.1145/2602204.2602219.
33. Felder, M., J. Hoffmann, and S. Schulte. 2025. “Smart Routing for Sustainable Supply Chain Networks Using Hybrid AI-Supported Knowledge Graphs.” *Applied Sciences* 15 (14): 8001. doi:10.3390/app15148001.
34. Fensel, Dieter, Frank van Harmelen, Ian Horrocks, Deborah McGuinness, and Peter Patel-Schneider. 2001. “OIL: An Ontology Infrastructure for the Semantic Web.” *IEEE Intelligent Systems* 16 (2): 38–45. doi:10.1109/5254.920601.
35. Fernández-López, Mariano, Asunción Gómez-Pérez, and Natalia Juristo. 1997. “METHONTOLOGY: From Ontological Art Towards Ontological Engineering.” *AAAI Spring Symposium Series*.
36. Fortz, Bernard, Jennifer Rexford, and Mikkel Thorup. 2002. “Traffic Engineering with Traditional IP Routing Protocols.” *IEEE Communications Magazine* 40 (10): 118–124. doi:10.1109/MCOM.2002.1039859.
37. Gómez-Pérez, Asunción, and Oscar Corcho. 2002. “Ontology Languages for the Semantic Web.” *IEEE Intelligent Systems* 17 (1): 54–60. doi:10.1109/5254.988453.
38. Gómez-Pérez, Asunción, Mariano Fernández-López, and Oscar Corcho. 2004. *Ontological Engineering*. London: Springer. doi:10.1007/b97353.

39. Gruber, Thomas R. 1993. "A Translation Approach to Portable Ontology Specifications." *Knowledge Acquisition* 5 (2): 199–220. doi:10.1006/knac.1993.1008.
40. Guarino, Nicola. 1998. "Formal Ontology and Information Systems." *Proceedings of FOIS 1998*, 3–15.
41. Hamilton, William, Rex Ying, and Jure Leskovec. 2017. "Inductive Representation Learning on Large Graphs." *NeurIPS 2017*. doi:10.48550/arXiv.1706.02216.
42. Harris, Steve, and Andy Seaborne. 2013. "SPARQL 1.1 Query Language." *W3C Recommendation*.
43. Herrera, Francisco. 2008. "Genetic Fuzzy Systems: Taxonomy, Current Research Trends and Prospects." *Evolutionary Intelligence* 1 (1): 27–46. doi:10.1007/s12065-007-0001-5.
44. Hitzler, Pascal, Markus Krötzsch, and Sebastian Rudolph. 2010. *Foundations of Semantic Web Technologies*. Boca Raton, FL: CRC Press. doi:10.1201/9781420090518.
45. Hogan, Aidan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, et al. 2021. "Knowledge Graphs." *ACM Computing Surveys* 54 (4): 1–37. doi:10.1145/3447772.
46. Horrocks, Ian, Oliver Kutz, and Ulrike Sattler. 2006. "The Even More Irresistible SROIQ." *Proceedings of KR 2006*. doi:10.5555/1597538.1597561.
47. Horrocks, Ian, Peter Patel-Schneider, and Frank van Harmelen. 2003. "From SHIQ and RDF to OWL: The Making of a Web Ontology Language." *Journal of Web Semantics* 1 (1): 7–26. doi:10.1016/S1570-8268(03)00008-8.
48. Horrocks, Ian, Peter Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosz, and Mike Dean. 2004. "SWRL: A Semantic Web Rule Language Combining OWL and RuleML." *W3C Member Submission*.
49. Horrocks, Ian. 2008. "Ontologies and the Semantic Web." *Communications of the ACM* 51 (12): 58–67. doi:10.1145/1409360.1409377.

50. Janowicz, Krzysztof, Pascal Hitzler, Benjamin Adams, Dave Kolas, and Charles Vardeman. 2014. "Five Stars of Linked Data Vocabulary Use." *Semantic Web* 5 (3): 173–176. doi:10.3233/SW-140173.
51. Jaziri, W., M. Ben Aouicha, and A. Hadj Kacem. 2026. "Ontology Quality Improvement in the Semantic Web: A Machine-Learning-Driven Framework." *Systems* 14 (2): 154. doi:10.3390/systems14020154.
52. Ji, Shaoxiong, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2022. "A Survey on Knowledge Graphs: Representation, Acquisition, and Applications." *IEEE Transactions on Neural Networks and Learning Systems* 33 (2): 494–514. doi:10.1109/TNNLS.2021.3070843.
53. Kanzouai, Chaimae, Soukaina Bouarourou, Abderrahim Zannou, Abdelhak Boulaalam, and El Habib Nfaoui. 2025. "Enhancing IoT Scalability and Interoperability Through Ontology Alignment and FedProx." *Future Internet* 17 (4): 140. doi:10.3390/fi17040140.
54. Kim, Hyojoon, and Nick Feamster. 2013. "Improving Network Management with Software Defined Networking." *IEEE Communications Magazine* 51 (2): 114–119. doi:10.1109/MCOM.2013.6461195.
55. Kipf, Thomas, and Max Welling. 2017. "Semi-Supervised Classification with Graph Convolutional Networks." *ICLR 2017*. doi:10.48550/arXiv.1609.02907.
56. Klir, George J., and Bo Yuan. 1995. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Upper Saddle River, NJ: Prentice Hall. doi:10.5555/212890.
57. Klyne, Graham, and Jeremy Carroll. 2004. "Resource Description Framework (RDF): Concepts and Abstract Syntax." W3C Recommendation.
58. Knublauch, Holger, Matthew Horridge, Mark Musen, Alan Rector, Robert Stevens, Nick Drummond, Phillip Lord, Natalya Noy, and Julian Seidenberg. 2004. "The Protégé OWL Plugin." *Lecture Notes in Computer Science* 3298: 95–109. doi:10.1007/978-3-540-30475-3_9.

59. Kotenko, Igor, and Evgeny Doynikova. 2015. "Ontology-Based Modeling and Analysis of Security Information for Security Event Management." *Journal of Information Security and Applications* 24: 70–82. doi:10.1016/j.jisa.2015.06.002.
60. Kreutz, Diego, Fernando M. V. Ramos, and Christian Rothenberg. 2013. "Towards Secure and Dependable Software-Defined Networks." *Proceedings of HotSDN 2013*, 55–60. doi:10.1145/2491185.2491199.
61. Kreutz, Diego, Fernando M. V. Ramos, Paulo Esteves Veríssimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. 2015. "Software-Defined Networking: A Comprehensive Survey." *Proceedings of the IEEE* 103 (1): 14–76. doi:10.1109/JPROC.2014.2371999.
62. Lara, Adrian, Anisha Kolasani, and Byrav Ramamurthy. 2014. "Network Innovation Using OpenFlow: A Survey." *IEEE Communications Surveys & Tutorials* 16 (1): 493–512. doi:10.1109/SURV.2013.081313.00105.
63. Lassila, Ora, and Ralph Swick. 1999. "Resource Description Framework (RDF) Model and Syntax Specification." W3C Recommendation.
64. Liu, Wenrui, Jiale Zhu, Xiangming Li, Yichao Fei, Hai Wang, Shangdong Liu, Xiaoyao Zheng, and Yimu Ji. 2025. "Resource Scheduling Algorithm for Edge Computing Networks Based on Multi-Objective Optimization." *Applied Sciences* 15 (19): 10837. doi:10.3390/app151910837.
65. M. Pylypchuk, N. Porplytsya, I. Stasiv, R. Monko, Y. Popyk, T. Yurchyshyn, Parallelization and Optimization of a Method for Web Content Analysis and Accessibility Evaluation for Visually Impaired Users. 2025 15th International Conference on Advanced Computer Information Technologies (ACIT), Sibenik, Croatia, 2025, pp. 879-883
66. Manzano-Farray, D., J. M. Corchado, and R. S. Alonso. 2026. "Ontology-Driven and Human-Centric Digital Twins in Service Environments." *Sensors* 26 (9): 2764. doi:10.3390/s26092764.

