

УДК 004.8:004.41

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

Мельник А.М. ^[1; ORCID ID: 0000-0001-7799-9877],

д.т.н., професор,

Попик Ю.І. ^[1],

аспірант

¹Західноукраїнський національний університет, м. Тернопіль

ПРОГРАМНА АРХІТЕКТУРА ОНТОЛОГІЧНО-КЕРОВАНОЇ СИСТЕМИ ДЛЯ УПРАВЛІННЯ СКЛАДНИМИ ОБ'ЄКТАМИ

У статті розглянуто проблему побудови програмної архітектури онтологічно-керованої системи управління складними об'єктами, орієнтованої на забезпечення ефективного, адаптивного та пояснюваного функціонування в умовах динамічних навантажень і гетерогенності ресурсів. Актуальність дослідження зумовлена зростанням складності сучасних інформаційно-комунікаційних інфраструктур, зокрема університетських корпоративних мереж, які характеризуються високою інтенсивністю користувацьких запитів, варіативністю профілів навантаження та необхідністю забезпечення гарантованих показників якості обслуговування.

Запропоновано інтегрований підхід, що поєднує онтологічне моделювання, математичні методи аналізу та оптимізації, а також програмні механізми інтеграції в межах єдиної багаторівневої архітектури. Розроблено формалізовану модель онтологічно-керованої системи, яка забезпечує узгоджене представлення знань предметної області, динамічних даних і процедур прийняття рішень. Запропоновано динамічну модель розподілу навантаження для центрів термінальних кластерів, що враховує часові характеристики потоків запитів, структурні особливості мережі та багатокритеріальні обмеження.

Обґрунтовано програмну архітектуру системи, яка включає рівні збору та попередньої обробки даних, онтологічного представлення знань, семантичної обробки, аналітико-оптимізаційного аналізу, інтеграції та представлення результатів. Особливістю архітектури є інтеграція декларативного (онтологічного) та процедурного (алгоритмічного) рівнів, що

забезпечує семантичну інтерпретацію станів системи, використання механізмів логічного виведення та формування обґрунтованих керуючих рішень.

Показано, що поєднання знання-орієнтованих підходів із математичним моделюванням дозволяє підвищити ефективність розподілу навантаження, зменшити затримки обробки запитів і забезпечити збалансоване використання обчислювальних та комунікаційних ресурсів. Запропонований підхід орієнтований на інтеграцію в існуючі системи управління мережами та може бути використаний як основа для побудови інтелектуальних систем управління ІКТ-інфраструктурами.

Перспективи подальших досліджень пов'язані з розширенням моделей за рахунок врахування невизначеності, інтеграцією методів машинного навчання та розробленням цифрових двійників складних систем.

Ключові слова: онтологія, програмна архітектура, складні системи, розподіл навантаження, корпоративні мережі, знання-орієнтовані системи, семантичні технології, адаптивне управління.

Вступ. Сучасні складні об'єкти управління, до яких належать кіберфізичні системи, енергетичні комплекси, корпоративні інформаційні мережі та інфраструктури закладів вищої освіти, характеризуються високим рівнем структурної складності, гетерогенністю компонентів, динамічністю станів і наявністю невизначеностей різної природи. Управління такими об'єктами потребує застосування методів, здатних забезпечити не лише обробку великих обсягів даних, але й інтерпретацію знань, адаптацію до змін середовища та обґрунтованість прийнятих рішень [1,2].

Традиційні підходи до побудови систем управління, які базуються на жорстко заданих алгоритмах і статичних моделях, обмежені у своїй здатності ефективно функціонувати в умовах динамічних змін, складних взаємозв'язків між компонентами та неповноті інформації. У зв'язку з цим актуальним є розвиток знання-орієнтованих підходів, що передбачають використання формалізованих моделей предметної області для підтримки процесів аналізу, прогнозування та прийняття рішень [3,4].

Одним із найбільш перспективних напрямів у цьому контексті є застосування онтологічного моделювання, яке забезпечує

формалізоване представлення знань про структуру об'єкта, його властивості, відношення та обмеження у вигляді логічно узгодженої системи. Онтології дозволяють інтегрувати різноманітні джерела даних, забезпечувати семантичну інтероперабельність компонентів системи та створювати основу для реалізації механізмів логічного виведення. Це, у свою чергу, відкриває можливості для побудови інтелектуальних систем управління з підвищеним рівнем пояснюваності та адаптивності [5,6].

Водночас ефективне управління складними об'єктами вимагає поєднання символічних (онтологічних) підходів із математичними моделями, які дозволяють кількісно оцінювати стан системи, прогнозувати її поведінку та оптимізувати використання ресурсів. Інтеграція знання-орієнтованих механізмів із методами математичного моделювання та аналізу продуктивності формує основу для створення гібридних систем управління, здатних забезпечувати як формальну коректність, так і практичну ефективність функціонування [7,8].

Особливої актуальності зазначені підходи набувають у контексті управління комп'ютерними мережами університетів, які є складними інформаційно-комунікаційними системами з високою інтенсивністю запитів, різноманітністю сервісів і значною варіативністю навантаження. Зростання обсягів цифрових освітніх ресурсів, впровадження дистанційного навчання, використання хмарних сервісів і збільшення кількості користувачів зумовлюють необхідність розробки нових методів управління, орієнтованих на забезпечення ефективності, надійності та збалансованого використання ресурсів [9-11].