67. Marler, R. Timothy, and Jasbir S. Arora. 2004. "Survey of Multi-Objective Optimization Methods for Engineering." *Structural and Multidisciplinary Optimization* 26 (6): 369–395. doi:10.1007/s00158-003-0368-6.
68. McGuinness, Deborah, and Frank van Harmelen. 2004. OWL Web Ontology Language Overview. W3C Recommendation. doi:10.5555/1626474.1626478.
69. Medhi, Deepankar, and Karthikeyan Ramasamy. 2017. *Network Routing: Algorithms, Protocols, and Architectures*. 2nd ed. Burlington, MA: Morgan Kaufmann. doi:10.1016/C2014-0-04148-9.
70. Medved, Jan, Robert Varga, Ales Tkacik, and Ken Gray. 2014. "OpenDaylight: Towards a Model-Driven SDN Controller Architecture." *Proceedings of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*. doi:10.1109/WoWMoM.2014.6918938.
71. Melnyk A, I. Romanets, R. Shevchuk, O. Kvasnytsia, Y. Popyk, V. Ivanyna. An Intelligent System for Detecting Malicious Messages in IP Telephony within a Corporate Network. 2025 IEEE 13th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Gliwice, Poland, 2025, pp. 1-5.
72. Melnyk A., Popyk Y. An Ontology-Driven Knowledge-Based Approach to Complex Systems Management. *Computer Systems and Information Technologies*. 2026. № 2. P. 86-99.
73. Mestres, Albert, Alberto Rodriguez-Natal, Josep Carner, Pere Barlet-Ros, Eduard Alarcón, Marc Solé, Victor Muntés-Mulero, et al. 2017. "Knowledge-Defined Networking." *ACM SIGCOMM Computer Communication Review* 47 (3): 2–10. doi:10.1145/3138808.3138810.
74. Mijumbi, Rashid, Joan Serrat, Juan-Luis Gorricho, Noura Bouten, Filip De Turck, and Raouf Boutaba. 2016. "Network Function Virtualization: State-of-the-Art and Research Challenges." *IEEE Communications Surveys & Tutorials* 18 (1): 236–262. doi:10.1109/COMST.2015.2477041.

75. Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2015. "Human-Level Control Through Deep Reinforcement Learning." *Nature* 518 (7540): 529–533. doi:10.1038/nature14236.
76. Motik, Boris, Peter Patel-Schneider, and Bijan Parsia. 2012. OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax. W3C Recommendation.
77. Musen, Mark. 2015. "The Protégé Project: A Look Back and a Look Forward." *AI Matters* 1 (4): 4–12. doi:10.1145/2757001.2757003.
78. Naganawa, H., Y. Sato, and T. Kobayashi. 2025. "Implementing a Knowledge Management System with Graph Retrieval-Augmented Generation." *Electronics* 14 (24): 4948. doi:10.3390/electronics14244948.
79. Nickel, Maximilian, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2016. "A Review of Relational Machine Learning for Knowledge Graphs." *Proceedings of the IEEE* 104 (1): 11–33. doi:10.1109/JPROC.2015.2483592.
80. Noy, Natalya F., and Michael A. Musen. 2003. "The PROMPT Suite: Interactive Tools for Ontology Merging and Mapping." *International Journal of Human-Computer Studies* 59 (6): 983–1024. doi:10.1016/S1071-5819(03)00118-7.
81. Noy, Natalya, and Deborah McGuinness. 2001. "Ontology Development 101: A Guide to Creating Your First Ontology." Stanford Knowledge Systems Laboratory Technical Report.
82. Nunes, Bruno Astuto A., Marc Mendonca, Xuan-Nam Nguyen, Katia Obraczka, and Thierry Turetletti. 2014. "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks." *IEEE Communications Surveys & Tutorials* 16 (3): 1617–1634. doi:10.1109/SURV.2014.012214.00180.
83. O'Connor, Martin, and Amar Das. 2009. "SQWRL: A Query Language for OWL." *Proceedings of OWLED 2009*. doi:10.1007/978-3-642-17746-0_11.

84. Obrst, Leo, Patrick Chase, and Robert Markeloff. 2012. "Developing an Ontology of the Cyber Security Domain." *Proceedings of STIDS 2012*. doi:10.48550/arXiv.1202.3333.
85. Pan, Jeff Z., Guido Vetere, José Manuel Gómez-Pérez, and Honghan Wu, eds. 2017. "Exploiting Linked Data and Knowledge Graphs in Large Organisations." Springer. doi:10.1007/978-3-319-45654-6.
86. Pan, Shirui, Ruiqi Hu, Sai-Fu Fung, Guodong Long, Jing Jiang, and Chengqi Zhang. 2024. "Trustworthy Graph Neural Networks: Aspects, Methods and Trends." *Artificial Intelligence Review* 57 (4): 91. doi:10.1007/s10462-023-10547-0.
87. Paulheim, Heiko. 2017. "Knowledge Graph Refinement: A Survey of Approaches and Evaluation Methods." *Semantic Web* 8 (3): 489–508. doi:10.3233/SW-160218.
88. Popyk Y., Ples M. Optimization of Energy Consumption in Data Centers Based on an Ontological Approach. *Opto-Electronic Devices and Components in Laser and Energy Technologies*. 2025, № 50 (2), 282-292.
89. Prud'hommeaux, Eric, and Andy Seaborne. 2008. "SPARQL Query Language for RDF." W3C Recommendation.
90. Russell, Stuart, and Peter Norvig. 2021. *Artificial Intelligence: A Modern Approach*. 4th ed. Hoboken, NJ: Pearson. doi:10.5555/3086952.
91. Schlichtkrull, Michael, Thomas Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. "Modeling Relational Data with Graph Convolutional Networks." *ESWC 2018*. doi:10.1007/978-3-319-93417-4_38.
92. Schoenwaelder, Juergen, Martin Bjorklund, and Phil Shafer. 2021. "Network Configuration and Management Automation with YANG." *IEEE Communications Magazine* 59 (1): 96–101. doi:10.1109/MCOM.001.2000444.
93. Schreiber, Guus, Yves Raimond, and Wouter Beek. 2020. "RDF 1.1 Primer." W3C Recommendation.