Постановка проблеми. Розроблення програмної архітектури онтологічно-керованих систем управління складними об'єктами постає як багатовимірною науково-технічною проблемою, що перебуває на перетині теорії знань, розподілених обчислень, мережевих технологій та методів математичного моделювання. Її складність зумовлена необхідністю синхронного узгодження двох принципово різних парадигм — декларативної (семантичної), яка оперує формалізованими знаннями у вигляді онтологій та аксіоматичних структур, і процедурної, що реалізує алгоритмічні механізми аналізу, оптимізації та прийняття рішень у динамічному середовищі [10,11].

У сучасних умовах функціонування університетських 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 [6,11].

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 користувачьких 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 поточкових 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 (АВох) 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 дослідження 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.

З огляду на зазначене, актуальним є створення архітектурних рішень, що дозволяють не лише підвищити ефективність управління складними об'єктами, але й забезпечити їх інтелектуалізацію за рахунок інтеграції знання-орієнтованих механізмів, здатних до адаптації, пояснення та прогнозування поведінки системи.

Моделювання динамічного розподілу навантаження центрів термінальних кластерів. 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 [11,12].

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 мережеских 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.

Для розв'язання зазначених проблем моделювання динамічного розподілу навантаження між центрами термінальних кластерів набуває статусу ключового наукового напряму. 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].

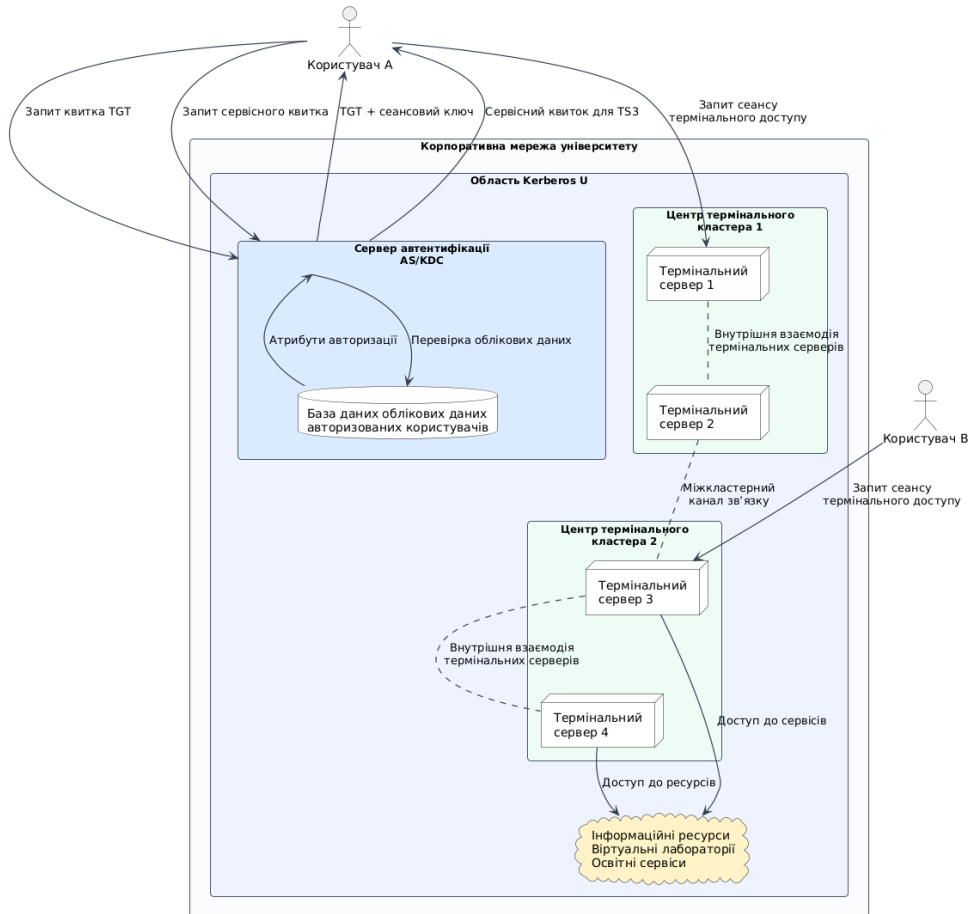


Рис. 1. Архітектура корпоративної мережі університету з центрами термінальних кластерів та механізмом автентифікації на основі Kerberos

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:

$$Us_i \cap Us_j = \emptyset, i \neq j \quad (1)$$

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

$$Us = \bigcup_{i=1}^n Us_i = \bigcup_{i=1}^n \{usr_{ki}\} = sn_i, \sum_{i=1}^n sn_i = u, u > n, \quad (2)$$



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 Tum_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:

$$\lambda_i = \alpha_i \lambda_0 \quad (3)$$

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 $QSS St_j, j = 1, \dots, sn$, 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 $\alpha_i, i = 1, \dots, sn$, яка має єдиний розв'язок:

$$\alpha_i = \frac{\sum_{kj=1}^{sn} \lambda_{kii}}{\sum_{j=1}^{sn} \sum_{k=j=1}^{sn} \lambda_{kii}} + \sum_{i=1}^{sn} \alpha_i \frac{\lambda_{ii}}{\lambda_i}, j = \overline{1, sn} \quad (4)$$

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

$$Tum_0 = \sum_{i=1}^{sn} \alpha_i t_i, \quad (5)$$

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:

$$x(k) = \langle x_1(k), \dots, x_i(k), \dots, x_{sn}(k) \rangle, \quad (6)$$

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} і з 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 (2) і (4), отримуємо таку оптимізаційну модель:

$$Tum_k = \sum_{j=1}^{sn} \alpha_i^{(k)t_j(k)} \rightarrow \min k = \overline{1, SN}, \quad (7)$$

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

$$\alpha_i(k) = \left[\frac{\sum_{k_j=1}^{sn} \lambda_{kji} + \sum_{i=1}^{sn} \sum_{k_i=1}^{sn} \lambda_{ki} X_i^{(k)} X_j^{(k)}}{\sum_{j=1}^n \sum_{k_j=1}^{nj} \lambda_{kj}} + (1 - x_i^{(k)}) \right] + \sum_{i=1}^{sn} \alpha_i^{(k)} \frac{\lambda_{ij}}{\lambda_i} (1 - x_i^{(k)} + x_j^{(k)}) \quad (8)$$

$$j = \overline{1, sn}, k = \overline{1, SN} \quad (9)$$

Середній час обслуговування для відповідних станів визначається як:

$$t_j^{(k)} = \frac{(1-x_i^{(k)})}{\mu - \left\{ \sum_{kj=1}^{sn} \lambda_{kj} + \sum_{i=1}^{sn} \left[\sum_{ki=1}^{sn} \lambda_{ki} x_{ij}^{(k)} + \lambda_{ij} (1-x_i^{(k)}) \right] \right\}} \quad (10)$$

$$I = \overline{1, sn}, j = \overline{1, sn}, k = \overline{1, SN} \quad (11)$$

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

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

$$\sum_{i=1}^{sn} \sum_{k=1}^{sn} \sum_{j=1}^{sn} c_{kj} x_{ij}^{(k)} x_i^{(k)} \leq C, k = \overline{1, SN} \quad (12)$$

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

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

Обмеження (12) 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 (8)–(12) 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 (8)–(12), 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:

$$\theta(n, m) = C_n^m (n - m) \cdot (n - m) \cdots (n - m) = C_n^m (n, m)^m. \quad (13)$$

Поширюючи цю формулу на всі значення m , де $1 \leq m \leq n - 1$, отримуємо обчислювальну складність моделі (9)–(12):

$$u(n) = \sum_{m=1}^{n-1} C_n^m (n - m)^m. \quad (14)$$

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 (9)–(12), ґрунтується на методі гілок і меж та враховує специфічну структуру досліджуваної задачі.

Введемо такі позначення:

$$I_k = \{i \mid x_i^{(k)} = 1\} J_k = \{i \mid x_i^{(k)} = 0\} \quad (15)$$

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 : $X_{j_1}, X_{j_2}, X_{j_l} \mid J_k$.

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 J_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 вибране. Така оцінка вважається допустимою лише за умови виконання таких умов здійсненності обмежень:

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

$$\sum_{j=1}^{jk} \sum_{m=1}^1 V_j + \min \sum_{m=1+1}^{Ik} \sum_{j=1}^{jk} V_i X_m^{(k)} X_{mj}^{(k)} < V_j^{\max} - V_j, j = \overline{1, J_k}, \quad (16)$$

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

На основі наведених формул алгоритм розв'язання оптимізаційної задачі формується таким чином (рис. 2).

Крок 0. Ініціалізація.

Крок 1. Введення вхідних даних. Задаються початкові параметри: кількість серверів n ; вартості перемикачів користувачів із сервера St_i на сервер St_j , C_{ij} , $i, j = 1, \dots, sn$; максимальні обсяги пам'яті обладнання, на якому розміщені сервери, V_j^{\max} , $j = 1, \dots, sn$; 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

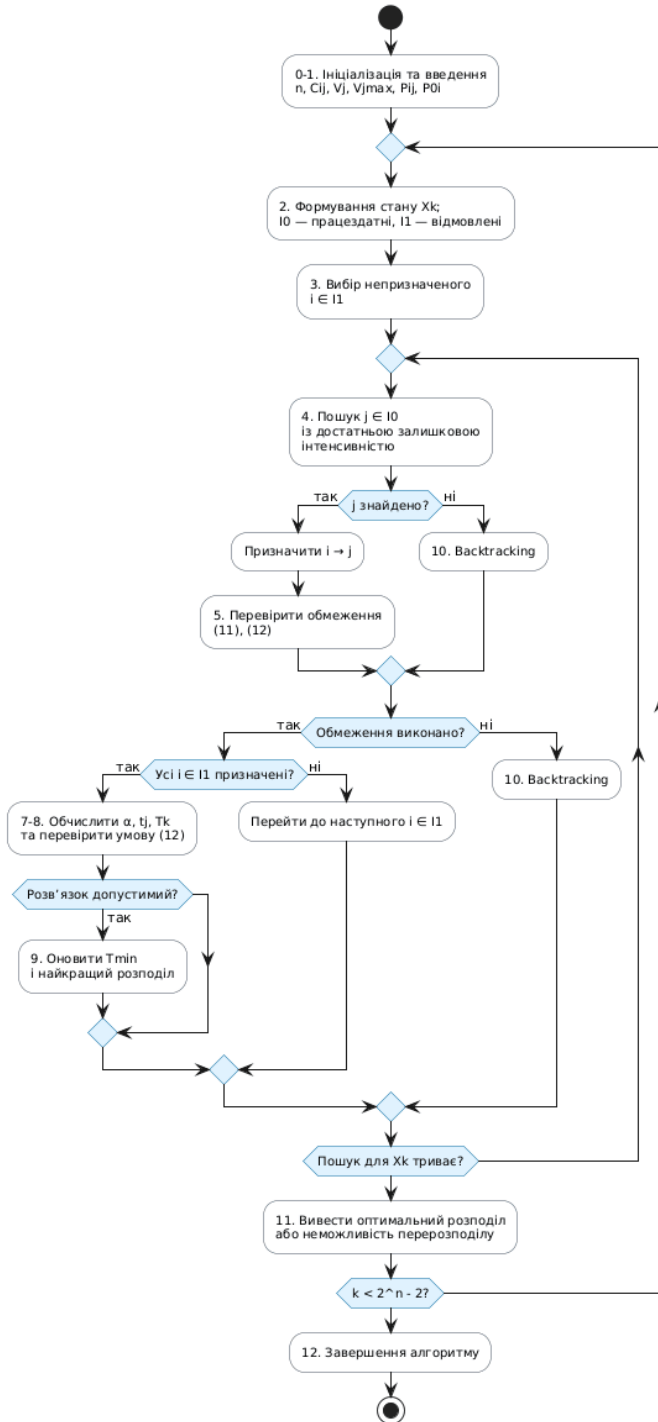


Рис 2. Алгоритм перерозподілу користувачів відмовлених серверів.