94. Segin A. I., Popyk Y. I. System of Automatic Translation of Verbal Audio Streams in Real Time. Materials of the 17th International Symposium of Croatian Metallurgical Society SHMD'2024 (published in: *Metalurgija* 63 (2024) 2). P. 317 -320.
95. Silver, David, Aja Huang, Chris Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, et al. 2016. "Mastering the Game of Go with Deep Neural Networks and Tree Search." *Nature* 529 (7587): 484–489. doi:10.1038/nature16961.
96. Smolira, P., M. S. M. Khan, and A. B. L. D. Ferreira. 2025. "Scoped Review and Evaluation of Ontologies in Operation and Maintenance of Civil Infrastructure." *Buildings* 16 (1): 81. doi:10.3390/buildings16010081.
97. Staab, Steffen, and Rudi Studer, eds. 2010. *Handbook on Ontologies*. 2nd ed. Berlin: Springer. doi:10.1007/978-3-540-92673-3.
98. Staab, Steffen, Hans-Peter Schnurr, Rudi Studer, and York Sure. 2001. "Knowledge Processes and Ontologies." *IEEE Intelligent Systems* 16 (1): 26–34. doi:10.1109/5254.920598.
99. Stallings, William. 2020. *Network Security Essentials: Applications and Standards*. 7th ed. Hoboken, NJ: Pearson. doi:10.5555/3460520.
100. Strassner, John. 2004. *Policy-Based Network Management*. San Francisco: Morgan Kaufmann. doi:10.1016/B978-155860859-8/50002-8.
101. Studer, Rudi, V. Richard Benjamins, and Dieter Fensel. 1998. "Knowledge Engineering: Principles and Methods." *Data & Knowledge Engineering* 25 (1–2): 161–197. doi:10.1016/S0169-023X(97)00056-6.
102. Sutton, Richard S., and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*. 2nd ed. Cambridge, MA: MIT Press. doi:10.1109/TNN.1998.712192.
103. Taurshia, Antony, Jaspher W. Kathrine, J. Andrew, and Jennifer Eunice R. 2024. "Securing Internet of Things Applications Using Software-Defined Network-Aided Group Key Management with a Modified One-Way Function Tree." *Applied Sciences* 14 (6): 2405. doi:10.3390/app14062405.

104. Undercoffer, Jeffrey, Anupam Joshi, and John Pinkston. 2003. "Modeling Computer Attacks: An Ontology for Intrusion Detection." *Recent Advances in Intrusion Detection*. doi:10.1007/978-3-540-45248-5_7.
105. Uschold, Mike, and Michael Gruninger. 1996. "Ontologies: Principles, Methods and Applications." *The Knowledge Engineering Review* 11 (2): 93–136. doi:10.1017/S0269888900007797.
106. Vahdat, Amirhossein, Thierry Badard, and Jacynthe Pouliot. 2025. "Development of an Ontology-Based Framework to Enhance Geospatial Data Discovery and Selection in Geoportals for Natural-Hazard Early Warning Systems." *ISPRS International Journal of Geo-Information* 14 (10): 369. doi:10.3390/ijgi14100369.
107. Verma, Dinesh. 2002. *Simplifying Network Administration Using Policy-Based Management*. IEEE Press. doi:10.1002/047122458X.
108. Wang, Quan, Zhendong Mao, Bin Wang, and Li Guo. 2017. "Knowledge Graph Embedding: A Survey of Approaches and Applications." *IEEE Transactions on Knowledge and Data Engineering* 29 (12): 2724–2743. doi:10.1109/TKDE.2017.2754499.
109. Wijesekara, Patikiri Arachchige Don Shehan Nilmantha, and Subodha Gunawardena. 2023. "A Comprehensive Survey on Knowledge-Defined Networking." *Telecom* 4 (3): 477–596. doi:10.3390/telecom4030025.
110. Wu, Zonghan, Shirui Pan, Guodong Long, Jing Jiang, Chengqi Zhang, and Yanhong Yu. 2021. "A Comprehensive Survey on Graph Neural Networks." *IEEE Transactions on Neural Networks and Learning Systems* 32 (1): 4–24. doi:10.1109/TNNLS.2020.2978386.
111. Xie, Yunpeng, Xiaoyao Huang, Jingchun Li, and Tianhe Liu. 2023. "Computing Power Network: Multi-Objective Optimization-Based Routing." *Sensors* 23 (15): 6702. doi:10.3390/s23156702.
112. Xu, Yongnan, Xiangrong Tang, Linyu Huang, Hamid Ullah, and Qian Ning. 2025. "Multi-Objective Optimization for Resource Allocation in Space–Air–

Ground Network with Diverse IoT Devices.” *Sensors* 25 (1): 274. doi:10.3390/s25010274.

113. Zadeh, Lotfi A. 1965. “Fuzzy Sets.” *Information and Control* 8 (3): 338–353. doi:10.1016/S0019-9958(65)90241-X.

114. Zhang, Chuxu, Huaxiu Yao, Chao Huang, Mengxiao Hu, and Zhiwei Liu. 2023. “Deep Learning on Graphs: A Survey.” *IEEE Transactions on Knowledge and Data Engineering* 35 (1): 249–270. doi:10.1109/TKDE.2021.3070214.

115. Zuo, J., Y. Wang, and H. Li. 2025. “Construction of Journal Knowledge Graph Based on Deep Learning and Its Application in Question Answering.” *Electronics* 14 (9): 1728. doi:10.3390/electronics14091728.

116. Мельник А. М., Попик Ю. І. Програмна архітектура онтологічно керованої системи для управління складними об’єктами. *Вісник Національного університету водного господарства та природокористування. Серія «Технічні науки»*. 2026. Вип. 1(125). С. 276-302.

117. Попик Ю.І. Знання-орієнтована програмна система управління корпоративними комп’ютерними мережами. *Матеріали школи-семінару молодих вчених і студентів «Комп’ютерні інформаційні технології» (СІТ’2026)*. Тернопіль. 2026, ст. 190-191 (0,1 д.а.).

118. Сегін А. І., Заставний О. М., Попик Ю. І. Модель передбачення географічного місцеположення і часу виникнення землетрусів. *Комп’ютерне моделювання та програмне забезпечення інформаційних систем і технологій (КМПЗ-2024): збірник наукових праць (тези доповідей та вибрані статті) IV Міжнародної науково-практичної конференції, Чернівці, 30 травня – 01 червня 2024 р. Львів: ЛНУ імені Івана Франка, 2024. С. 137-141.*

119. Сегін А. І., Когут І. Р., Когут Ю., Павловський А. О., Возняк В. С. Кореляційні моделі в полярній системі координат // *Збірник матеріалів проблемно-наукової міжгалузевої конференції «Інформаційні проблеми комп’ютерних систем, юриспруденції, енергетики, моделювання та управління» (ICSM-2023)*. Надвірна, 2023. С. 188-191.