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 (11) і (12) для поточного часткового призначення. Якщо хоча б одне з обмежень порушується, алгоритм переходить до кроку 10; в іншому випадку виконання продовжується.

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 (11) 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 (7) і (10). Оскільки знаменник у формулі (12) задовольняє умову збіжності, ітераційна процедура є збіжною.

Крок 8. Перевірка умови (12). Якщо умова (12) 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.

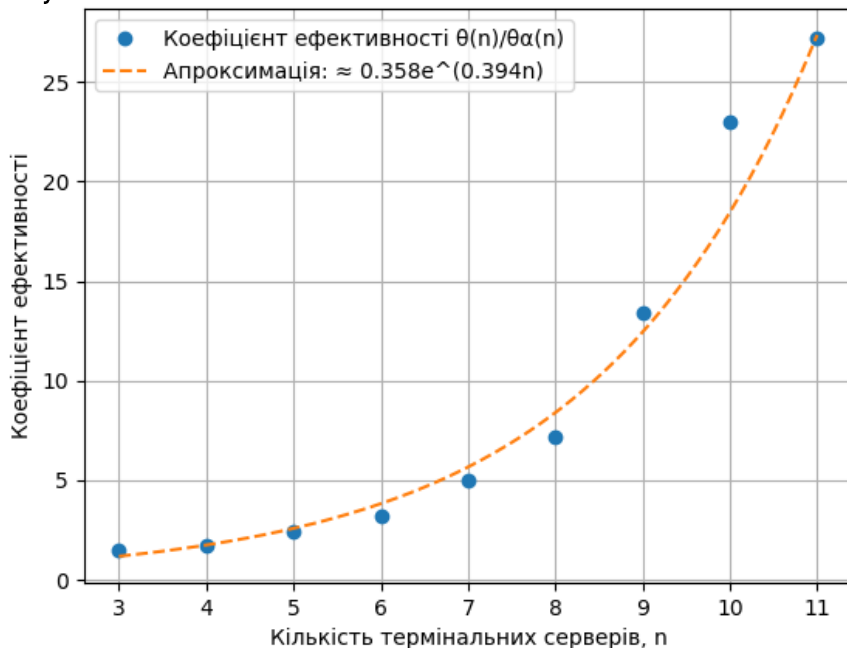


Рис. 3. Експоненційне зростання коефіцієнта ефективності $\theta(n)/\theta_\alpha(n)$ залежно від кількості термінальних серверів з апроксимаційною кривою.

Програмна архітектура онтологічно-керованої системи управління складними об'єктами розглядається як багаторівнева інтеграційна структура, що забезпечує узгоджену взаємодію між джерелами даних, онтологічною базою знань, математичними моделями, алгоритмами оптимізації та зовнішніми системами управління. У контексті університетської корпоративної мережі така архітектура орієнтована на підтримку прийняття управлінських рішень щодо динамічного розподілу навантаження між центрами термінальних кластерів, запобігання перевантаженням і забезпечення стабільних показників якості обслуговування.

Особливість запропонованої архітектури полягає у поєднанні двох взаємодоповнювальних рівнів: декларативного, який формує семантичне представлення знань про об'єкт управління, та процедурного, який реалізує алгоритмічні процедури аналізу, прогнозування, оптимізації та виконання керуючих дій. Декларативний рівень забезпечує опис понять предметної області, зв'язків між ними, обмежень і правил логічного виведення. Процедурний рівень відповідає за обробку поточних даних, виконання математичних моделей, оцінювання стану мережі та формування рішень щодо балансування навантаження.

Загальну архітектуру системи доцільно подати у вигляді сукупності взаємопов'язаних функціональних рівнів, рисунок 4:

$$SA = \{LD, LK, LS, LA, LI, LP\}, \quad (17)$$

де LD — рівень збору та попередньої обробки даних; LK — рівень онтологічного представлення знань; LS — рівень семантичної обробки та логічного виведення; LA — аналітико-оптимізаційний рівень; LI — інтеграційний рівень; LP — рівень представлення результатів і взаємодії з користувачем.

Рівень збору даних є вхідною частиною архітектури та забезпечує отримання інформації про поточний стан університетської корпоративної мережі. До таких даних належать параметри завантаженості вузлів, інтенсивність користувацьких запитів, затримки обробки, пропускна здатність каналів, кількість активних сесій, стан серверних ресурсів, показники доступності сервісів і журнали подій.

Функціонально цей рівень виконує такі задачі: агрегацію мережевих метрик, фільтрацію шумових або неповних даних, нормалізацію показників, часову синхронізацію подій та підготовку

даних до подальшої семантичної інтерпретації. Його призначення полягає не лише у технічному зборі інформації, а й у формуванні структурованого потоку спостережень, придатного для подальшого відображення на елементи онтологічної моделі.