ДОДАТОК А

ФРАГМЕНТ OWL-ОНТОЛОГІЇ КОРПОРАТИВНОЇ КОМП'ЮТЕРНОЇ МЕРЕЖІ

```

<?xml version="1.0"?>
<rdf:RDF xmlns="http://www.semanticweb.org/corporate-network#"
  xml:base="http://www.semanticweb.org/corporate-network"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

  <owl:Ontology rdf:about="http://www.semanticweb.org/corporate-network">
    <rdfs:comment>
      OWL-онтологія корпоративної комп'ютерної мережі для знання-
      орієнтованої системи управління мережевими ресурсами, сервісами та задачами.
    </rdfs:comment>
  </owl:Ontology>

  <!-- КЛАСИ ОНТОЛОГІЇ -->

  <owl:Class rdf:about="#NetworkResource"/>
  <owl:Class rdf:about="#Server">
    <rdfs:subClassOf rdf:resource="#NetworkResource"/>
  </owl:Class>
  <owl:Class rdf:about="#ApplicationServer">
    <rdfs:subClassOf rdf:resource="#Server"/>
  </owl:Class>
  <owl:Class rdf:about="#DatabaseServer">
    <rdfs:subClassOf rdf:resource="#Server"/>
  </owl:Class>
  <owl:Class rdf:about="#AuthenticationServer">
    <rdfs:subClassOf rdf:resource="#Server"/>
  </owl:Class>
  <owl:Class rdf:about="#TerminalClusterCenter">
    <rdfs:subClassOf rdf:resource="#NetworkResource"/>
  </owl:Class>
  <owl:Class rdf:about="#Router">
    <rdfs:subClassOf rdf:resource="#NetworkResource"/>
  </owl:Class>
  <owl:Class rdf:about="#Switch">
    <rdfs:subClassOf rdf:resource="#NetworkResource"/>

```

```

</owl:Class>

<owl:Class rdf:about="#NetworkService"/>
<owl:Class rdf:about="#LMSService">
  <rdfs:subClassOf rdf:resource="#NetworkService"/>
</owl:Class>
<owl:Class rdf:about="#DatabaseService">
  <rdfs:subClassOf rdf:resource="#NetworkService"/>
</owl:Class>
<owl:Class rdf:about="#AuthenticationService">
  <rdfs:subClassOf rdf:resource="#NetworkService"/>
</owl:Class>

<owl:Class rdf:about="#User"/>
<owl:Class rdf:about="#Administrator">
  <rdfs:subClassOf rdf:resource="#User"/>
</owl:Class>
<owl:Class rdf:about="#Employee">
  <rdfs:subClassOf rdf:resource="#User"/>
</owl:Class>
<owl:Class rdf:about="#Student">
  <rdfs:subClassOf rdf:resource="#User"/>
</owl:Class>

<owl:Class rdf:about="#NetworkTask"/>
<owl:Class rdf:about="#Configuration"/>
<owl:Class rdf:about="#SecurityPolicy"/>
<owl:Class rdf:about="#NetworkState"/>
<owl:Class rdf:about="#OptimizationCriterion"/>

<!-- ОБ'ЄКТИ ВЛАСТИВОСТІ -->

<owl:ObjectProperty rdf:about="#usesService">
  <rdfs:domain rdf:resource="#NetworkTask"/>
  <rdfs:range rdf:resource="#NetworkService"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="#hostedOn">
  <rdfs:domain rdf:resource="#NetworkService"/>
  <rdfs:range rdf:resource="#NetworkResource"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="#requiresResource">
  <rdfs:domain rdf:resource="#NetworkTask"/>

```

```

    <rdfs:range rdf:resource="#NetworkResource"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="#assignedTo">
  <rdfs:domain rdf:resource="#NetworkTask"/>
  <rdfs:range rdf:resource="#NetworkResource"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="#hasPolicy">
  <rdfs:domain rdf:resource="#NetworkResource"/>
  <rdfs:range rdf:resource="#SecurityPolicy"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="#hasState">
  <rdfs:domain rdf:resource="#NetworkResource"/>
  <rdfs:range rdf:resource="#NetworkState"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="#includedInConfiguration">
  <rdfs:domain rdf:resource="#NetworkResource"/>
  <rdfs:range rdf:resource="#Configuration"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="#solvesTask">
  <rdfs:domain rdf:resource="#User"/>
  <rdfs:range rdf:resource="#NetworkTask"/>
</owl:ObjectProperty>

<!-- DATA PROPERTIES -->

<owl:DatatypeProperty rdf:about="#hasCPU">
  <rdfs:domain rdf:resource="#NetworkResource"/>
  <rdfs:range rdf:resource="xsd:double"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasRAM">
  <rdfs:domain rdf:resource="#NetworkResource"/>
  <rdfs:range rdf:resource="xsd:double"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasBandwidth">
  <rdfs:domain rdf:resource="#NetworkResource"/>
  <rdfs:range rdf:resource="xsd:double"/>
</owl:DatatypeProperty>

```

```
<owl:DatatypeProperty rdf:about="#hasLoad">
  <rdfs:domain rdf:resource="#NetworkResource"/>
  <rdfs:range rdf:resource="xsd:double"/>
</owl:DatatypeProperty>
```

```
<owl:DatatypeProperty rdf:about="#hasPriority">
  <rdfs:domain rdf:resource="#NetworkTask"/>
  <rdfs:range rdf:resource="xsd:integer"/>
</owl:DatatypeProperty>
```

```
<owl:DatatypeProperty rdf:about="#hasResponseTime">
  <rdfs:domain rdf:resource="#NetworkService"/>
  <rdfs:range rdf:resource="xsd:double"/>
</owl:DatatypeProperty>
```

```
<!-- ИНДИВИДУИ -->
```

```
<owl:NamedIndividual rdf:about="#EDU_SRV_01">
  <rdf:type rdf:resource="#ApplicationServer"/>
  <hasCPU rdf:datatype="xsd:double">16</hasCPU>
  <hasRAM rdf:datatype="xsd:double">32</hasRAM>
  <hasBandwidth rdf:datatype="xsd:double">1000</hasBandwidth>
  <hasLoad rdf:datatype="xsd:double">0.74</hasLoad>
</owl:NamedIndividual>
```

```
<owl:NamedIndividual rdf:about="#EDU_SRV_02">
  <rdf:type rdf:resource="#ApplicationServer"/>
  <hasCPU rdf:datatype="xsd:double">24</hasCPU>
  <hasRAM rdf:datatype="xsd:double">64</hasRAM>
  <hasBandwidth rdf:datatype="xsd:double">1000</hasBandwidth>
  <hasLoad rdf:datatype="xsd:double">0.61</hasLoad>
</owl:NamedIndividual>
```

```
<owl:NamedIndividual rdf:about="#EDU_DB_01">
  <rdf:type rdf:resource="#DatabaseServer"/>
  <hasCPU rdf:datatype="xsd:double">16</hasCPU>
  <hasRAM rdf:datatype="xsd:double">64</hasRAM>
  <hasBandwidth rdf:datatype="xsd:double">1000</hasBandwidth>
  <hasLoad rdf:datatype="xsd:double">0.58</hasLoad>
</owl:NamedIndividual>
```

```
<owl:NamedIndividual rdf:about="#AUTH_SRV_01">
  <rdf:type rdf:resource="#AuthenticationServer"/>
```

```

    <hasCPU rdf:datatype="xsd:double">8</hasCPU>
    <hasRAM rdf:datatype="xsd:double">16</hasRAM>
    <hasBandwidth rdf:datatype="xsd:double">1000</hasBandwidth>
    <hasLoad rdf:datatype="xsd:double">0.41</hasLoad>
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#NormalState">
  <rdf:type rdf:resource="#NetworkState"/>
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#HighLoadState">
  <rdf:type rdf:resource="#NetworkState"/>
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#LMS">
  <rdf:type rdf:resource="#LMSService"/>
  <hostedOn rdf:resource="#EDU_SRV_02"/>
  <hasResponseTime rdf:datatype="xsd:double">96</hasResponseTime>
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#Database">
  <rdf:type rdf:resource="#DatabaseService"/>
  <hostedOn rdf:resource="#EDU_DB_01"/>
  <hasResponseTime rdf:datatype="xsd:double">72</hasResponseTime>
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#Authentication">
  <rdf:type rdf:resource="#AuthenticationService"/>
  <hostedOn rdf:resource="#AUTH_SRV_01"/>
  <hasResponseTime rdf:datatype="xsd:double">54</hasResponseTime>
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#Task_LMS_Access">
  <rdf:type rdf:resource="#NetworkTask"/>
  <usesService rdf:resource="#LMS"/>
  <requiresResource rdf:resource="#EDU_SRV_02"/>
  <hasPriority rdf:datatype="xsd:integer">1</hasPriority>
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="#Admin_01">
  <rdf:type rdf:resource="#Administrator"/>
  <solvesTask rdf:resource="#Task_LMS_Access"/>
</owl:NamedIndividual>

```

<!-- ПРИКЛАДИ ЛОГІЧНИХ ОБМЕЖЕНЬ У ВИГЛЯДІ КОМЕНТАРІВ -->

<owl:Axiom>

<rdfs:comment>

Якщо задача використовує сервіс, а сервіс розміщено на певному ресурсі,
то цей ресурс є кандидатом для конфігурації задачі.

</rdfs:comment>

</owl:Axiom>

<owl:Axiom>

<rdfs:comment>

Якщо ресурс має стан HighLoadState, то він не повинен включатися
до оптимальної конфігурації для нової задачі.

</rdfs:comment>

</owl:Axiom>

</rdf:RDF>

ДОДАТОК Б
ЛІСТИНГ КОДУ РЕАЛІЗАЦІЇ МЕТОДУ ОНТОЛОГІЧНО-КЕРОВАНОВОГО
КОНФІГУРУВАННЯ КОРПОРАТИВНОЇ МЕРЕЖІ

```
package ua.edu.wunu.network.configuration;

import java.util.*;
import java.util.stream.Collectors;

enum NodeState {
    NORMAL,
    WARNING,
    HIGH_LOAD,
    OFFLINE
}

class NetworkTask {

    private String id;
    private String name;
    private String serviceType;
    private int priority;

    private double requiredCpu;
    private double requiredRam;
    private double requiredBandwidth;

    public NetworkTask(
        String id,
        String name,
        String serviceType,
        int priority,
        double requiredCpu,
        double requiredRam,
        double requiredBandwidth) {

        this.id = id;
        this.name = name;
        this.serviceType = serviceType;
        this.priority = priority;

        this.requiredCpu = requiredCpu;
        this.requiredRam = requiredRam;
        this.requiredBandwidth = requiredBandwidth;
    }

    public String getId() {
        return id;
    }

    public String getServiceType() {
        return serviceType;
    }

    public double getRequiredCpu() {
        return requiredCpu;
    }
}
```

```

    }

    public double getRequiredRam() {
        return requiredRam;
    }

    public double getRequiredBandwidth() {
        return requiredBandwidth;
    }

    public int getPriority() {
        return priority;
    }
}

class NetworkResource {

    private String id;

    private double cpuCapacity;
    private double ramCapacity;
    private double bandwidthCapacity;

    private double cpuLoad;
    private double ramLoad;
    private double bandwidthLoad;

    private NodeState state;

    private Set<String> supportedServices;
    private Set<String> securityPolicies;

    public NetworkResource(
        String id,
        double cpuCapacity,
        double ramCapacity,
        double bandwidthCapacity,
        double cpuLoad,
        double ramLoad,
        double bandwidthLoad,
        NodeState state,
        Set<String> supportedServices,
        Set<String> securityPolicies) {

        this.id = id;

        this.cpuCapacity = cpuCapacity;
        this.ramCapacity = ramCapacity;
        this.bandwidthCapacity = bandwidthCapacity;

        this.cpuLoad = cpuLoad;
        this.ramLoad = ramLoad;
        this.bandwidthLoad = bandwidthLoad;

        this.state = state;

        this.supportedServices = supportedServices;
        this.securityPolicies = securityPolicies;
    }
}

```

```

public String getId() {
    return id;
}

public double getAvailableCpu() {
    return cpuCapacity - cpuLoad;
}

public double getAvailableRam() {
    return ramCapacity - ramLoad;
}

public double getAvailableBandwidth() {
    return bandwidthCapacity - bandwidthLoad;
}

public NodeState getState() {
    return state;
}

public Set<String> getSupportedServices() {
    return supportedServices;
}

public Set<String> getSecurityPolicies() {
    return securityPolicies;
}
}

class OntologyKnowledgeBase {

    private List<NetworkResource> resources;

    public OntologyKnowledgeBase() {
        resources = new ArrayList<>();
    }

    public void addResource(NetworkResource resource) {
        resources.add(resource);
    }

    public List<NetworkResource> getResources() {
        return resources;
    }

    public List<NetworkResource> findResourcesByService(
        String serviceType) {

        return resources.stream()
            .filter(r ->
                r.getSupportedServices()
                    .contains(serviceType))
            .collect(Collectors.toList());
    }
}

class OntologyReasoner {

    public boolean checkResourceAvailability(
        NetworkTask task,