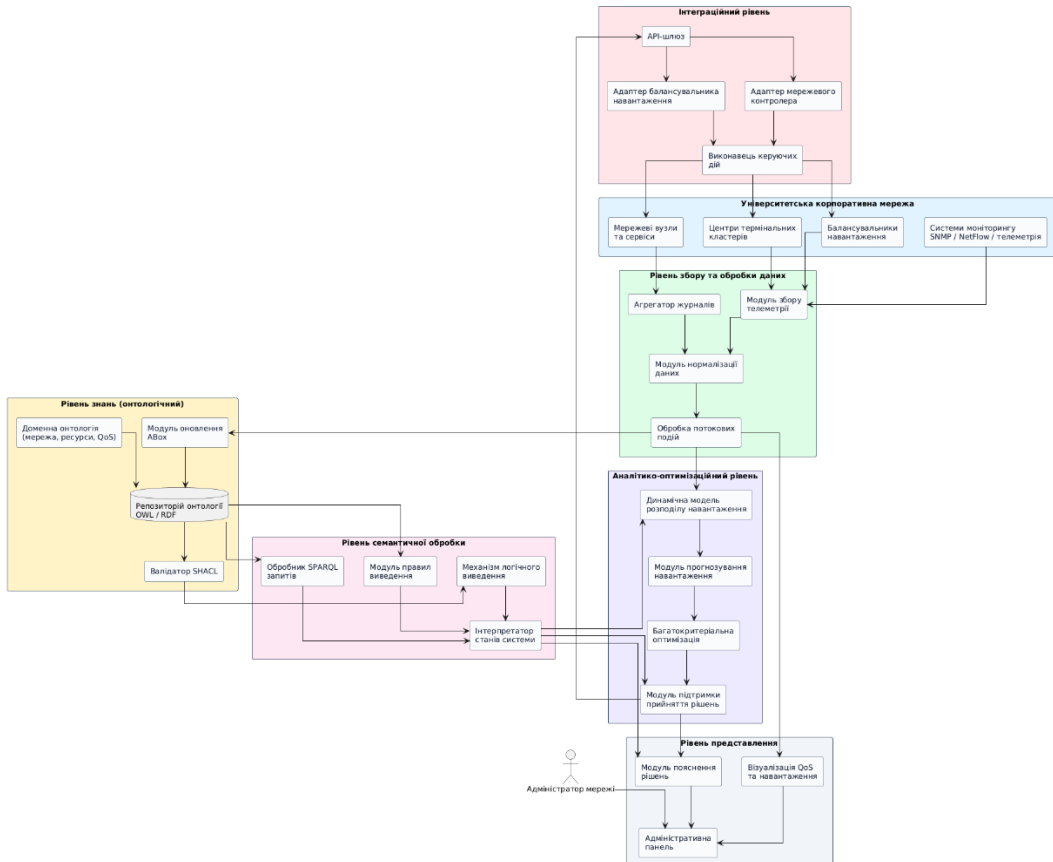


Рис. 4. Діаграма компонентів онтологічно-керованої системи управління корпоративною мережею

На цьому рівні можуть використовуватися дані з систем моніторингу, мережевої телеметрії, журналів серверів, балансувальників навантаження, контролерів віртуалізації та засобів управління контейнеризованими сервісами. У результаті формується множина спостережень:

$$D(t) = \{d_1(t), d_2(t), \dots, d_n(t)\}, \quad (18)$$

де $d_i(t)$ — окремий параметр стану мережі або кластерного вузла в момент часу t .

Рівень онтологічного представлення знань. Рівень знань є центральним компонентом архітектури, оскільки саме він забезпечує формалізацію предметної області та створює семантичну основу для прийняття рішень. Онтологія корпоративної мережі описує основні класи об'єктів, їх властивості, відношення, обмеження та правила інтерпретації станів системи.

У межах цього рівня доцільно виділити такі основні класи:

$$C = \{NetworkNode, TerminalCluster, UserRequest, Service, Resource, LoadState, QoSMetric, ControlAction\}. \quad (19)$$

Клас *NetworkNode* описує мережеві або обчислювальні вузли; *TerminalCluster* — центри термінальних кластерів; *UserRequest* — запити користувачів; *Service* — цифрові сервіси університету; *Resource* — обчислювальні та комунікаційні ресурси; *LoadState* — поточний або прогнозований стан навантаження; *QoSMetric* — показники якості обслуговування; *ControlAction* — можливі керуючі дії.

Онтологія виконує не лише описову, а й регулятивну функцію. Вона визначає допустимі стани системи, зв'язки між типами запитів і ресурсами, умови виникнення перевантаження, обмеження на перенаправлення запитів, а також правила вибору альтернативних вузлів для обробки навантаження. Наприклад, якщо певний термінальний кластер перебуває у стані критичного навантаження, а суміжний вузол має достатній резерв ресурсів, онтологічні правила можуть ініціювати рекомендацію щодо перерозподілу запитів.

Формально онтологічний рівень можна подати як:

$$Onet = \langle C, R, P, A, I \rangle, \quad (20)$$

де C — множина класів; R — множина відношень між класами; P — множина властивостей; A — множина аксіом і обмежень; I — множина індивідів, що відповідають конкретним елементам мережевої інфраструктури.

Рівень семантичної обробки та логічного виведення.

Семантичний рівень забезпечує перехід від сирих технічних даних до інтерпретованих знань про стан системи. Його функція полягає у зіставленні поточних значень параметрів із поняттями онтології, виявленні прихованих залежностей, перевірці узгодженості станів і формуванні нових знань на основі правил.

На цьому рівні виконуються такі операції: семантична аотація вхідних даних; оновлення онтології відповідно до поточного стану

мережі; перевірка логічної консистентності; виконання SPARQL-запитів; застосування правил виведення; формування семантичних подій, які передаються до аналітико-оптимізаційного рівня.

Наприклад, технічний показник завантаження процесора на рівні 92 %, кількість активних сесій понад допустиме порогове значення та збільшення середньої затримки відповіді можуть бути семантично інтерпретовані як стан:

$$LoadState(n_j, t) = Critical. \quad (21)$$

Така інтерпретація є принципово важливою, оскільки система працює не лише з числовими значеннями, а з узагальненими семантичними станами, зрозумілими для механізмів прийняття рішень. Це забезпечує пояснюваність функціонування системи, оскільки кожна рекомендація або керуюча дія може бути пов'язана з відповідними онтологічними умовами.

Аналітико-оптимізаційний рівень. Аналітико-оптимізаційний рівень реалізує математичну модель розподілу навантаження для центрів термінальних кластерів. Його основним завданням є визначення такого варіанта розподілу користувацьких запитів між доступними вузлами, який забезпечує мінімізацію затримок, запобігання перевантаженням і збалансоване використання ресурсів.

На вхід цього рівня надходять як числові параметри стану мережі, так і семантично інтерпретовані знання. Це дозволяє поєднати кількісну оптимізацію з контекстними обмеженнями, визначеними в онтології. Наприклад, запит може бути перенаправлений не на будь-який вільний вузол, а лише на той, який відповідає вимогам сервісу, політикам доступу, пріоритету користувача та допустимим часовим обмеженням.

Функцію прийняття рішення можна подати у вигляді:

$$Decision(t) = F(M_L(t), Onet, R_{inf}, C_{qos}), \quad (22)$$

де $M_L(t)$ — динамічна модель навантаження; $Onet$ — онтологія корпоративної мережі; R_{inf} — правила логічного виведення; C_{qos} — обмеження якості обслуговування.

Результатом роботи аналітико-оптимізаційного рівня є множина рекомендованих або автоматично виконуваних керуючих дій:

$$A_c(t) = \{a_1(t), a_2(t), \dots, a_k(t)\}, \quad (23)$$

де $a_i(t)$ може відповідати перенаправленню запитів, масштабуванню ресурсів, обмеженню низькопріоритетного трафіку, активації резервного вузла або зміні параметрів балансувальника.

Інтеграційний рівень забезпечує взаємодію запропонованої архітектури з наявними системами управління університетською мережею. Його основне призначення полягає у тому, щоб онтологічно-керована система могла бути впроваджена без повної заміни чинної інфраструктури.

Цей рівень виконує функції обміну даними з системами моніторингу, балансувальниками навантаження, серверами автентифікації, системами журналювання, мережевими контролерами та адміністративними панелями. Для цього доцільним є використання API-орієнтованої або подієво-орієнтованої архітектури, у якій окремі компоненти взаємодіють через стандартизовані інтерфейси.

У структурі інтеграційного рівня можна виділити такі компоненти: модуль прийому телеметрії, адаптери до зовнішніх систем, сервіс оновлення онтологічної бази, модуль виконання керуючих дій, шину подій та API для взаємодії з адміністративними інструментами. Такий підхід забезпечує масштабованість системи та можливість поступового розширення її функціональності.

Особливо важливо, що інтеграційний рівень виконує роль посередника між абстрактними рішеннями, сформованими на рівні знань і моделей, та конкретними технічними діями у мережевій інфраструктурі. Наприклад, семантичне рішення «перерозподілити навантаження з перевантаженого термінального кластера на резервний вузол» трансформується в конкретний API-виклик до балансувальника або системи оркестрації.

Рівень представлення та пояснення рішень. Рівень представлення забезпечує взаємодію адміністратора або оператора з системою. Він не обмежується звичайною візуалізацією показників, а реалізує функції пояснюваного управління. Це означає, що система повинна не лише відображати стан мережі, але й пояснювати причини сформованих рекомендацій або виконаних керуючих дій.

Інтерфейс користувача має містити інформацію про поточне навантаження термінальних кластерів, прогнозовані стани, виявлені ризики перевантаження, рекомендовані керуючі дії, очікуваний ефект від їх виконання та логічне обґрунтування. Наприклад, система може сформулювати пояснення такого типу: перенаправлення



частини запитів з вузла n_1 на вузол n_3 рекомендовано через перевищення допустимого порогу затримки, зростання кількості активних сесій і наявність резервної обчислювальної потужності на вузлі n_3 . Таким чином, рівень представлення забезпечує прозорість функціонування архітектури та підвищує довіру до автоматизованих рішень.

Запропонована програмна архітектура має низку суттєвих переваг порівняно з традиційними системами управління мережами. По-перше, вона забезпечує семантичну інтерпретацію технічних даних, що дозволяє працювати не лише з метриками, а й зі змістовними станами системи. По-друге, архітектура підтримує інтеграцію математичних моделей з онтологічними правилами, що підвищує обґрунтованість прийнятих рішень. По-третє, модульна структура дає змогу адаптувати систему до різних типів складних об'єктів без повної перебудови програмної реалізації.

Крім того, важливою перевагою є можливість поступового впровадження системи в наявну інфраструктуру університетської мережі. Завдяки інтеграційному рівню запропонована архітектура може працювати поверх існуючих систем моніторингу та керування, розширюючи їх функціональність за рахунок семантичної інтерпретації, прогнозування та знання-орієнтованого прийняття рішень.