```

```

        NetworkResource resource) {

    return resource.getAvailableCpu()
        >= task.getRequiredCpu()
        &&
        resource.getAvailableRam()
        >= task.getRequiredRam()
        &&
        resource.getAvailableBandwidth()
        >= task.getRequiredBandwidth();
}

public boolean checkServiceSupport(
    NetworkTask task,
    NetworkResource resource) {

    return resource.getSupportedServices()
        .contains(task.getServiceType());
}

public boolean checkSecurityPolicy(
    NetworkTask task,
    NetworkResource resource) {

    String requiredPolicy =
        "ALLOW_" + task.getServiceType();

    return resource.getSecurityPolicies()
        .contains(requiredPolicy);
}

public boolean checkNodeState(
    NetworkResource resource) {

    return resource.getState()
        == NodeState.NORMAL;
}

public boolean validateTaskPlacement(
    NetworkTask task,
    NetworkResource resource) {

    return checkResourceAvailability(task, resource)
        &&
        checkServiceSupport(task, resource)
        &&
        checkSecurityPolicy(task, resource)
        &&
        checkNodeState(resource);
}
}

class ConfigurationCandidate {

    private NetworkTask task;
    private NetworkResource resource;

    private double criterionValue;

    public ConfigurationCandidate(

```

```

        NetworkTask task,
        NetworkResource resource,
        double criterionValue) {

    this.task = task;
    this.resource = resource;
    this.criterionValue = criterionValue;
}

public NetworkResource getResource() {
    return resource;
}

public double getCriterionValue() {
    return criterionValue;
}

@Override
public String toString() {

    return "Task=" + task.getId()
        + ", Resource="
        + resource.getId()
        + ", Criterion="
        + criterionValue;
}
}

class MultiCriteriaOptimizer {

    private static final double ALPHA = 0.4;
    private static final double BETA = 0.3;
    private static final double GAMMA = 0.3;

    public double evaluate(
        NetworkTask task,
        NetworkResource resource) {

        double cpuCriterion =
            task.getRequiredCpu()
                / resource.getAvailableCpu();

        double ramCriterion =
            task.getRequiredRam()
                / resource.getAvailableRam();

        double bandwidthCriterion =
            task.getRequiredBandwidth()
                / resource.getAvailableBandwidth();

        return ALPHA * cpuCriterion
            +
            BETA * ramCriterion
            +
            GAMMA * bandwidthCriterion;
    }

    public ConfigurationCandidate selectOptimal(
        List<ConfigurationCandidate> candidates) {

```

```

        return candidates.stream()
            .min(
                Comparator.comparingDouble(
                    ConfigurationCandidate::
                        getCriterionValue))
            .orElse(null);
    }
}

public class OntologyDrivenConfigurator {

    private OntologyKnowledgeBase knowledgeBase;

    private OntologyReasoner reasoner;

    private MultiCriteriaOptimizer optimizer;

    public OntologyDrivenConfigurator(
        OntologyKnowledgeBase knowledgeBase) {

        this.knowledgeBase = knowledgeBase;

        this.reasoner = new OntologyReasoner();

        this.optimizer =
            new MultiCriteriaOptimizer();
    }

    public ConfigurationCandidate configure(
        NetworkTask task) {

        List<NetworkResource> resources =
            knowledgeBase.findResourcesByService(
                task.getServiceType());

        List<ConfigurationCandidate>
            candidates = new ArrayList<>();

        for (NetworkResource resource : resources) {

            boolean valid =
                reasoner.validateTaskPlacement(
                    task,
                    resource);

            if (!valid) {
                continue;
            }

            double criterion =
                optimizer.evaluate(
                    task,
                    resource);

            candidates.add(
                new ConfigurationCandidate(
                    task,
                    resource,
                    criterion));
        }
    }
}

```

```

        return optimizer.selectOptimal(
            candidates);
    }

    public static void main(String[] args) {

        OntologyKnowledgeBase kb =
            new OntologyKnowledgeBase();

        kb.addResource(
            new NetworkResource(
                "TS-01",
                32,
                64,
                1000,
                10,
                20,
                150,
                NodeState.NORMAL,
                Set.of(
                    "DATABASE",
                    "WEB",
                    "ERP"),
                Set.of(
                    "ALLOW_DATABASE",
                    "ALLOW_WEB",
                    "ALLOW_ERP")
            ));

        kb.addResource(
            new NetworkResource(
                "TS-02",
                24,
                48,
                1000,
                15,
                30,
                300,
                NodeState.NORMAL,
                Set.of(
                    "WEB",
                    "ERP"),
                Set.of(
                    "ALLOW_WEB",
                    "ALLOW_ERP")
            ));

        NetworkTask task =
            new NetworkTask(
                "TASK-001",
                "Accounting System",
                "ERP",
                1,
                4,
                8,
                100
            );

        OntologyDrivenConfigurator configurator =

```

```
        new OntologyDrivenConfigurator(kb);  
    ConfigurationCandidate result =  
        configurator.configure(task);  
  
    System.out.println(  
        "Selected configuration:");  
  
    System.out.println(result);  
    }  
}
```

ДОДАТОК В

ЛІСТИНГ КОДУ ОСНОВНИХ МОДУЛІВ СИСТЕМИ

```
import numpy as np
```

A.1. Модуль управління онтологічною базою знань (OntologyManager.java)

Призначення: забезпечує завантаження, збереження та оновлення онтології корпоративної комп'ютерної мережі.

Основні функції:

```
public class OntologyManager {  
  
    public void loadOntology(String filePath);  
  
    public void saveOntology(String filePath);  
  
    public void addResource(NetworkResource resource);  
  
    public void addService(NetworkService service);  
  
    public void addTask(NetworkTask task);  
  
    public List<NetworkResource> getResources();  
  
    public List<NetworkTask> getTasks();  
}
```

A.2. Модуль логічного виведення (OntologyReasoner.java)

Призначення: реалізує перевірку аксіом та правил предметної області й формує нові знання про стан мережі.

Основні функції:

```
public class OntologyReasoner {  
  
    public boolean checkResourceAvailability(  
        NetworkTask task,  
        NetworkResource resource);  
}
```

```

public boolean checkSecurityPolicy(
    NetworkTask task,
    NetworkResource resource);

public boolean validateTaskPlacement(
    NetworkTask task,
    NetworkResource resource);

public List<InferenceResult> infer();
}

```

A.3. Модуль онтологічно-керованого конфігурування (OntologyDrivenConfigurator.java)

Призначення: реалізує метод онтологічно-керованого конфігурування корпоративної мережі.

Основні функції:

```

public class OntologyDrivenConfigurator {

    public ConfigurationCandidate configure(
        NetworkTask task);

    public List<ConfigurationCandidate>
        generateConfigurations(
            NetworkTask task);

    public boolean applyConfiguration(
        ConfigurationCandidate configuration);
}

```

A.4. Модуль моніторингу мережевої інфраструктури (NetworkMonitoringService.java)

Призначення: забезпечує збір даних про стан мережевих ресурсів та сервісів.

Основні функції:

```

public class NetworkMonitoringService {

    public ResourceMetrics collectMetrics();

    public double getCpuLoad(String nodeId);
}

```

```

public double getMemoryUsage(String nodeId);

public double getNetworkTraffic(String nodeId);

public NetworkState analyzeState();
}

```

A.5. Модуль розподілу навантаження між центрами термінальних кластерів (LoadBalancingService.java)

Призначення: реалізує математичну модель багатокритеріального розподілу навантаження.