Отже, програмна архітектура онтологічно-керованої системи виступає не лише технічним каркасом програмної реалізації, а й методологічною основою інтеграції знань, даних і моделей у єдиному середовищі управління. Саме така інтеграція створює передумови для побудови інтелектуальних систем, здатних забезпечувати адаптивне, пояснюване та ефективне управління складними об'єктами в умовах динамічного навантаження.

Висновки та перспективи подальших досліджень. У роботі розв'язано актуальну науково-прикладну задачу розроблення програмної архітектури онтологічно-керованої системи управління складними об'єктами на прикладі університетської корпоративної мережі. Запропоновано концептуально цілісний підхід, що поєднує формалізоване представлення знань, математичне моделювання динамічних процесів і програмні механізми інтеграції в єдиному керуючому середовищі.

Наукова новизна дослідження полягає у формуванні інтегрованої архітектурної моделі, яка забезпечує узгоджену взаємодію декларативного (онтологічного) та процедурного (алгоритмічного) рівнів. Це дозволяє перейти від традиційного опрацювання мережевих метрик до семантично збагаченого аналізу станів системи, в якому кількісні параметри інтерпретуються у контексті предметної області, обмежень і цільових критеріїв управління.

Розроблено динамічну модель розподілу навантаження для центрів термінальних кластерів, яка враховує часову змінність потоків запитів, гетерогенність ресурсів та багатокритеріальний характер задачі оптимізації. 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.

Обґрунтовано програмну архітектуру системи, яка включає рівні збору та обробки даних, онтологічного представлення знань, семантичної обробки, аналітико-оптимізаційного аналізу, інтеграції та представлення результатів. Запропонована архітектура забезпечує замкнений цикл управління, в якому дані трансформуються у знання, знання — у рішення, а рішення — у керуючі дії з подальшою оцінкою їх ефективності.

Показано, що інтеграція онтологічних моделей із математичними методами оптимізації дозволяє підвищити ефективність функціонування мережі, зменшити затримки обробки запитів і забезпечити більш рівномірне використання обчислювальних та комунікаційних ресурсів. Важливою перевагою підходу є його здатність до адаптації в умовах динамічного навантаження, а також можливість пояснення прийнятих рішень на основі формалізованих знань.

Практичне значення отриманих результатів полягає у можливості впровадження запропонованої архітектури в існуючі системи управління університетськими ІКТ-інфраструктурами без необхідності їх повної перебудови. Використання інтеграційного рівня дозволяє забезпечити сумісність із традиційними засобами моніторингу та керування, що створює передумови для поступової інтелектуалізації наявних мережевих систем.



Перспективи подальших досліджень пов'язані з розширенням математичної моделі з урахуванням невизначеності параметрів (інтервальні та нечіткі оцінки), інтеграцією методів машинного навчання для прогнозування навантаження, а також розробленням цифрових двійників мережевих інфраструктур для підтримки сценарного аналізу та стратегічного планування розвитку систем управління.

Дане дослідження реалізується завдяки грантовій підтримці Національного фонду досліджень України, Проєкт 2025.07/0207 «Концепція знання-орієнтованої технології для моделювання та прогнозування стану навколишнього природного середовища на основі неточних даних».

1. Radovic, M.; Petrovic, N.; Tomic, M. An Ontology-Driven Learning Assessment Using the Script Concordance Test. *Appl. Sci.* 2022, 12, 1472. <https://doi.org/10.3390/app12031472>. 2. 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, no. 10: 369. <https://doi.org/10.3390/ijgi14100369>. 3. Sreevalsan-Nair, J.; Mundayatt, A. Evolution of Data-driven Single- and Multi-Hazard Susceptibility Mapping and Emergence of Deep Learning Methods. *arXiv* 2025, arXiv:2502.09045. 4. Beretta, V.; Desconnets, J.-C.; Mougnot, I.; Arslan, M.; Barde, J.; Chaffard, V. A user-centric metadata model to foster sharing and reuse of multidisciplinary datasets in environmental and life sciences. *Comput. Geosci.* 2021, 154, 104807. 5. Tōnisson, L.; Preden, J. Ontology-Based Data Representation Prototype for Indoor Air Quality, Building Energy Performance, and Health Data Computation. *Sustainability* 2024, 16, 5677. <https://doi.org/10.3390/su16135677> 6. Guerrero-Avedaño, A.; Nieto Bernal, W.; Luna Amaya, C. Governance and Corporate Management System Supported by Innovation, Technology, and Digital Transformation as a Driver of Change. *Sustainability* 2023, 15, 13150. <https://doi.org/10.3390/su151713150> 7. Cardoni, A.; Kiseleva, E.; Lombardi, R. A sustainable governance model to prevent corporate corruption: Integrating anticorruption practices, corporate strategy and business processes. *Bus. Strategy Environ.* 2020, 29, 1173–1185. 8. Min, S.; Kim, B. AI Technology Adoption in Corporate IT Network Operations Based on the TOE Model. *Digital* 2024, 4, 947-970. <https://doi.org/10.3390/digital4040047> 9. El Khatib, M.M.; Al-Nakeeb, A.; Ahmed, G. Integration of cloud computing with artificial intelligence and its impact on telecom sector: A case study. *iBusiness* 2019, 11, 1–10. 10. Dyvak,

M., Melnyk, A., Rot, A., Hernes, M., & Pukas, A. (2022). Ontology of Mathematical Modeling Based on Interval Data. *Complexity*, 2022. **11**. Y. Popyk and M. Ples, "Optimization of energy consumption in data centers based on an ontological approach", *Опт-ел. інф-енерг. техн.*, vol. 50, no. 2, pp. 282–292, Jan. 2026. **12**. Melnyk A., Voznyak S., Sachenko S., Naumchuk V., Korkushko O. Mathematical and Knowledge-Oriented Modeling of Efficient and Balanced Operation of a University Computer Network . *CEUR-WS*. 2025. Vol. 4160. P. 111-125. ISSN: 1613-0073.