Основні функції:

```

public class LoadBalancingService {

    public AllocationResult optimizeAllocation(
        List<Task> tasks,
        List<ClusterCenter> centers);

    public double calculateCriterion(
        Task task,
        ClusterCenter center);

    public AllocationResult
        selectOptimalConfiguration();
}

```

A.6. Модуль підтримки прийняття рішень (DecisionSupportService.java)

Призначення: формує рекомендації щодо реконфігурування мережі на основі результатів логічного виведення та оптимізації.

Основні функції:

```

public class DecisionSupportService {

    public Recommendation generateRecommendation();

    public RiskAssessment assessRisk();

    public List<ActionPlan> generateActionPlan();
}

```

```
}
```

A.7. Модуль управління політиками безпеки (SecurityPolicyService.java)

Призначення: забезпечує перевірку відповідності конфігурацій політикам безпеки.

Основні функції:

```
public class SecurityPolicyService {  
  
    public boolean validatePolicy(  
        Configuration configuration);  
  
    public List<Violation>  
        detectViolations();  
  
    public SecurityReport generateReport();  
}
```

A.8. Модуль генерації звітів та журналів (ReportingService.java)

Призначення: формує звіти про стан корпоративної мережі та результати роботи системи.

Основні функції:

```
public class ReportingService {  
  
    public Report generateNetworkReport();  
  
    public Report generateLoadReport();  
  
    public Report generateSecurityReport();  
  
    public void exportToPDF(String fileName);  
}
```

A.9. Контролер користувацького інтерфейсу (DashboardController.java)

Призначення: забезпечує взаємодію між вебінтерфейсом та серверною частиною системи.

Основні функції:

```
@RestController
```

```

@RequestMapping("/dashboard")
public class DashboardController {

    @GetMapping("/topology")
    public TopologyDto getTopology();

    @GetMapping("/monitoring")
    public MonitoringDto getMonitoring();

    @GetMapping("/ontology")
    public OntologyDto getOntology();

    @PostMapping("/optimize")
    public OptimizationResult optimize();
}

```

A.10. Модуль доступу до даних (NetworkRepository.java)

Призначення: забезпечує взаємодію з базою даних PostgreSQL.

Основні функції:

```

@Repository
public interface NetworkRepository
    extends JpaRepository<NetworkNode, Long> {

    List<NetworkNode> findByStatus(String status);

    List<NetworkNode> findByType(String type);

    List<NetworkNode> findByCluster(String clusterId);
}

```

ДОДАТОК Г
СПИСОК ПУБЛІКАЦІЙ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

1. Dyvak, M.; Tsapiv, Y.; Pukas, A.; Petrovskyi, Y.; Melnyk, A.; Dyvak, A.; Banasik, A.; Czupryna-Nowak, A.; Pikiiewicz, P.; Popyk, Y.; Dzyha, Y. AR Technology for Restoring Upper-Limb Joint Mobility in Patients. *Appl. Sci.* 2026, 16, 5878.
2. Melnyk A., Popyk Y. An Ontology-Driven Knowledge-Based Approach to Complex Systems Management. *Computer Systems and Information Technologies.* 2026. № 2. P. 86-99.
3. Мельник А. М., Попик Ю. І. Програмна архітектура онтологічно керованої системи для управління складними об'єктами. *Вісник Національного університету водного господарства та природокористування. Серія «Технічні науки».* 2026. Вип. 1(125). С. 276-302.
4. Popyk Y., Ples M. Optimization of Energy Consumption in Data Centers Based on an Ontological Approach. *Opto-Electronic Devices and Components in Laser and Energy Technologies.* 2025, № 50 (2), 282-292.
5. M. Pylypchuk, N. Porplytsya, I. Stasiv, R. Monko, Y. Popyk, T. Yurchyshyn, Parallelization and Optimization of a Method for Web Content Analysis and Accessibility Evaluation for Visually Impaired Users. 2025 15th International Conference on Advanced Computer Information Technologies (ACIT), Sibenik, Croatia, 2025, pp. 879-883
6. A. Melnyk, I. Romanets, R. Shevchuk, O. Kvasnytsia, Y. Popyk, V. Ivanyna. An Intelligent System for Detecting Malicious Messages in IP Telephony within a Corporate Network. 2025 IEEE 13th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Gliwice, Poland, 2025, pp. 1-5.
7. Попик Ю.І. Знання-орієнтована програмна система управління корпоративними комп'ютерними мережами. Матеріали школи-семінару молодих вчених і студентів «Комп'ютерні інформаційні технології» (СІТ'2026). Тернопіль. 2026, ст. 190-191 (0,1 д.а.).

8. Segin A. I., Popyk Y. I. System of Automatic Translation of Verbal Audio Streams in Real Time. Materials of the 17th International Symposium of Croatian Metallurgical Society SHMD'2024 (published in: Metalurgija 63 (2024) 2). P. 317 -320.

9. Сегін А. І., Заставний О. М., Попик Ю. І. Модель передбачення географічного місцеположення і часу виникнення землетрусів. Комп'ютерне моделювання та програмне забезпечення інформаційних систем і технологій (КМПЗ-2024): збірник наукових праць (тези доповідей та вибрані статті) IV Міжнародної науково-практичної конференції, Чернівці, 30 травня – 01 червня 2024 р. Львів: ЛНУ імені Івана Франка, 2024. С. 137-141.

10. Сегін А. І., Когут І. Р., Когут Ю., Павловський А. О., Возняк В. С. Кореляційні моделі в полярній системі координат // Збірник матеріалів проблемно-наукової міжгалузевої конференції «Інформаційні проблеми комп'ютерних систем, юриспруденції, енергетики, моделювання та управління» (ICSM-2023). Надвірна, 2023. С. 188-191.

ДОДАТОК Д

АКТИ ВПРОВАДЖЕНЬ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОГО ДОСЛІДЖЕННЯ



46020 м. Тернопіль
вул. Київська, 2
тел. (0352) 51-18-71
e-mail: office@columbus.in.ua

ЄДРПОУ 21150086
р/р UA 7138 0805 0000 0000 2600 5582 758
в АТ Райффайзен Банк Аваль,
МФО 380805

АКТ

про впровадження результатів дисертаційної роботи
Попика Юрія Ігоровича на тему:
“Знання-орієнтована програмна система управління корпоративними
комп’ютерними мережами”

Даний акт складений в тому, що результати дисертаційної роботи Попика Ю.І. «Знання-орієнтована програмна система управління корпоративними комп’ютерними мережами» апробовано та впроваджено у корпоративній мережі ПП «КОЛУМБУС», зокрема:

1. Онтологічна модель корпоративної комп’ютерної мережі, яка формується на основі структурованого представлення знань про мережеві ресурси, сервіси, користувачів, політики доступу та множину задач, що виконуються в мережевому середовищі. Дана онтологічна модель сприяє підвищенню ефективності використання ресурсів корпоративної мережі, скороченню часу адміністрування та покращенню якості обслуговування користувачів.