REFERENCES

1. Radovic, M.; Petrovic, N.; Tomic, M. An Ontology-Driven Learning Assessment Using the Script Concordance Test. *Appl. Sci.* 2022, **12**, 1472. <https://doi.org/10.3390/app12031472>. **2.** 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, no. 10: 369. <https://doi.org/10.3390/ijgi14100369>. **3.** Sreevalsan-Nair, J.; Mundayatt, A. Evolution of Data-driven Single- and Multi-Hazard Susceptibility Mapping and Emergence of Deep Learning Methods. *arXiv* 2025, arXiv:2502.09045. **4.** Beretta, V.; Desconnets, J.-C.; Mougnot, I.; Arslan, M.; Barde, J.; Chaffard, V. A user-centric metadata model to foster sharing and reuse of multidisciplinary datasets in environmental and life sciences. *Comput. Geosci.* 2021, **154**, 104807. **5.** Tönisson, L.; Preden, J. Ontology-Based Data Representation Prototype for Indoor Air Quality, Building Energy Performance, and Health Data Computation. *Sustainability* 2024, **16**, 5677. <https://doi.org/10.3390/su16135677> **6.** Guerrero-Avedaño, A.; Nieto Bernal, W.; Luna Amaya, C. Governance and Corporate Management System Supported by Innovation, Technology, and Digital Transformation as a Driver of Change. *Sustainability* 2023, **15**, 13150. <https://doi.org/10.3390/su151713150> **7.** Cardoni, A.; Kiseleva, E.; Lombardi, R. A sustainable governance model to prevent corporate corruption: Integrating anticorruption practices, corporate strategy and business processes. *Bus. Strategy Environ.* 2020, **29**, 1173–1185. **8.** Min, S.; Kim, B. AI Technology Adoption in Corporate IT Network Operations Based on the TOE Model. *Digital* 2024, **4**, 947-970. <https://doi.org/10.3390/digital4040047> **9.** El Khatib, M.M.; Al-Nakeeb, A.; Ahmed, G. Integration of cloud computing with artificial intelligence and its impact on telecom sector: A case study. *iBusiness* 2019, **11**, 1–10. **10.** Dyvak, M., Melnyk, A., Rot, A., Hernes, M., & Pukas, A. (2022). Ontology of Mathematical Modeling Based on Interval Data. *Complexity*, 2022. **11.** Popyk, Yu.I., and Marek Ples. 2026. "Optimization of Energy Consumption in Data Centers Based on an Ontological Approach". *Optoelectronic Information-Power Technologies* 50

(2):282-92. <https://doi.org/10.31649/1681-7893-2025-50-2-282-292>. 12.
Melnyk A., Voznyak S., Sachenko S., Naumchuk V., Korkushko O. Mathematical
and Knowledge-Oriented Modeling of Efficient and Balanced Operation of a
University Computer Network . CEUR-WS. 2025. Vol. 4160. P. 111-125. ISSN:
1613-0073.

Andriy Melnyk ^[1; ORCID ID: 0000-0001-7799-9877],
D.Sc. (Tech.), Professor,
Yurii Popyk ^[1],
PhD Student

¹West Ukrainian National University

SOFTWARE ARCHITECTURE OF AN ONTOLOGY-DRIVEN SYSTEM FOR MANAGING COMPLEX

This paper addresses the problem of designing a software architecture for an ontology-driven system for managing complex objects, aimed at ensuring efficient, adaptive, and explainable operation under conditions of dynamic workloads and heterogeneous resources. The relevance of the study is driven by the increasing complexity of modern information and communication infrastructures, particularly university corporate networks, which are characterized by high-intensity user requests, variability of workload profiles, and strict quality-of-service requirements.

An integrated approach is proposed that combines ontology-based modeling, mathematical methods for analysis and optimization, and software integration mechanisms within a unified multi-layer architecture. A formal model of an ontology-driven system is developed, providing a consistent representation of domain knowledge, dynamic data, and decision-making procedures. A dynamic load distribution model for terminal cluster centers is introduced, taking into account temporal characteristics of request flows, structural properties of the network, and multi-criteria constraints.

The software architecture of the system is substantiated, including data acquisition and preprocessing, ontology-based knowledge representation, semantic processing, analytical and

optimization modules, integration mechanisms, and presentation components. A key feature of the architecture is the integration of declarative (ontology-based) and procedural (algorithmic) layers, enabling semantic interpretation of system states, application of logical reasoning mechanisms, and generation of well-founded control decisions.

The results demonstrate that combining knowledge-oriented approaches with mathematical modeling significantly improves load balancing efficiency, reduces response latency, and ensures more balanced utilization of computational and communication resources. The proposed approach is designed for integration into existing network management systems and can serve as a foundation for intelligent management of ICT infrastructures.

Future research directions include extending the models to handle uncertainty (e.g., interval and fuzzy representations), integrating machine learning techniques for workload prediction, and developing digital twins of network infrastructures for scenario-based analysis and strategic planning.

Keywords: ontology, software architecture, complex systems, load balancing, corporate networks, knowledge-driven systems, semantic technologies, adaptive management.

Отримано: 20 січня 2026 року
Прорецензовано: 23 лютого 2026 року
Прийнято до друку: 27 березня 2026 року



© 2026 [Andriy Melnyk, Yurii Popyk]. Licensee [NUWEE]. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial (CC BY-NC) license (creativecommons.org).