2. Метод онтологічно-керованого конфігурування корпоративної комп’ютерної мережі базується на використанні онтологічної бази знань, механізмів логічного виведення та алгебри кортежів для формування допустимих конфігурацій мережевих ресурсів. Запропонований метод дозволяє автоматизовано аналізувати відповідність ресурсів вимогам задач, перевіряти виконання політик доступу, оцінювати поточні стани мережевих вузлів та формувати раціональні конфігураційні рішення відповідно до визначених критеріїв ефективності.

Впровадження розробленої знання-орієнтованої програмної системи управління корпоративними комп’ютерними мережами забезпечує підвищення ефективності використання мережевих ресурсів, покращення якості обслуговування користувачів, автоматизацію процесів конфігурування та підтримки прийняття рішень.

Цей акт не є підставою для фінансових взаєморозрахунків.

Директор ПП «Колумбус»



Зазуля Н.Г.

АКТ

про впровадження результатів дисертаційної роботи
 Попка Юрія Ігоровича на тему:
 “Знання-орієнтована програмна система управління корпоративними
 комп’ютерними мережами ”

Даний акт складений в тому, що результати дисертаційної роботи Попка Ю.І. «Знання-орієнтована програмна система управління корпоративними комп’ютерними мережами» апробовано та впроваджено у Центрі інформаційних технологій Західноукраїнського національного університету, зокрема:

1. Онтологічну модель корпоративної комп’ютерної мережі, яка забезпечує структуроване представлення знань про мережеві ресурси, сервіси, користувачів, політики доступу та множину задач, що виконуються в мережевому середовищі. Використання онтологічної моделі забезпечує централізоване управління знаннями про мережеву інфраструктуру, підвищує ефективність використання ресурсів та покращує якість обслуговування користувачів.

2. Метод онтологічно-керованого конфігурування корпоративної комп’ютерної мережі, який базується на використанні онтологічної бази знань, механізмів логічного виведення та алгебри кортежів для автоматизованого формування конфігурацій мережевих ресурсів. Метод забезпечує аналіз відповідності ресурсів вимогам задач, контроль дотримання політик доступу, оцінювання поточного стану мережевих вузлів та підтримку прийняття рішень щодо конфігурування мережі.

Впровадження розробленої знання-орієнтованої програмної системи управління корпоративними комп’ютерними мережами забезпечує підвищення ефективності використання мережевих ресурсів, покращення якості обслуговування користувачів, автоматизацію процесів конфігурування та адміністрування мережевої інфраструктури, а також підтримку прийняття управлінських рішень на основі онтологічної бази знань і механізмів логічного виведення. Це сприяє скороченню часу виконання операцій з управління мережею, підвищенню надійності її функціонування та зменшенню навантаження на адміністраторів корпоративної мережі.

Директор
 Центру інформаційних технологій



ІГОР РОМАНЕЦЬ



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

вул. Львівська, 11, м. Тернопіль, 46009; тел./факс +380 (352) 51-75-75;
www.wunu.edu.ua; rektor@wunu.edu.ua; ідентифікаційний код за ЄДРПОУ 33680120

№ 126-27/1551

15 червня 2026 р.



ЗАТВЕРДЖУЮ

Проректор з наукової роботи
Західноукраїнського національного
університету

Микола ДИВАК
06 2026 р.

АКТ

про використання результатів дисертаційного дослідження
Полика Юрія Ігоровича на тему «Знання-орієнтована програмна система
управління корпоративними комп'ютерними мережами»

Комісія у складі декана факультету комп'ютерних інформаційних технологій Якименка І.З. (голова комісії), начальника науково-дослідної частини Семанюк В.З., професора кафедри інформаційно-обчислювальних систем і управління Комара М.П. склала цей акт про те, що результати дисертаційної роботи Полика Ю.І. використано під час виконання науково-дослідних робіт факультету комп'ютерних інформаційних технологій Західноукраїнського національного університету за безпосередньої участі автора, а саме:

- 1) «Мультифункціональне Фермерство / Multifunctionality Farming (MyFarm) 101140288 ERASMUS-EDU-2023-PI-ALL-INNO-EDU-ENTERP;
- 2) «TALKING HEADS / Спілкування з мистецтвом минулого» № 101174331-CREA-CULT-2024-COOP-2.

Голова комісії:
Декан факультету комп'ютерних
інформаційних технологій

Ігор ЯКИМЕНКО

Члени комісії:
Начальник науково-дослідної частини

Віта СЕМАНЮК

Професор кафедри
інформаційно-обчислювальних
систем і управління

Мирослав КОМАР



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

НАУКОВО-ДОСЛІДНА ЧАСТИНА

46009, Україна, м. Тернопіль, вул. Львівська, буд. 5А, тел. (0352) 51-75-52

№ 21/ 122-2026

«08» 06 2026 року

ДОВІДКА
про участь у виконанні науково-дослідної роботи

Видана **ПОПИКУ Юрію Ігоровичу** про участь у 2026 році на посаді молодшого наукового співробітника у підготовці держбюджетної науково-дослідної роботи, яка виконується Західноукраїнським національним університетом на тему «Інформаційна технологія забезпечення ефективної акумуляції електроенергії гібридною електростанцією» (державний реєстраційний номер 0126U002209).

Начальник
науково-дослідної частини



Віта СЕМАНЮК



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

вул. Львівська, 11, м. Тернопіль, 46009; тел./факс +380 (352) 51-75-75;
www.wunu.edu.ua; rektor@wunu.edu.ua; ідентифікаційний код за ЄДРПОУ 33680120

126-27/1515

10 серпня 2026

ЗАТВЕРДЖУЮ

Проректор з науково-педагогічної роботи
Західноукраїнського національного
університету, к.е.н., доцент

Віктор ОСТРОВЕРХОВ

АКТ

про впровадження в освітній процес Західноукраїнського національного
університету результатів дисертаційної роботи

Попика Юрія Ігоровича

на тему «Знання-орієнтована програмна система управління корпоративними
комп'ютерними мережами»

Даний акт складений про те, що результати дисертаційної роботи здобувача ступеня доктора філософії Попика Юрія Ігоровича на тему «Знання-орієнтована програмна система управління корпоративними комп'ютерними мережами» використані в освітньому процесі кафедри комп'ютерних наук факультету комп'ютерних інформаційних технологій, зокрема під час підготовки студентів спеціальності F2 (121) – Інженерія програмного забезпечення. При викладанні дисципліни «Архітектура та проєктування програмного забезпечення» розглядається архітектура програмної онтологічно-керованої системи управління корпоративними комп'ютерними мережами. Можливості застосування знання-орієнтованих та онтологічних підходів для підтримки процесів управління корпоративними комп'ютерними мережами та їх онтологічні моделі вивчаються на практичних заняттях курсу «Web-онтології».

Декан факультету комп'ютерних
інформаційних технологій,
д.т.н., доцент

Ігор ЯКИМЕНКО

Завідувач кафедри комп'ютерних наук,
д.т.н., професор

Андрій ПУКАС